Chapter 1

Formulation and Examples

The first chapter is devoted to the introduction of a framework of reference for partition problems. In particular, partition problems are classified by three main characteristics: (i) the set of partitions over which an optimization problem takes place, (ii) the number of characteristics associated with each of the partitioned elements, and (iii) the objective function that is to be optimized. Further, 15 examples from diverse areas are introduced to demonstrate the expressive power of partition problems. Some of these examples are known as NP-hard, implying that the development of efficient solution methods is unlikely. Still, we show in Chapter 12 how the theory we develop can be used to solve most of these examples rather efficiently. Some of the examples we mention have broad modelling potential that is useful to describe complicated situations. The description of these examples in the current chapter focuses on basic scopes of the models. More general or more complicated variants of the models will be provided in Chapter 12 (along with the corresponding solution methods).

1.1 Formulation and Classification of Partitions

Consider a finite set $\mathcal{N}$ of distinct positive integers (for most of our development $\mathcal{N} = \{1, \ldots, |\mathcal{N}|\}$). A *partition of* $\mathcal{N}$ is a finite collection of sets $\pi = (\pi_1, \ldots, \pi_p)$ where $\pi_1, \ldots, \pi_p$ are pairwise disjoint nonempty sets whose union is $\mathcal{N}$. In this case we refer to p as the *size of* π, and to the sets $\pi_1, \ldots, \pi_p$ as the *parts of* π. Further, if $n_1, \ldots, n_p$ are the sizes of $\pi_1, \ldots, \pi_p$, respectively, we define the *shape of* π as the vector $(n_1, \ldots, n_p)$; of course, in this case $\sum_{j=1}^{p} n_j = |\mathcal{N}|$. We sometimes add prefix "p-" or "$(n_1, \ldots, n_p)$-" to explicitly express the size or shape of a partition, referring to a *p-partition of* $\mathcal{N}$ or to an $(n_1, \ldots, n_p)$*-partition of* $\mathcal{N}$. Further, for brevity, we frequently omit the reference to the set $\mathcal{N}$ as the partitioned set, and simply refer to *partitions*. In our development, we sometimes require that partitions' parts are nonempty while at other times this requirement is relaxed.

At times, we restrict attention to the set of all partitions or to the set of all partitions whose size or shape satisfies prescribed restrictions; we refer, respectively, to *open*, *constrained-size* and *constrained-shape sets of partitions*. If the restrictions on the size or shape are expressed by prescribing a single element, then we refer to as *single-size* or *single-shape sets of partitions* and if the restrictions are in terms of lower and upper bounds on the size or shape we refer to as *bounded-size* or *bounded-shape sets of partitions*, respectively. Note that all of the above classes of sets of partitions can be treated as special cases of a constrained-shape class. Their respective names simply emphasize the kind of constraints on the shapes. For example, the class of p-partitions collects those partitions with p (nonempty) parts, and the class of open partitions collects all partitions without restriction on the number of (nonempty) parts. Thus, we will treat the constrained-shape class as the most general class. Still, the general framework of constrained-shape sets of partitions does not appear in the forthcoming development and whenever constrained-shape partition problems are mentioned, all shapes have the same size; consequently, we shall use the terminology "constrained shape" for set of partitions with restricted shapes that have the same size (though formally, these are *single-size constrained-shape sets of partitions*). Sometimes, we casually use the above adjectives that describe sets of partitions to partitions that belong to the corresponding given sets.

Let the (partitioned) set $\mathcal{N}$ be given and let $n \equiv |\mathcal{N}|$. When a single-size set of partitions is considered, the prescribed single-size is given as a positive integer p. Given a positive integer p and a set Γ of positive integer-vectors $(n_1, \ldots, n_p)$, each satisfying $\sum_{j=1}^{p} n_j = n$, let Π^Γ be the corresponding constrained shape partitions, that is, all partitions with shape in Γ. In particular, if Γ consists of a single vector $(n_1, \ldots, n_p)$, we use the notation $\Pi^{(n_1,\ldots,n_p)}$ for (the single-shape set of partitions) Π^Γ. Also, if L and U are nonnegative integer p-vectors satisfying $L \le U$ and $\sum_{j=1}^{p} L_j \le n \le \sum_{j=1}^{p} U_j$, we let $\Gamma^{(L,U)}$ denote the set of nonnegative integer-vectors $(n_1, \ldots, n_p)$ satisfying $L_j \le n_j \le U_j$ for $j = 1, \ldots, p$; in this case, we use the notation $\Pi^{(L,U)}$ for (the bounded-shape set of partitions) $\Pi^{\Gamma^{(L,U)}}$. (The restrictions on L and U assure that $\Gamma^{(L,U)}$ and $\Pi^{(L,U)}$ are nonempty.) Note that single-size and single-shape sets of partitions are instances of bounded-shape sets obtained, respectively, by setting $L_j = 1$ and $U_j = n$ for all j or $L_j = U_j = n_j$ for all j. Similarly, open and single-size sets partitions are instances of bounded-size sets. The hierarchy of the classification of partitions is summarized in Table 1.1.1.

