國 立 交 通 大 學

應 用 數 學 系

博士論文

在一維度上的有界形和分割及單一形均分割問題

The Bounded-shape Sum-partition and the Single-shape Mean-partition Problems in One-dimension

研 究 生: 張飛黃

指導教授: 黃光明 教授

中 華 民 國 九 十 四 年 六 月

在一維度上的有界形和分割及單一形均分割問題

The Bounded-shape Sum-partition and the Single-shape Mean-partition Problems in One-dimension

研 究 生 : 張飛黃 Student: Fei-huang Chang
指導教授: 黃光明 教授 Advisor: Frank K. Hwang

國 立 交 通 大 學

應 用 數 學 系

博 士 論 文

A Dissertation
Submitted to Department of Applied Mathematics
College of Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Applied Mathematics
June 2005
Hsinchu, Taiwan, Republic of China

中 華 民 國 九 十 四 年 六 月

Abstract

The optimal partition problem considers the partition of n objects into p nonempty parts, and finding a partition(optimal partition) to maximize the objective function $F : R^p \to R$. A brute force method is to compare the values of objective function $F(\pi)$ for each partition π. Thus, we are concerned with the number of all partitions which determines whether the brute force method is practical. However, a more desirable solution is to prove that the objective function has some suitable property which leads to the existence of an optimal partition in a special class of partitions. Then, we need pay attention only to this class of partitions.
The vector of the size of each part is called a shape. If a partition problem has a restriction where the size of each part lies in an interval, then it is called a bounded-shape partition problem. If each interval is degenerated, then it is called single-shape partition problem. In Chapter 2, we use the generating function to count the number of ordered(unordered) shapes and the number of bounded-shape partitions. In Chapter 3, we prove that for bounded-shape sum-partition problem with Schur-convex objective function, there must be a nonmajorized shape such that the corresponding size-consecutive partition is optimal. We also bound the number of nonmajorized shapes, and develop an algorithm to find all nonmajorized shape-types. In Chapter 4, we prove that for single-shape mean-partition problem with quasi-convex objective function, there must be a consecutive optimal partition. We also give some new results for the mean-partition problems.

摘要

最優分法問題主要是對於目標函數$F : R^p \to R$考慮n個物件分成p個非空部份, 如何找到一個分法π(最優分法), 使得$F(\pi)$爲最大值。一個最直接的方法是去比較所有分法的目標函數值。此時, 我們關心所有分法的數量, 因爲它決定了直接法是否可行。更好的是當目標函數具備有某些性質, 使得某些特定的分法類中含有一最優分法, 我們就可以只關心這一些數量較小的特定分法。

每個部分容量所形成的向量我們稱爲形, 若一個問題對每個部分的容量分別落於一個區間內, 我們稱爲有限形分割問題。若每個區間爲單一的點, 則我們稱爲單一形分割問題。在第二章, 我們使用生成函數去計算對於有序及無序的有界形分割各有多少形及分法。在第三章, 我們証明對於有界形和分割問題, 當目標函數爲 Schur 凸函數, 必存在一不被偏形其所對應的容量連續分法爲最優分法。並給出了不被偏形的數量上界及發展了找出所有不被偏等價形的演算法。在第四章, 我們証明對於單一形均分割問題, 當目標函數爲準凸函數, 其最優分法必是一連續分法。並且對均分割問題給了一些新的結果。

誌謝

二十幾年的就學生涯到此算是告一段落了, 今日的我能夠以理工博士的身份結束個人的求學生涯, 要感謝的人實在太多太多, 第一: 要感謝的是在我求學期間指導過我的老師, 首先是當我就讀東華應數系三年級時, 遠從清大來花蓮任教一年的張企老師, 您給了我大學生涯中最有意義的一個A^+, 這個全班唯一的一個A^+, 讓我重新開始拾起書本, 也讓我了解問一個好問題是多麼的有價值, 沒有您就沒有今日的我。再來是我碩士班的指導教授王立中老師以及在我就讀大學及碩士期間給我很多生涯規劃上建議的郭大衛老師, 是因爲這二位老師對學生我的指導及愛護, 使我能在碩士期間得以有論文發表, 也種下了來交大就讀博士班的種子。而來到交大後, 感謝系上的黃大原老師、傅恆霖老師以及翁志文老師, 教導了我許多組合學上的基本知識; 也謝謝陳秋媛老師 給予我很多課業及生活上的意見, 最後也是最重要的要感謝我博士班的指導老師, 黃光明老師, 是您做學問及做人做事的態度深深影響著我, 不但能夠給予我課業上的解惑, 也能在我偷懶或做錯事時給我當頭棒喝, 還能忍受我天馬行空的想法以及耐心的指導我許許多多的缺點; 當學生的我只要有一點點表現, 就給予很大的讚賞及鼓勵, 而當表現不好時, 就馬上給予告誡, 但於他人前仍是稱許學生的好, 對於您的身教及言教, 做學生的我現在無以爲謝, 只有將來於研究這條路上盡心盡力, 方是最好的回報。

第二: 還要感謝我許許多多的好朋友, 首先是陪我一同走過十幾個年頭的國小、國中同窗及好友, 欽垚、世芳、奕霖、麒銘、志銘, 有你們在身旁的日子裡總是特別歡樂, 總會在不經意時想起一同騎機車上草山的日子, 總是在半夜蹺家打撞球, 還有兩個瘋子半夜騎機車到花蓮來找我還是當天來回, 讓我在情緒煩燥的同時或隨時, 有人可以陪我一起瘋狂,

一起打球, 一起做一些很無聊的事, 但這卻是我一直往前走的動力之一, 也一直是我很珍惜的回憶。高中同學, 翊帆、明謙, 雖是不常見面, 但是一見面卻又熟得跟天天見面似的, 能認識你們是我的福氣; 大學好友及碩士同學, 鴻志、于菁、俊良, 總是可以常常一聊就聊得沒日沒夜的朋友; 最後感謝交大碩班91及92級組合組的學弟妹 (族繁不及備載), 同學君逸、宏賓以及學姐琲琪, 謝謝你們在交大期間的陪伴、照顧及帶給我一個歡樂的交大生活。

第三: 就是謝謝我的家人, 爸、媽及老弟是你們在背後的支持, 我才能無憂無慮的完成我的學業, 當然也要謝謝我的岳父及岳母, 能夠體諒我的難處, 肯將您們心愛的女兒托付給我, 也在我求學的最後一段時間能幫我照顧小孩, 當然最要感謝的是一直支持我的親愛老婆蕙慈, 從大學時期起, 就一直陪伴在我的身旁, 很多時候都是有你的鼓勵與支持, 才能讓我這一路走的這樣順利, 最後也謝謝最新產生的動力及希望, "騰達", 看著你小小的身軀, 老爸就會覺得一切的努力都是值得的。誌謝文短, 感謝卻意長, 謹以此篇論文, 獻給您們。

(p.s.要感謝的人實在太多太多, 總之謝謝曾經或者一路陪我走來的師長、朋友及家人們)

Contents

Chapter 1

Introduction

In this Chapter, we introduce the background of the optimal partition problem and give a summary of the following Chapters.

1.1 The one dimensional partition problem

The partition problem studies the partitioning of n numbers into finite nonempty parts so as to maximize an objective function subject to certain constraints on the number of elements in each part. Applications of the partition problem include inventory grouping, scheduling, reliability, graph partitioning, hypothesis testing in statistics, circuit layout, clustering, symbolic computation, location problems, storage allocation, group testing, system reliability, etc.., see [16] for a survey.

Consider a partition π of $\{1, ..., n\}$ into p nonempty parts $\pi_1, ..., \pi_p$. If the number of parts is fixed to be p, we call it a p-partition(*size*-partition); otherwise we call it an *open* partition. Let $n_1, ..., n_p$ be the sizes of $\pi_1, ..., \pi_p$ where $\sum_{i=1}^{p} n_i = n$. We define the *shape* of π as the vector $(n_1, ..., n_p)$. If the cardinalities of the p parts are fixed to be $(n_1, ..., n_p)$, then we call it a $(n_1, ..., n_p)$-partition(*single-shape*-partition). If the size of each parts must lie in a range, i.e., nonnegative integer p-vectors $L = (L_1, ..., L_p)$ and $U = (U_1, ..., U_p)$ are given where

$$\sum_{i=1}^{p} L_i \leq n \leq \sum_{i=1}^{p} U_i, \tag{1.1.1}$$

and the shape $(n_1, ..., n_p)$ of a feasible partition satisfies for each i

$$L_i \leq n_i \leq U_i, i \in \{1, ..., p\}, \tag{1.1.2}$$

then we call it a *bounded-shape*-partition. Define $\Gamma(L, U)$ to be the set of all partitions whose shapes satisfy (1.1.1) and (1.1.2). If the number of parts is fixed at p and only a set of shapes is allowed, then we called it a *constrained-shape*-partition. In addition, we have the two categories of *ordered* partitions and *unordered* partitions. An ordered partition is a sequence $(\pi_1, ..., \pi_p)$, while an unordered partition is a set $\{\pi_1, ..., \pi_p\}$.

Given an objective function $F(\pi)$, our goal is to find a partition(optimal solution) to maximize it. A brute-force way is to enumerate and evaluate all legitimate partitions to get the optimal solution of the objective function. Whether this is a practical method depend on the number of legitimate partitions. The counting of ordered and unordered partitions for open partitions, size partitions and shape partitions are fundamental combinatorial problems and have been well documented [16]. Although, theoretically, we could count bounded-shape partitions also by summing up all legitimate shapes, that could be unwieldy in practice. We will study the generating function approach to count bounded-shape partitions in Chapter 2.

Suppose we know that the objective function has an optimal solution in a special class of partitions, then we need pay attention only to this class of partitions. Two such classes are consecutive partitions and size-consecutive partitions. A partition is called *consecutive* if each part consists of consecutive integers. 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. In Chapter 3, we prove that when the objective function of the sum-partition problem with bounded-shape is Schur convex(see Sec.1.2), we need pay attention only to size-consecutive partitions with nonmajorized shapes. In Chapter 4, we show that for the single-shape partition, results obtained for the sum-partition problem also apply to the mean-partition problems.

1.2 The bounded-shape sum-partition problem for Schur convex objective function

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

$$\sum_{i=1}^{k} a_{[i]} \geq \sum_{i=1}^{k} b_{[i]} \quad \text{for} \quad k = 1, \ldots, p-1 \tag{1.2.1}$$

and

$$\sum_{i=1}^{p} a_i = \sum_{i=1}^{p} b_i \,. \tag{1.2.2}$$

We say that a *strictly majorizes* b if $a \in S$ majorizes b but b does not majorize a. If a majorizes b for each $b \in S \subseteq R^p$, then a is called *majorizing* vector in S, if a is not majorized by any $b \in S$, a is called a nonmajorized vector in S. A real-valued function f on R^p is *Schur convex* if $f(a) \geq f(b)$ whenever a majorizes b. A Schur convex function is known to be symmetric. For a partition $\pi = (\pi_1, ..., \pi_p)$ let

$$\theta_\pi = (\sum_{j\in\pi_1} \theta_j, \ldots, \sum_{j\in\pi_p} \theta_j), \tag{1.2.3}$$

Hwang and Rothblum [15] considered the sum-partition problem of maximizing the objective function

$$F(\pi) = f(\sum_{j\in\pi_1} \theta_j, \sum_{j\in\pi_2} \theta_j, \ldots, \sum_{j\in\pi_p} \theta_j), \tag{1.2.4}$$

over partitions π having shape in a prescribed set with f being Schur convex. In particular, they proved Theorem 1.2.1.

Theorem 1.2.1. *Suppose f is Schur convex, Γ is a set of positive integer p-vector 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 $f(\theta_{\pi'}) \geq f(\theta_\pi)$.

In particular, when Γ contains a single shape, if $n_1 \leq ... \leq n_p$, then the following explicit partitions are optimal under (a) or (b), respectively

$$\pi_i = (\sum_{j=1}^{i-1} n_j + 1, ..., \sum_{j=1}^{i} n_j) \text{ for } i = 1, ..., p \tag{1.2.5}$$

and

$$\pi_i = (n - \sum_{j=1}^{i} n_j + 1, ..., n - \sum_{j=1}^{i-1} n_j) \text{ for } i = 1, ..., p. \tag{1.2.6}$$

They also considered the problem with bounded shapes. They 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. They also gave a sufficient condition for the existence of the majorizing shape. The sufficient condition is that the order of upper bounds over the p parts equal to the order of lower bounds. By Theorem 1.2.1, if f is Schur convex and $\Gamma = \Gamma(L, U)$ has the majorized shape, then we can find an optimal solution in the (reverse)size-consecutive class.

In Chapter 3, we extend Theorem 1.2.1 to the case when the majorized shape doesn't exist and Γ is a set of bounded-shapes.

1.3 The polytope approach to the sum-partition problem

Given a real-value function λ on the subsets of $\{1, ..., p\}$ with $\lambda(\phi) = 0$, each permutation $\sigma = (\sigma_1, ..., \sigma_p)$ of $\{1, ..., p\}$ defines a vector $\lambda_\sigma = ((\lambda_\sigma)_1, ..., (\lambda_\sigma)_p)$ such that

$$(\lambda_\sigma)_k = \lambda(\bigcup_{i=1}^{j} \sigma_i) - \lambda(\bigcup_{i=1}^{j-1} \sigma_i), \text{ with } \sigma_j = k \text{ for } 1 \leqslant k \leqslant p.$$

λ is called *supermodular* if for all subsets I, J of $\{1, ..., p\}$,

$$\lambda(I \cup J) + \lambda(I \cap J) \geqslant \lambda(I) + \lambda(J),$$

and *strictly supermodular* if the inequality is strict for all I, J not satisfying $I \subseteq J$ or $J \subseteq I$. The *permutation polytope* induced by λ, denoted H^λ, is the convex hull of $\{\lambda_\sigma : \text{all } \sigma\}$. For example, Shapley [18] studied the case of convex p-person game. For a subset $I \subseteq \{1, ..., p\}$, let $\lambda(I)$ denote the payoff to I, if the members of I form an alliance. Then stability of an alliance $I \cup J$ requires λ to be supermodular. The core of a convex p-person game is the solution set of the linear inequality system

$$\sum_{i \in I} x_i \geq \lambda(I) \text{ for all } I \subseteq \{1, ..., p\} \text{ and } \sum_{i=1}^{p} x_i = \lambda(\{1, ..., p\}). \qquad (1.3.1)$$

Let C^λ denoted the polytope defined by (1.3.1). Shapley [18] proved

Theorem 1.3.1. *Suppose λ is supermodular. Then*

(a) $H^\lambda = C^\lambda$,

(b) the vectices of H^λ are precisely the λ_σ's where σ ranges over all permutations of $\{1, ..., p\}$.

Gao et al.[9] studied the single-shape sum-partition problem to maximize an objective function $f(\sum_{j \in \pi_1} \theta_j, ..., \sum_{j \in \pi_p} \theta_j)$. For I a subset of $\{1, ..., p\}$, define $n(I) = \sum_{i \in I} n_i$. They defined

$$\lambda_S(I) = \sum_{j=1}^{n(I)} \theta_j \qquad (1.3.2)$$

and proved λ_S is supermodular. Here, H^{λ_S} is the convex hull of all partitions corresponding to $\{(\lambda_S)_\sigma : \text{ all } \sigma\}$ (each partition is a point), and C^{λ_S} is the polytope defined by

$$\sum_{i\in I}\sum_{j\in\pi_i}\theta_j \geqslant \lambda_S(I) \text{ for all } I \subseteq \{1,...,p\} \text{ and } \sum_{j=1}^{n}\theta_j = \lambda_S(\{1,...,p\}).$$

Let P denote the convex hull of all partitions satisfying the given single shape. Clearly, $H^{\lambda_S} \subseteq P \subseteq C^{\lambda_S}$. By Theorem 1.3.1, $H^{\lambda_S} = P = C^{\lambda_S}$. They proved the existence of a consecutive optimal partition for the single-shape partition problem when f is quasi-convex.

1.4 The mean-partition problem

Consider $N = \{1,...,n\}$ where each element i in N is associated with a number $\theta_i \in R$. Partition problems are further classified by their objective function $F(.)$. For a subset S of $\{1,...,n\}$, let

$$\bar{\theta}_S = \frac{1}{|S|}\sum_{i\in S}\theta_i \in R, \tag{1.4.1}$$

and for a partition $\pi = (\pi_1,...,\pi_p)$, let

$$\bar{\theta}_\pi = (\bar{\theta}_{\pi_1},...,\bar{\theta}_{\pi_p}) \in R^p. \tag{1.4.2}$$

A class of partition is the mean-partition problem in R in which

$$F(\pi) = g(\bar{\theta}_\pi), \tag{1.4.3}$$

where g is a real-valued function on R^p. The mean-partition polytope corresponding to a set of partitions Π is denoted by M^Π. While the sum partition problem has been dominating in optimal partition problems, the mean partition problems have also been considered. Anily and Federgruen [1] first studied the single-shape mean-partition problem. In Chapter 4, we will give an approach to solve the single-shape mean-partition problem, and discuss the difficulties of the bounded-shape mean-partition problem. Finally, we give some new results in the mean-partition problem.

Chapter 2

Counting the Number of Bounded-shape

In this chapter, we use the generating function approach to count bounded-shape partitions. When the θ_i's are constant, then the number of bounded-shape partitions is reduced to the number of bounded shapes. We obtain a neat solution of that number for ordered partitions.

2.1 The generating function approach

For given lower bound L_i and upper bounds U_i, $1 \le i \le p$, define

$\#_n^*$: The number of ordered bounded shapes.

$\overline{\#}_n^*$: The number of unordered bounded shapes.

$\#_n$: The number of ordered bounded-shape partitions.

$\overline{\#}_n$: The number of unordered bounded-shape partitions.

Define

$$L^* = \sum_{i=1}^{p} L_i$$

and

$$\begin{aligned} g_n(x) &= \prod_{i=1}^{p} \left(\sum_{j=L_i}^{U_i} a_j x^j \right) \\ &\equiv \sum_{k \geq L^*} c_k x^k. \end{aligned}$$

It is well known [2] that

$$c_k = \frac{g_n^{(k)}(0)}{k!}.$$

We show that $\#_n^*$, $\overline{\#}_n^*$, $\#_n$ and $\overline{\#}_n$ can be expressed as different functions of c_n.

Theorem 2.1.1.

(i) $\#_n^* = c_n$ *by setting* $a_j = 1$ *in* $g_n(x)$.

(ii) $\overline{\#}_n^* =$ *the number of distinct terms in* c_n.

(iii) $\#_n = n! c_n$ *by setting* $a_j = 1/(j!)$.

(iv) $\overline{\#}_n =$ *same as* (iii) *except counting only distinct terms in* c_n, *and dividing a term* $\prod_{i=1}^{p} a_i^{e_i}$ *by* $\prod_{i=1}^{p}(e_i!)$.

Proof.

(i) Every ordered shape $(n_1, n_2, \ldots, n_p)$ summing to n contributes 1 to c_n.

(ii) Two shapes $(n_1, \ldots, n_p)$ and $(n'_1, \ldots, n'_p)$ are not distinguishable if $\{n_1, \ldots, n_p\} = \{n'_1, \ldots, n'_p\}$. A shape $(n_1, \ldots, n_p)$ is preserved in the coefficient term $a_{n_1}, \ldots, a_{n_p}$. Hence we count only distinct $a_{n_1}, \ldots, a_{n_p}$(as coefficient, the ordering is not important) terms.

(iii) Each shape $(n_1, \ldots, n_p)$ yields $\binom{n}{n_1, \ldots, n_p}$ distinct partitions.

(iv) The division is because interchanging two parts of same size results in the same unordered partition.

$\square$

Example 1.

$$n = 10, n_1 \in [2,4], n_2 \in [2,6], n_3 \in [3,5].$$

(i)

$$\begin{aligned} g(x) &= (x^2 + x^3 + x^4)(x^2 + x^3 + x^4 + x^5 + x^6)(x^3 + x^4 + x^5) \\ &= x^7 + 3x^8 + 6x^9 + 8x^{10} + 9x^{11} + 8x^{12} + 6x^{13} + 3x^{14} + x^{15}. \end{aligned}$$

There are $c_{10} = 8$ ordered bounded shapes which are $(2,3,5)$, $(2,4,4)$, $(2,5,3)$, $(3,2,5)$, $(3,3,4)$, $(3,4,3)$, $(4,2,4)$, $(4,3,3)$.

(ii)

$$\begin{aligned} g(x) &= (a_2x^2 + a_3x^3 + a_4x^4)(a_2x^2 + \cdots + a_6x^6)(a_3x^3 + a_4x^4 + a_5x^5) \\ &= \cdots + (3a_2a_3a_5 + 2a_2a_4^2 + 3a_3^2a_4)x^{10} + \ldots. \end{aligned}$$

Hence, there are 3 unordered bounded shapes which are $(2,3,5)$, $(2,4,4)$, $(3,3,4)$.

(iii)

$$\begin{aligned} g(x) &= (\frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!})(\frac{x^2}{2!} + \cdots + \frac{x^6}{6!})(\frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!}) \\ &= \frac{1}{24}x^7 + \frac{11}{288}x^8 + \frac{89}{4320}x^9 + \frac{7}{960}x^{10} + \frac{17}{8640}x^{11} + \ldots. \end{aligned}$$

Hence, there are $10! \times \frac{7}{960} = 26460$ ordered bounded-shape partitions.

(iv)

$$\begin{aligned} g(x) &= (\frac{a_2x^2}{2!} + \cdots + \frac{a_4x^4}{4!})(\frac{a_2x^2}{2!} + \cdots + \frac{a_6x^6}{6!})(\frac{a_3x^3}{3!} + \cdots + \frac{a_5x^5}{5!}) \\ &= \cdots + (\frac{3a_2a_3a_5}{1440} + \frac{2a_2a_4^2}{1152} + \frac{3a_3^2a_4}{864})x^{10} + \ldots. \end{aligned}$$

The number of unordered bounded-shape partitions is

$$\frac{10!}{1440} \times \frac{1}{(1!)(1!)(1!)} + \frac{10!}{1152} \times \frac{1}{(1!)(2!)} + \frac{10!}{864} \times \frac{1}{(2!)(1!)} = 6195.$$

Although a generating function counting is equivalent to enumeration, it gives a particular way of enumeration, hence doable by a computer program.

2.2 A neat solution

We show that the generating function approach leads to a neat formula for the number of ordered bounded shapes.

Define

$$R_i = U_i - L_i, 1 \le i \le p,$$

$$S = \{(s_1, \ldots, s_p) \mid s_i \in \{0, R_i + 1\} \text{ for } 1 \le i \le p, \sum_{i=1}^{p} s_i \le n - L^*\}$$

and

$$s_{p+1} = n - L^* - \sum_{i=1}^{p} s_i.$$

Let $|(s_1, \ldots, s_p)|$ denote the number of positive s_i.

Theorem 2.2.1.

$$\#_n = \sum_{(s_1,\ldots,s_p)\in S} (-1)^{|(s_1,\ldots,s_p)|} \binom{p + s_{p+1} - 1}{p - 1}.$$

Proof.

$$\begin{aligned} g(x) &= (x^{L_1} + x^{L_1+1} + \cdots + x^{U_1}) \ldots (x^{L_p} + x^{L_p+1} + \cdots + x^{U_p}) \\ &= x^{L^*}(1 + x + \cdots + x^{R_1})(1 + x + \cdots + x^{R_2}) \ldots (1 + x + \cdots + x^{R_p}) \\ &= x^{L^*}(1 - x^{R_1+1})(1 - x^{R_2+1}) \ldots (1 - x^{R_p+1})(1 - x)^{-p}. \end{aligned}$$

Using the Leibniz formula,

$$\left(\prod_{i=1}^{m} f_i(x)\right)^{(n)} = \sum_{n_1+\cdots+n_m=n} \binom{n}{n_1, \ldots, n_m} \prod_{i=1}^{m} f_i^{(n_i)}(x),$$

where n_i is the largest exponent of $f_i(x)$, on the first $p+1$ terms of $g(x)$,

$$\begin{aligned} g^{(n)}(0) &= \sum_{(s_1,\ldots,s_p)\in S} \binom{n}{L^*, s_1, \ldots, s_p, s_{p+1}} (-1)^{|(s_1,\ldots,s_p)|} L^*! s_1! \ldots s_p! \left[\frac{d^{s_{p+1}}}{dx^{s_{p+1}}}(1 - x)^{-p}\right]_{x=0} \\ &= \sum_{(s_1,\ldots,s_p)\in S} \frac{n!}{s_{p+1}!} (-1)^{|(s_1,\ldots,s_p)|} \frac{(p + s_{p+1} - 1)!}{(p - 1)!} \left[(1 - x)^{-p-s_{p+1}}\right]_{x=0} \\ &= n! \sum_{(s_1,\ldots,s_p)\in S} (-1)^{|(s_1,\ldots,s_p)|} \binom{p + s_{p+1} - 1}{p - 1} \left[(1 - x)^{-p-s_{p+1}}\right]_{x=0}. \end{aligned}$$

Thus $\#_n = c_n = \frac{g_n^{(n)}(0)}{n!} = \sum_{(s_1,\ldots,s_p)\in S} (-1)^{|(s_1,\ldots,s_p)|} \binom{p+s_{p+1}-1}{p-1}$. $\square$

A size-partition can be interpreted as a bounded-shape partition with uniform bounds $L_i = 1, U_i = n$. Then

$$L^* = p,$$
$$R_i = n-1,\ 1 \le i \le p.$$

Necessarily,

$$s_i = 0 \text{ for } 1 \le i \le p.$$

and

$$s_{p+1} = n - L^* = n - p.$$

Corollary 2.2.2. *The number of ordered size-partitions is* $\binom{n-1}{p-1}$.

Example 2.

$$n = 14, n_1 \in [2,4], n_2 \in [2,8], n_3 \in [3,6].$$
$$R_1 = 2, R_2 = 6, R_3 = 3, L^* = 7, n - L^* = 7.$$
$$S = \{(0,0,0), (3,0,0), (0,7,0), (0,0,4), (3,0,4)\}.$$

Hence

$$\begin{aligned}\#([2,4],[2,8],[3,6]) &= (-1)^0\tbinom{3+7-1}{3-1} + (-1)^1\tbinom{3+4-1}{3-1} + (-1)^1\tbinom{3+0-1}{3-1} \\ &+ (-1)^1\tbinom{3+3-1}{3-1} + (-1)^2\tbinom{3+0-1}{3-1} \\ &= \tbinom{9}{2} - \tbinom{6}{2} - 1 - \tbinom{5}{2} + 1 \\ &= 11.\end{aligned}$$

The shapes are: $(2,6,6)$, $(2,7,5)$, $(2,8,4)$, $(3,5,6)$, $(3,6,5)$, $(3,7,4)$, $(3,8,3)$, $(4,4,6)$, $(4,5,5)$, $(4,6,4)$, $(4,7,3)$.

Chapter 3

The Bounded-shape Sum-partition Problem in R^1 with Schur Convex Objective Function

In this chapter, we consider the bounded-shape sum-partition problem in R^1 with Schur convex objective function. We will show that the θ_i's are one-sided, one can restrict attention to (reverse) size-consecutive partitions with a nonmajorized shape. As a (reverse) size-consecutive partition with a given shape is easy to determine(see (1.2.5) and (1.2.6)), the problem of finding an optimal partition is reduced to the task of identifying a set of shapes that contains all nonmajorized ones. Since Schur convex functions are symmetric, 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 restrict attention to representatives of *shape-types* which are the equivalence classes of shapes with respect to coordinate-permutations. We will study nonmajorized shapes, bound their numbers and develop algorithms to enumerate them, too. Our study extends the analysis of a previous paper [15] which discussed the above problem assuming the existence of a majorizing shape.

3.1 Nonmajorized shapes

We explore the relation between shape-majorization and the optimization problem with Schur-convex objective function over partitions introduced in the Introduction. In particular, we explore the role of nonmajorized shapes, with respect to $\Gamma(L, U)$.

Corollary 3.1.1. *Suppose f and Γ are as in Theorem 1.2.1, but no majorizing shape exists.*

(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.

Corollary 3.1.1 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 shapes are 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 (1.2.4)). We conclude that the underlying optimization problem can be solved by obtaining a list that contains all nonmajorized shapes, determining corresponding size-consecutive partitions, and evaluating the right-hand side of (1.2.4) for each one of them to select the best. 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 $\Gamma(L, U)$.

In the bounded-shape case which the majoring shape doesn't exist [4], consider 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 3.1.2. *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*

$$[a_J \text{ majorizes } b_J] \Leftrightarrow [a \text{ majorizes } b]; \tag{3.1.1}$$

further (3.1.1) 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 $m \equiv |W|$ ($W = \phi$ and $m = \phi$ 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 $H \setminus J$ with $|H| = k$. As a majorizes b,

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

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 (3.1.1) 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 3.1.2 will be used particularly with sets J consisting of two elements.

Throughout the remainder of this section, let L and U be nonnegative integer p-vectors that satisfy (1.1.1)–(1.1.2). 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 3.1.3. *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*

$$L_j < n_i < U_i \text{ and } L_j < n_j < U_i, \tag{3.1.2}$$

(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) $\Rightarrow$ *(b)* $\Rightarrow$ *(c)* $\Rightarrow$ *(d).*

Proof. (a) $\Rightarrow$ (b). Suppose n_i and n_j satisfy (3.1.2) 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 3.1.2).

(b) $\Rightarrow$ (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) $\Rightarrow$ (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 3.1.3 does not imply condition (a).

Example 3. 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 (3.1.2), observe that the only coordinate of s that is strictly larger than the lower bound is the first one, so if (3.1.2) 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 3.1.3 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 3.1.4. *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 3.1.3. 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 3.1.3 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 $r, t \in \{1, \ldots, p\}$

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

and

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

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

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

Lemma 3.1.5. *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 coordinates i and coordinate j, where $n_i > n'_i$, then s' is obtained from s by permuting these coordinates,*

$$n_i = U_i \text{ or } n_j = L_j \tag{3.1.3}$$

and

$$n'_i = L_i \text{ or } n'_j = U_j. \tag{3.1.4}$$

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 3.1.3 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 $n'_i = L_i$. □

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 $\Gamma(L, U)$.

Corollary 3.1.6. *If s and s' are nonmajorized shapes which are not equivalent, then they differ in at least 3 coordinates.*

3.2 The number of nonmajorized shape-types

In this section, we continue to assume that L and U are integer p-vectors satisfying (1.1.1) and $L \ll U$. As strict-majorization is invariant of the corresponding shape-types, we can and will refer to nonmajorized shape-types.

We note that a single nonmajorized shape-type may correspond to many shapes as example 4.

Example 4. 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 shape-type can be identified with the multiset $\{n_1, \ldots, n_p\}$ where $(n_1, \ldots, n_p)$ is a shape in $\Gamma(L, U)$.

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, $U(s) = \{j \in \{1, \ldots, p\} : n_j = U_j\}$, $M(s) = \{j \in \{1, \ldots, p\} : L_j < n_j < U_j\}$ and $L(s) = \{j \in \{1, \ldots, p\} : n_j = L_j\}$.

Lemma 3.2.1. *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 $j \notin U(s)$ that satisfies

$$U_{i'} > n'_j \text{ for every } i \text{ in } U(s') \tag{3.2.1}$$

and

$$U_i \leq n'_j \text{ for every } i \text{ in } U(s) \setminus U(s'). \tag{3.2.2}$$

Proof. (a) Lemma 3.1.3 assures that $|M(s)| \leq 1$ and $|M(s')| \leq 1$. Thus, if $U(s) = U(s')$, then s and s' can differ in at most 2 coordinates; it then follows from Corollary 3.1.6 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 3.1.3, 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)⇒(c) part of Lemma 3.1.3 implies that $n'_j < n'_i = U_i$, proving (3.1.6).

Next, assume that i is in $U(s) \setminus U(s')$. As s and s' differ by at least 3 coordinates (Corollary 3.1.6), 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

$$n'_j - n_j > n_i - n'_i = U_i - n'_i. \tag{3.2.3}$$

Assume that $U_i > n'_j$ and we will establish a contradiction. By summing $U_i > n'_j$ and (3.2.3), 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 3.1.2). Further, (3.2.3) 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 (3.2.2). □

Corollary 3.2.2. *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 3.2.1 assures that $M(s')$ and $M(s'')$ are nonempty. Let $M(s') = \{i\}$ and

$M(s'') = \{j\}$. Without loss of generality, assume that $n'_i \leq n''_j$. By Lemma 3.2.1(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 3.2.1(b), $n'_i \geq U_k > n''_j$, contradicting our assumption $n'_i \leq n''_j$. $\square$

We next explore the combinatorial restriction imposed by the conclusion of Corollary 3.2.2. 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 conclusions of Corollary 3.2.2, 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 $p = 0, 1, 2, 3, 4, 5, 6$.

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

Table 1

Theorem 3.2.3. *$f(p) \leq 2^{p-1}$ for $p \geq 4$.*

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 $\{U \setminus \{p\} : U \in F_1(p)\}$ 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 $p \geq 4$. $\square$

Corollary 3.2.4. *For $p \geq 4$, there are at most 2^{p-1} nonmajorized shape-types.*

Proof. Corollary 3.2.2 and Lemma 3.2.1 show that $f(p)$ bounds the number of nonmajorized shape-types and Theorem 3.2.3 shows that $f(p) \leq 2^{p-1}$. □

The proof of Corollary 3.2.4 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 [19], $\binom{p-1}{\lfloor (p-1)/2 \rfloor}$ 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 examples achieve this (conjectured) bound.

Example 5. Let $U = (20, 19, 18, 17, 16)$, $L = (1, 2, 3, 4, 5)$, $n = 51$. Using the algorithms of Section 4 (see Example 7), one can show that the set of all nonmajorized shape-types contains 6 shapes that are listed below in Table 2.

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

Table 2

Example 6. For any $p \geq 3$, let $U = (2(p + (\lceil \frac{p-1}{2} \rceil)(\lfloor \frac{p-1}{2} \rfloor)) - 1, 2(p + (\lceil \frac{p-1}{2} \rceil)(\lfloor \frac{p-1}{2} \rfloor)) - 2, \ldots, p + 2(\lceil \frac{p-1}{2} \rceil)(\lfloor \frac{p-1}{2} \rfloor))$, $L = (1, 2, \ldots, p)$, and $n = p^2 + (p - X(p))(\lceil \frac{p-1}{2} \rceil)(\lfloor \frac{p-1}{2} \rfloor)$, where $X(p) = 1$, if p is odd, otherwise, $X(p) = 0$. Then the number of nonmajorized shape-types achieve conjectured bound.

3.3 Identifying all nonmajorized shapes and shape-types

Algorithm 1. (For enumerating all nonmajorized shapes in $\Gamma(L, U)$)

The input for the algorithm consists of integer p-vectors L and U that satisfy (1.1.1) and $L \ll U$.

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

(i) Set $B = \{1, \ldots, p\} \setminus A \setminus \{u\}$, $U_A = \sum_{i \in A} U_i$, $L_B = \sum_{i \in B} L_i$ and $M_u = n - U_A - L_B$.

(ii) If $L_u \leq M_u \leq U_u$, set

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

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 majorized by any shape in TEMP.

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

Lemma 3.3.1. (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* $O(p^3 2^p)$.

Proof. (a) Lemma 3.1.3 (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 $O[(p+p\lg p)p2^{p-1}+(p2^{p-1})^2p] = O[p^3 2^{2p}]$. $\square$

Given integer p-vectors L and U satisfying (1.1.1)–(1.1.2), the set of *floating indices of* (L,U) is defined as $\{i = 1,\dots,p : L_i < U_i\}$. 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

$$U_i + \sum_{j\neq i} L_j \leq n; \tag{3.3.1}$$

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

$$L_i + \sum_{j\neq i} U_j \geq n, \tag{3.3.2}$$

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, (1.2.6) and (1.1.1) stay in effect when an upper bound or a lower bound is adjusted.

Lemma 3.3.2. *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, (1.2.6) and (1.1.1) 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 of an upper bound does not invalidate the effectiveness of a lower bound and an increase of 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 3.3.2 as an *adjustment of the bounds*. We observe that (1.2.6) 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 $\Gamma(L, U)$)

The input for the algorithm consists of integer p-vectors L and U that satisfy (1.2.6). Set $r = 1$.

<u>Iteration r</u>:

(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 $i \in \{1, \ldots, p\} \setminus F$.

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 $F \setminus \{i\}$ can be partitioned into two sets A and B such that

$$|A| \geq 2, \ |B| \geq 2 \tag{3.3.3}$$

$$\max_{k \in B} U_k \leq n - \sum_{j \in A} U_j - \sum_{k \in B} L_k \leq \min_{j \in A} L_j \tag{3.3.4}$$

and

$$L_i < n - \sum_{j \in A} U_j - \sum_{k \in B} L_k < U_i. \tag{3.3.5}$$

When the above holds with 3.3.4 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 3.3.4. Suppose $n_i = \max_{k \in B} U_k \equiv U_x$. Check the existence of a part y in $B \setminus \{x\}$ 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 3.3.4 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 $F \setminus \{i\}$ into sets A and B that satisfy (3.3.3)–(3.3.5). 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 how Algorithm 2 is executed without the need for an E-step.

Example 7. Applying Algorithm 2 to Example 5 is summarized in Figure 1.

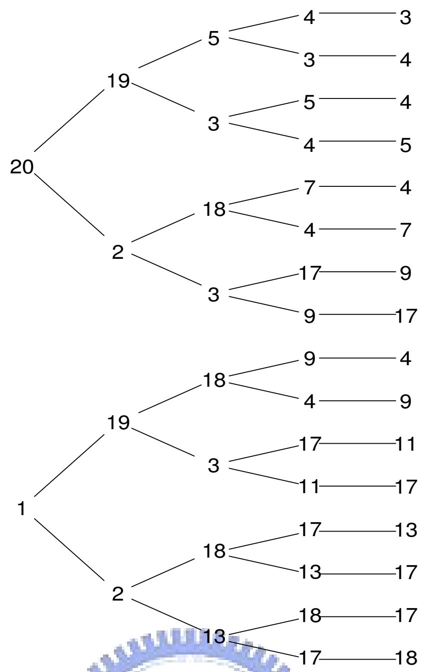

Figure 1.

The corresponding nonmajorized shapes are listed in Table 2.

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 8. $U = (13, 12, 12, 8, 8, 4, 4)$, $L = (1, 10, 10, 6, 6, 2, 2)$ 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 partitions of the three groups $\{2, 3\}$, $\{4, 5\}$, $\{6, 7\}$ of parts in Example 8 having, respectively, the same bounds. In general, g groups would yield up to $g - 1$ partitions.

Example 9. $U = (11, 10, 10, 10, 7, 7, 7, 5, 3, 3, 3)$, $L = (1, 9, 9, 9, 6, 6, 6, 4, 2, 2, 2)$ 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 (3.3.4), and s^3 and s^4 are examples of an E-shape with nonstrict inequalities in (3.3.4).

Example 10. $U = (11, 9, 8, 10, 4, 4, 4, 4, 4)$, $L = (3, 6, 6, 0, 2, 2, 2, 2, 2)$ 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 3.3.3. *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 3.3.4, 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 3.3.4. *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 largest n'_i and the $k-j$ smallest n'_i must be equivalent to N_k.*

Proof. We prove Lemma 3.3.4 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 3.3.3 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 3.3.5. *A regular shape output by Algorithm 2 is nonmajorized.*

Theorem 3.3.6. *(a)Every shape that is constructed by Algorithm 2 is non-majorized.*

(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 3.3.5, we only need to consider an E-shape s. Suppose to the contrary that s' majorizes s. By Lemma 3.3.4, s' majorizes s in the remaining $p-k$ parts. But this is impossible by our construction of an E-shape whose largest k-sum, $1 \le k \le |A|$, is $\ge$ the largest k-sum of s', and whose smallest k-sum, $1 \le k \le |B|$, $\le$ 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 $i \ne j$. 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 3.3.3, $U_i \succ U_j$ 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 $(L_j, U_j) \cap (L_x, U_x) \neq \varnothing$. 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 (3.3.4) is satisfied.

Finally, we justify (3.3.3). 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 of the upper bound of i is due to 3.3.3 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 $U = (U_1, ..., U_p)$, $L = (L_1, ..., L_p)$, and n. The outputs are all possible combinations of A and B.

1: Obtain $U_1 \geq U_2 \geq \cdots \geq U_p$ by sorting U.
2: $sep := L_1$
3: Determine the order statistic, say r, of sep in U.
4: **for** $i = 2$ to p **do**
5:     **if** i=r **then**
6:         $sep := L_r$
7:     **else if** $i = r - 1$ **then**
8:         Output $A = \{1, 2, \ldots, i\}$, $B = \{i+1, i+2, \ldots, p\}$
9:     **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 (3.3.4) and (3.3.5) 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 taking $O(p \log p)$ time. There are $O(2^p)$ steps in Algorithm 2 with at most $O(2^{p-5})$ of them can contain an E-step. The generation of regular shapes takes constant time at every step. Therefore the total time is $O(2^p) + O(2^{p-5})O(p \log p) = O(2^p + p2^{p-5} \log p)$. $\square$

3.4 Determining 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 3.3 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 3.4.1. *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 $U_i = \max\limits_{1 \le j \le p} U_j$. 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, $s'_T = s_T \setminus \{U_i\}$. We prove $s'_B = s_B \setminus \{U_i\}$ (here we refer to shape-types as *multisets*).

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

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

or equivalently,

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

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 $s'_B = s_B \setminus \{U_i\}$.

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, $s \setminus \{U_i\}$ majorizes $s' \setminus \{U_i\}$. 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 3.4.1 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 11. $s_T = (20, 19, 3, 4, 5)$ and $s_B = (1, 2, 18, 17, 13)$. Hence no majorizing shape exists.

Example 12. $U = (100, 90, 60, 50, 17)$, $L = (10, 70, 10, 48, 10)$. 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.

Chapter 4

The Mean-partition Problem

In the mean-partition problem the goal is to partition a finite set of elements, each associated with a number, into p disjoint parts so as to optimize an objective function which depends on the averages of the vectors that are assigned to each part. A partition is then associated with a p-vector $\bar{\theta}_\pi = (\bar{\theta}_1, \bar{\theta}_2, ..., \bar{\theta}_p)$ where $\bar{\theta}_i$ is the mean of part i. A useful approach in studying the problem is to explore the mean-partition polytope M^Π.
When f is quasi-convex, there exists an optimal partition π^* with $\bar{\theta}_\pi$ being a vertex of the mean-partition polytope M^Π. In such a case, it is useful to study M^Π, in particular, to identify properties of partitions π for which $\bar{\theta}_\pi$ is a vertex of the mean-partition polytope. In Sec 4.1, we will make a linear transformation of the mean-partition polytope to the sum-partition polytope, thus allowing the transformation of results from the latter to the former. Unfortunately, this linear transformation technique can not be extended to the bounded-shape problem since we cannot identify the linear transformation. We also explore the approache introduced in Sec. 1.3 for the sum-partition problem to construct mean-partition polytopes. Note that this approach works depending on two things: $(i)H^\lambda \subseteq P \subseteq C^\lambda$ and $(ii)\lambda$ is supermodular. We will study the two issues separately for the single-shape mean-partition problem. In particular, we will shaw that (i) is not satisfied but (ii) is. Thus we cannot conclude $H^\lambda = P = C^\lambda$. However, the proof of supermodularity is mathematically interesting, and hopefully, accomplishing

this challenging proof may bring some benefit in some unexpected direction in the future.

4.1 Linear transformation of mean-partition problems to sum-partition problems

We observe that the single-shape mean-partition problem with prescribed-shape $(n_1, ..., n_p)$ and objective function given by (1.4.3) coincides with the corresponding sum-partition problem with objective function given by (1.2.4) where f satisfies

$$f(x_1, ..., x_p) = g(\frac{x_1}{n_1}, ..., \frac{x_p}{n_p}) \text{ for } x \in R^p \tag{4.1.1}$$

In particular, properties of optimal solutions for single-shape mean-partition problems are deducible from established properties of optimal solutions of corresponding sum-partition problems. For example, it is known [3] that:

A real number function f is called *quasi-convex* if the maximum over every line segment contained in the domain of f is attained at one of the two endpoints.

Theorem 4.1.1. *When the θ_π's are distinct, every single-shape sum-partition problem with f quasi-convex has at least one consecutive optimal partition.*

This result establish the polynomial solvability of the single-shape sum-partition problem. Now, as a function g is quasi-convex if and only if so is the function f that is defined through (4.1.1), we conclude Theorem 4.1.1 that when g is quasi-convex, each single-shape mean-partition problem has at least one consecutive optimal solution and is solvable in polynomial time. Furthermore, by applying the one-to-one transformation

$$(x_1, ..., x_p) = (\frac{x_1}{n_1}, ..., \frac{x_p}{n_p}) \tag{4.1.2}$$

we see that the single-shape mean-partition polytope is the one-to-one linear image of the corresponding single-shape sum-partition polytope. A virtue of

this transformation is that it preserves vertices.
Let $(n_1, ..., n_p)$ be a vector of positive integers with coordinate-sum n and let Π be the set of partitions with shape $(n_1, ..., n_p)$. We observe that for every partition $\pi \in \Pi$, $\bar{\theta}_\pi = (\frac{\theta_{\pi_1}}{n_1}, ..., \frac{\theta_{\pi_p}}{n_p})$, and therefore

$$\begin{aligned} M^\Pi &= \text{conv}\{\bar{\theta}_\pi : \pi \in \Pi\} = \text{conv}\{(\frac{\theta_{\pi_1}}{n_1}, \ldots, \frac{\theta_{\pi_p}}{n_p}) : \pi \in \Pi\} \\ &= \{(\frac{x_1}{n_1}, \ldots, \frac{x_p}{n_p}) : (x_1, \ldots, x_p) \in \text{conv}\{\theta_\pi : \pi \in \Pi\} = P^\Pi\} \\ &= \{(y_1, \ldots, y_p) : (n_1 y_1, \ldots, n_p y_p) \in P^\Pi\}. \end{aligned}$$

Using the representation of P^Π through (1.3.1) we get the representation of M^Π as the set of vectors $y \in R^p$ that satisfy

$$\sum_{i \in I} n_i y_i \geq \lambda(I) \text{ for all } I \subseteq \{1, ..., p\} \text{ and } \sum_{i=1}^{p} n_i y_i = \lambda(\{1, ..., p\}). \quad (4.1.3)$$

Thus we have Theorem 4.1.2.

Theorem 4.1.2. *When the θ_π's are distinct, every single-shape mean-partition problem with g quasi-convex has at least one consecutive optimal partition.*

The linear transformation approach does not apply to the bounded-shape mean-partition problem, since the variation in shape prevent the transformation form being linear as in (4.1.2). Consequently, vertices are not preserved in this nonlinear transformation. Example 13 shows that a partition which is not a vertex of bounded-shape sum-partition polytope becomes a vertex of bounded-shape mean-partition polytope.

Example 13. Let $n = 4, \theta_i = i$, for $i = 1, ..., 4, p = 2$, $U = (2, 3), L = (1, 2)$. Then the sum-partition polytope is the line-segment connecting $(1, 9)$ and $(7, 3)$, the mean-partition polytope is the parallelogram with vertices $\{(1, 3), (4, 2), (1.5, 3.5), (3.5, 1.5)\}$. A partition $\pi = (\{1, 2\}, \{3, 4\})$ is not a vertex of the sum-partition polytope but is a vertex of the mean-partition polytope.

Although we cannot use the linear transformation approach to obtain the bounded shape mean-partition polytope, we still have the following result.

Theorem 4.1.3. *When the θ_π's are distinct and g is quasi-convex, each constrained-shape mean-partition problem has a consecutive optimal partition.*

Proof. An optimal mean-partition must have a shape. Theorem 4.1.3 now follows from Theorem 4.1.2. □

Anily and Federgruen [1] studied the bounded-shape mean-partition problem under the objective function $f(\pi) = \sum_{i=1}^{p} h(\bar{\theta}_\pi, n_i)$. They proved that if for each n_i, $h(x, n_i)$ is convex and nondecreasing in x, then there exists a disjoint optimal partition. Their result follows form Theorem 4.1.3 when the objective function $f(\pi)$ as a special type of quasi-convex function. We note that with stronger assumptions on $h(x, y)$, Anily and Federgruen obtained additional, tighter, results which are not available from our approach.

4.2 Supermodularity of λ_M

In this section, we explore a direct approach, along the line of Sec. 1.3 to construct the single-shape mean partition polytope. Without loss of generality, we assume that $n_1 \le n_2 \le ... \le n_p$.

For $I = \{i_1, i_2, ..., i_k\} \subseteq \{1, ..., p\}$, we suppose that $i_i < i_2 < ... < i_k$. Define $N_{i_k} = \sum_{x=1}^{k} n_{i_x}$ for $1 \le k \le |I|$. Set

$$\lambda_M(I) = \sum_{k=1}^{|I|} \Big(\sum_{j=N_{i_{k-1}}+1}^{N_{i_k}} \theta_j / n_{i_k} \Big). \qquad (4.2.1)$$

Example 14. Let $n = 3, \theta_i = i,$ for $i = 1, 2, 3, p = 2$ and consider the mean partition problem corresponding to the set Π of partitions with shape $(1, 2)$. The set Π contains the three partitions the three partitions $(\{1\}, \{2, 3\})$, $(\{2\}, \{1, 3\})$ and $(\{3\}, \{1, 2\})$ whose corresponding vectors are, respectively,

$(1, 2.5), (2, 2)$ and $(3, 1.5)$. The mean-partition polytope M^{Π} is then the line-segment connecting $(1, 2.5)$ and $(3, 1.5)$. Also, we have that $\lambda_M(\{1\}) = \frac{1}{1} = 1, \lambda_M(\{2\}) = \frac{1+2}{2} = 1.5$ and $\lambda_M(\{1, 2\}) = \min\{\frac{1}{1} + \frac{2+3}{2} = 3.5, \frac{2}{1} + \frac{1+3}{2} = 4, \frac{3}{1} + \frac{1+2}{2} = 4.5\} = 3.5$. So, C^{λ_M} is the polytope defined by the inequalities $x_1 \geq 1, x_2 \geq 1.5, x_1 + x_2 = 3.5$, that is, it is the line-segment connecting $(1, 2.5)$ and $(2, 1.5)$. Finally, the two permutations $(1, 2)$ and $(2, 1)$ of $\{1, 2\}$ correspond, respectively, to the vectors $(\lambda_M)_{(1,2)} = (\lambda_M(\{1\}), \lambda_M(\{1, 2\}) - \lambda_M(\{1\}) = (1, 2.5)$ and $(\lambda_M)_{(2,1)} = (\lambda_M(\{1, 2\}) - \lambda_M(\{2\}), \lambda_M(\{2\})) = (2, 1.5)$, and H^{λ_M} is the line-segment connecting these points.

Example 14 explains that $(i) H^{\lambda_M} \subseteq M^{\Pi} \subseteq C^{\lambda_M}$ isn't satisfied. Now we show that $(ii) \lambda_M$ is supermodular. We first prove

Lemma 4.2.1. *For any shape partition* $\pi = (\pi_1, ..., \pi_p)$, $\sum_{i \in I} \overline{\theta}_{\pi_i} \geq \lambda_M(I)$.

Proof. Define $A = \{\theta_j : j \in \pi_i, i \in I\}$ and $B = \{\theta_1, ..., \theta_{N_{i_{|I|}}}\}$. Suppose $\lambda_M(I)$ is defined on A but $A \neq B$. Then we can reduce $\sum_{i \in I} \overline{\theta}_{\pi_i}$ by replacing any $\theta_j \in A \setminus B$ with a $\theta_k \in B \setminus A$. Therefore we assume $A = B$. Note that

$$\overline{\theta}_{\pi_i} = \sum_{j \in \pi_i} \theta_j (1/n_i), \tag{4.2.2}$$

and $\theta_1, ..., \theta_{N_{i_{|I|}}}$ are ordered from small to large. In $\lambda_M(I)$, the sequence of the multipliers for the θ_j's is

$$\underbrace{\frac{1}{n_{i_1}}, ..., \frac{1}{n_{i_1}}}_{n_{i_1}}, \underbrace{\frac{1}{n_{i_2}}, ..., \frac{1}{n_{i_2}}}_{n_{i_2}}, ..., \underbrace{\frac{1}{n_{i_{|I|}}}, ..., \frac{1}{n_{i_{|I|}}}}_{n_{i_{|I|}}},$$

which are ordered from large to small. Since for any π, $\sum_{i \in I} \overline{\theta}_{\pi_i}$ is computed by multiplying the same set of θ_j's with the same set of multipliers, except in different parings, $\lambda_M(I)$ achieves the minimum by pairing reversely. □

Define $\Delta_I(\pi) = \lambda_M(I) - \lambda_M(I \setminus \{i_1\})$.

Lemma 4.2.2. *Suppose* $I \subset J$ *and* $i_1 = j_1$. *Then* $\Delta_I(\pi) \leq \Delta_J(\pi)$.

Proof. First assume $n_{j_1} = 1$

$$J : \overbrace{\theta_1}^{\pi_{j_1}}, \overbrace{\theta_2, ..., \theta_{n_{j_2}}, \theta_{n_{j_2}+1}}^{\pi_{j_2}}, \overbrace{\theta_{n_{j_2}+2}, ..., \theta_{n_{j_2}+n_{j_3}}, \theta_{n_{j_2}+n_{j_3}+1}}^{\pi_{j_3}}, ...$$

$$J' : \underbrace{\theta_1, \theta_2, ..., \theta_{n_{j_2}}}_{\pi'_{j_2}}, \underbrace{\theta_{n_{j_2}+1}, \theta_{n_{j_2}+2}, ..., \theta_{n_{j_2}+n_{j_3}}}_{\pi'_{j_3}}, \theta_{n_{j_2}+n_{j_3}+1}, ...$$

Figure 2.$\pi(J)$ and $\pi'(J')$

Let π' represent the corresponding partition on $J' = J \setminus \{j_1\}$. We use the same subscript j_k to remind the reader that $n_{j_k} = n'_{j_k}$ for all $2 \leq k \leq |J|$. Figure 2 illustrates $\pi(J)$ and $\pi'(J')$. Note that the components of $\overline{\theta}_{\pi_{j_k}}$ (as in the representation (4.2.2)) cancels with the components in $\overline{\theta}_{\pi'_{j_k}}$ except the first one in $\overline{\theta}_{\pi_{j_k}}$ and the last one in $\overline{\theta}_{\pi'_{j_k}}$. Hence

$$\overline{\theta}_{\pi_{j_k}} - \overline{\theta}_{\pi'_{j_k}} = (\theta_{N_{j_k}} - \theta_{N_{j_{k-1}}})/n_{j_k} \text{ for } 1 \leq k \leq |J|.$$

Consequently,

$$\Delta_J(\pi) = \sum_{k=1}^{|J|} \frac{\theta_{N_{j_k}} - \theta_{N_{j_{k-1}}}}{n_{j_k}}.$$

Similarly,

$$\Delta_I(\pi) = \sum_{k=1}^{|I|} \frac{\theta_{N_{i_k}} - \theta_{N_{i_{k-1}}}}{n_{i_k}}.$$

Suppose $i_k = j_{g(k)}$ with $k \leq g(k)$, $2 \leq k \leq |I|$. Then

$$G_k(J) \equiv \sum_{h=g(k-1)+1}^{g(k)} \frac{\theta_{N_{j_h}} - \theta_{N_{j_{h-1}}}}{n_{j_h}} \geq \sum_{h=g(k-1)+1}^{g(k)} \frac{\theta_{N_{j_h}} - \theta_{N_{j_{h-1}}}}{n_{j_{g(k)}}} = \frac{\theta_{N_{j_{g(k)}}} - \theta_{N_{j_{g(k-1)}}}}{n_{j_{g(k)}}}. \tag{4.2.3}$$

Note that

$$\Delta_J(\pi) - \Delta_I(\pi) \geq \sum_{x=1}^{|I|} [G_x(J) - \frac{(\theta_{N_{i_x}} - \theta_{N_{i_{x-1}}})}{n_{i_x}}].$$

We prove for all $1 \leq k \leq |I|$,

$$\sum_{x=1}^{k} [G_x(J) - \frac{(\theta_{N_{i_x}} - \theta_{N_{i_{x-1}}})}{n_{i_x}}] \geq \frac{(\theta_{N_{j_{g(k)}}} - \theta_{N_{i_k}})}{n_{i_k}},$$

by induction on k. For $k = 1$

$$G_1(J) - \frac{(\theta_{N_{i_1}} - \theta_{N_{i_0}})}{n_{i_1}} = \frac{(\theta_{N_{j_1}} - \theta_{N_{j_0}})}{n_{j_1}} - \frac{(\theta_{N_{i_1}} - \theta_{N_{i_0}})}{n_{i_1}} = 0$$

since $j_1 = i_1, N_{i_1} = n_{i_1} = n_{j_1} = N_{j_1} = 1, \theta_{N_{j_0}} = \theta_{N_{i_0}} = 0$. For general $k > 1$,

$$\sum_{x=1}^{k}[G_x(J) - \frac{(\theta_{N_{i_x}} - \theta_{N_{i_{x-1}}})}{n_{i_x}}] \geq G_k(J) - \frac{(\theta_{N_{i_k}} - \theta_{N_{i_{k-1}}})}{n_{i_k}} + \frac{(\theta_{N_{j_{g(k-1)}}} - \theta_{N_{i_{k-1}}})}{n_{i_{k-1}}}$$

$$\geq \frac{(\theta_{N_{j_{g(k)}}} - \theta_{N_{j_{g(k-1)}}})}{n_{j_{g(k)}}} - \frac{(\theta_{N_{i_k}} - \theta_{N_{i_{k-1}}})}{n_{i_k}} + \frac{(\theta_{N_{j_{g(k-1)}}} - \theta_{N_{i_{k-1}}})}{n_{i_k}} = \frac{(\theta_{N_{j_{g(k)}}} - \theta_{N_{i_k}})}{n_{i_k}},$$

since $n_{j_{g(k)}} = n_{i_k} \geq n_{i_{k-1}}$. Lemma 4.2.2 is proved.

For $n_j > 1$, we can handle in two ways. The first way is to notice that the only difference from the $n_{j_1} = 1$ case is that π_{j_k} and π'_{j_k} would miss each other out in n_{j_1} elements instead of 1 in Figure 2.1. So the numerator of (4.2.3) would be a difference between two n_{j_k}-sums; but the same logic applies. The second way is to notice that $\overline{\theta}_{n_{j_1}}$ gets cancelled out in $\Delta_J(\pi) - \Delta_I(\pi)$. So the scenario is to compare the impact on I and J when both moves back n_{j_1} elements. But this is equivalent to moving one element back n_{j_1} times. □

Finally, we are ready to prove the main result of this section.

Theorem 4.2.3. *λ_M as defined in (4.2.1) is supermodular.*

Proof. Let I and J, be two subsets of $\{1, ..., p\}$. Without loss of generality, assume $I \cup J = \{1, 2, ..., m\}$. We prove Theorem 4.2.3 by induction on m. Theorem 4.2.3 is trivially true for $m = 1$. We prove the general $m \geq 2$ case.
Case(1) $1 \in I \cap J$, i.e. both I and J contain 1. Delete π_1 and the θ_j's in it. Suppose $n_1 = k$. Then the reduced partition problem is to partition the set $\{\theta_{k+1}, ..., \theta_n\}$ into $p - 1$ parts. Theorem 4.2.3 follows by induction.
Case(2) $1 \notin I \cap J$. Without loss of generality, assume $1 \in I$. Let $J^* = J \cup \{1\}$. By case(1),

$$\begin{aligned} 0 &\leq \lambda_M(I \cup J^*) + \lambda_M(I \cap J^*) - \lambda_M(I) - \lambda_M(J^*) \\ &= [\lambda_M(I \cup J^*) - \lambda_M(I)] + [\lambda_M(I \cap J^*) - \lambda_M(J^*)] \\ &\leq [\lambda_M(I \cup J) - \lambda_M(I)] + [\lambda_M(I \cap J) - \lambda_M(J)]. \end{aligned}$$

Since the first difference is unchanged, and the second becomes larger by Lemma 4.2.2 , i.e, $\lambda_M(I\cap J^*)-\lambda_M(I\cap J)=\Delta_{I\cap J^*}(\pi)\leq\Delta_{J^*}(\pi)=\lambda_M(J^*)-\lambda_M(J)$. $\square$

4.3 Some new results in the mean-partition problem

Given vectors a and b in R^p, we say that *a weakly submajorizes b*, written $a\ _w\succ b$ if

$$\sum_{i=1}^{k} a_{[i]} \geq \sum_{i=1}^{k} b_{[i]} \quad \text{for} \quad k=1,\dots,p \tag{4.3.1}$$

It is also well known [17]:

Theorem 4.3.1. *Suppose f is Schur convex nondecreasing and $a\ _w\succ b$. Then $f(a)\geq f(b)$.*

Lemma 4.3.2. *Suppose $(\bar{\theta}_{\pi_1^*},...,\bar{\theta}_{\pi_p^*})$ is the mean vector of the reverse size-consecutive partition, π^*, and let $(\bar{\theta}_{\pi_1},...,\bar{\theta}_{\pi_p})$ denote the mean vector of an arbitrary partition π with the shape is equivalent to the shape of π^*. Then $(\bar{\theta}_{\pi_1^*},...,\bar{\theta}_{\pi_p^*})\ _w\succ(\bar{\theta}_{\pi_1},...,\bar{\theta}_{\pi_p})$.*

Proof. It was proved in [5] that reverse size-consecutive is a 2-shape-sortable property, namely, it suffices to prove Lemma 4.3.2 by assuming $p=2$. Define W to be the set consisting of $\frac{1}{n_1}$, n_1 of them, and $\frac{1}{n_2}$, n_2 of them. In the sum $\bar{\theta}_{\pi_1}+\bar{\theta}_{\pi_2}$, each $\theta_j\in\pi_1$ contributes $\frac{\theta_j}{n_1}$ and each $\theta_j\in\pi_2$ contributes $\frac{\theta_j}{n_2}$. Therefore $\bar{\theta}_{\pi_1}+\bar{\theta}_{\pi_2}$ is determined by a ont-to-one mapping between W and the set of n_1+n_2 θ's. By the Hardy, Littlewood and Polya theorem, the sum is maximized when the mapping is monotone, larger element in W mapped to larger θ, which implies the reverse size-consecutive partition achieves the maximum sum.
Next, we prove that

$$\max\{\bar{\theta}_{\pi_1^*},\bar{\theta}_{\pi_2^*}\}\geq\max\{\bar{\theta}_{\pi_1},\bar{\theta}_{\pi_2}\}.$$

Without loss of generality, assume $n_1 \leq n_2$. It is trivial that $\bar{\theta}_{\pi_1^*} \geq \bar{\theta}_{\pi_1}$. Let π_2' consist of the n_2 largest θ's. Then clearly,

$$\bar{\theta}_{\pi_2} \leq \bar{\theta}_{\pi_2'}$$

and the average of the n_1 largest θ's is larger than the average of the n_2 largest θ's, that means

$$\bar{\theta}_{\pi_2'} \leq \bar{\theta}_{\pi_1^*}.$$

So,

$$\max\{\bar{\theta}_{\pi_1^*}, \bar{\theta}_{\pi_2^*}\} = \bar{\theta}_{\pi_1^*} \geq \max\{\bar{\theta}_{\pi_1}, \bar{\theta}_{\pi_2}\}.$$

□

Using Theorem 4.3.1 and Lemma 4.3.2, we obtain

Theorem 4.3.3. *There exists a reverse size-consecutive optimal partition for the single-shape mean partition problem.*

Corollary 4.3.4. *There exists a reverse size-consecutive optimal partition for the constrained-shape mean partition problem.*

Note that for a given shape, the size-consecutive partition is unique. So for the mean-partition problem with the constrained-shape set Γ, we only need to compare the f-values of $|\Gamma|$ partitions, one from each shape in $|\Gamma|$. For bounded-shape partitions, $|\Gamma|$ is not explicit. It suffices to consider only those shapes in $|\Gamma|$ which is not majorized by any other shape in Γ. Further, in Chapter 3, we bounded the number of these nonmajorized shape by 2^{p-1}(Sec. 3.3).

Although we don't know how to characterize the constrained-shape mean partition polytope, we can bound its number of vertices by the sum of the number of vertices on the single-shape mean-partition polytope for each shape in Γ. Since there is a one-to-one mapping between the vertices of the single-shape mean-partition polytope and the vertices of the single-shape sum-partition polytope, and also a one-to-one mapping is well known [18] between the latter and the set of consecutive partitions, we obtain a bound of $|\Gamma|p!$. This is indeed an upper bound as the following example shows that a consecutive partition of a shape in Γ is not a vertex of the constrained-shape polytope.

Example 15. Let $\Gamma = \{(1,3),(2,2),(3,1)\}$ $n = 4, \theta_i = i$ for $i = 1...4, p = 2$. We give the two points generated by the two consecutive partitions for each shape:

shape	consecutive partitions
$(1,3)$	$(1,3)$ $(4,2)$
$(2,2)$	$(\frac{3}{2},\frac{7}{2})$ $(\frac{7}{2},\frac{3}{2})$
$(3,1)$	$(3,1)$ $(2,4)$

Thus the polytope has 4 vertices $(1,3)(4,2)(3,1)(2,4)$ while the two points yielded by shape $(2,2)$ are internal.

Theorem 4.3.5. *Suppose f is quasi-convex. Then there exists a consecutive optimal partition in a set of cardinality at most $|\Gamma|p!$ for the constrained-shape mean-partition problem with set Γ.*

Chapter 5

Conclusion and remarks

In this thesis, we develop the generation function approach to count the number of bounded-shape partitions, which helps us to estimate the practicability of the brute-force method to find an optimal partition. We extend the concept of majorizing shape to the concept of nonmajorized shape for bounded-shape sum-partition problem with Schur convex objective function, we prove that there exists a nonmajorized shape for which the corresponding size-consecutive partition is optimal. Moreover, we prove 2^{p-1} is an upper bound of nonmajorized shape-types, and develop algorithms to find all nonmajorized shapes(shape-types). In the last chapter, we research the mean-partition problem. We use the linear transformation approach to characterize the single-shape mean-partition polytope and prove that if the objective function is quasi-convex, then there exist a consecutive optimal partition. We also give a bound of the cardinality of the candidate set to find optimal partition for constrained-shape mean-partition case.

We list some topics for future research:

(i) to find a more explicit formula to count the number of bounded-shape partitions,

(ii) to give the exactly value of $f(p)$,

(iii) prove our $\binom{p-1}{\lfloor (p-1)/2 \rfloor}$ conjecture,

(iv) develop the faster algorithm to find all nonmajorized shapes(shape-types),

(v) characterize the bounded-shape mean-partition polytope.

References

[1] S. Anily and A. Federgruen, Structured partition problems, *Oper. Res.* 39 (1991), 130–149.

[2] R. A. Brualdi, Introductory Combinatorics, 3nd ed., Prentice Hall, 1999, Chapter 8.

[3] E. R. Barnes, A. J. Hoffman and U. G. Rothblum, Optimal partitions having disjoint convex and conic hulls, *Mathematical Programming*: Series A, 54 (1992) 69-86.

[4] F. H. Chang, H. B. Chen, J. Y. Guo, F. K. Hwang and U. G. Rothblum, One-dimensional optimal bounded-shape partitions for Schur convex sum objective functions, *to appear*.

[5] G. J. Chang, F. L. Chen, L. L. Hwang, F. K. Hwang, S. T. Nuan, U. G. Rothblum, I-Fan Sun, J. W. Wang, and H. G. Yen, Sortabilities of partition properties, *Journal of Combinatorial Optimization* 2 (1999) 413-427.

[6] F. H. Chang, J. Y. Guo, F. K. Hwang and Y. C. Pan, A generating function approach to count the number of bounded-shape partitions, *to appear*.

[7] F. H. Chang and F. K. Hwang, Supermodularity in mean-partition problems, *Journal of Global Optimization*.

[8] F. H. Chang, F. K. Hwang and U. G. Rothblum, The mean-partition problem, *preprint.*

[9] B. Gao, F. K. Hwang, W. W.-C. Li and U. G. Rothblum, Partition polytopes over 1-dimensional points, *Math. Program.* 85 (1999) 335–362.

[10] F. K. Hwang, M. M. Liao and C. Y. Chen, Supermodularity of various partition problems, *J. Global Optimization* 18 (2000) 275–282.

[11] F. K. Hwang, J. S. Lee and U. G. Rothblum, Permutation polytopes corresponding to strongly supermodular functions, *Disc. Appl. Math.* 142 (2004) 52–97.

[12] F. K. Hwang, S. Onn and U. G. Rothblum, Representations and characterizations of vertices of bounded-shape partition polytopes, *Linear Algebra and its Applications*, 278 (1998) 263-284.

[13] F. K. Hwang, S. Onn and U. G. Rothblum, Explicit solution of partition problems over a 1-dimensional parameter space, *Naval Research Logistics*, 47 (2000) 531-540.

[14] F. K. Hwang and U. G. Rothblum, Directional-quasi-convexity, asymmetric Schur-convexity and optionality of consecutive partitions, *Math. Oper. Res.* 21 (1996) 540–554.

[15] F. K. Hwang and U. G. Rothblum, Partition-optimization with Schur-convex sum objective functions, *SIAM J. Disc. Math.*, to appear.

[16] F. K. Hwang and U. G. Rothblum, Partition: Optimality and clustering, World Scientific, Singapore, to appear.

[17] A. W. Marshall and I. Olkin, Inequalties, Theory of majorization and its applications, *Academic Press, New York*, 1979.

[18] L. S. Shapely, Cores of convex gormes, *Intern. J. Game Theory* 1 (1971) 11–29.

[19] E. Sperner, Ein Satz über Untermengen einer endlichen Menge, *Mathematische Zeitschrift*. 27 (1928) 544-548.