Table 1.1.1: Classification of sets of partitions

open	constrained-size	constrained-shape (size given)
	bounded-size	bounded-shape (size given)
	single-size	single-shape

A *multiset* is a group of elements where each element is allowed to have multiple occurrence. The formal notation of a multiset has double brackets, e.g., $\{\{1, 1, 2, 2, 3\}\}$, or is given as a bracketed list of distinct elements with superscripts designating their duplications, e.g., $\{1^2, 2^2, 3\}$. However, at times, we abuse notation and use single brackets, e.g., $\{1, 1, 2, 2, 3\}$.

It is implicitly assumed in the above definitions that the parts of partitions are distinguishable. But, in some applications the parts are indistinguishable and can be permuted without any restrictions. Thus, we also consider unlabeled partitions. Specifically, an *unlabeled partition of* $\mathcal{N}$ is a finite collection of sets $\pi = \{\pi_1, \ldots, \pi_p\}$ where the π_j's are as above. Again, we refer to p and to the sets $\pi_1, \ldots, \pi_p$ as the *size* and the *parts of* π, respectively. Further, if $n_1, \ldots, n_p$ are the sizes of $\pi_1, \ldots, \pi_p$, respectively, we define the *shape of* π as the multiset $\{\{n_1, \ldots, n_p\}\}$; again, we must have that $\sum_{j=1}^{p} n_j = |\mathcal{N}|$. In the literature, unlabeled partitions are commonly referred to as *allocations*. While we reserve the term "partitions" for labeled ones, we sometimes refer to *labeled partitions* (when potential ambiguity may arise).

We apply the same classification to sets of unlabeled partitions as we do to sets of labeled ones; see Table 1.1.1. Single-shape and bounded-shape sets of unlabeled partitions are defined, respectively, by a multiset $\{\{n_1, \ldots, n_p\}\}$ or a multiset of pairs $\{\{(L_1, U_1), \ldots, (L_p, U_p)\}\}$; the specification or the bounds on the sizes of the parts of partitions then hold for some labeling of the parts. Frequently, as parts of unlabeled partitions are indistinguishable, sizes and bounds of unlabeled partitions are uniform, namely all parts have the same size and a multiset of bounds $\{\{(L_1, U_1), \ldots, (L_p, U_p)\}\}$ consists of p identical pairs.

1.2 Formulation and Classification of Partition Problems over Parameter Spaces

In this section, we introduce the framework for partition problems over parameter spaces which are the main goal of this book. Specifically, a *partition problem* concerns the selection of a partition π out of a given set Π of partitions so as to optimize (that is, minimize or maximize) an objective function F that is defined over Π.

We assume throughout that each element i of the partitioned set $\mathcal{N}$ is *associated* with a vector $A^i \in \mathbb{R}^d$ where d is a fixed positive integer (independent of i); we refer to the coordinates of A^i as *parameters* or *characteristics* associated with i. The vectors $A^1, \ldots, A^n$ are part of the data of the problem and are given in the form of a real $d \times n$ matrix A. For a subset S of $\mathcal{N} = \{1, \ldots, n\}$, A^S is the submatrix of A consisting of the columns of A indexed by S, ordered as in A. Also, we use "bars" over matrices, to denote the multiset consisting of their columns, for example, a subset S of $\mathcal{N} = \{1, \ldots, n\}$, $\overline{A^S}$ is the set of columns of A^S, accounting for multiplicities.

An *objective function* $F(\cdot)$ that is to be maximized (or a cost function that is to be minimized). It associates a value $F(\pi)$ to each (feasible) p-partition π and this value depends on the parameters of the elements that are assigned to each part. In the most general case we consider, for each positive integer v, a column-symmetric function $h_v : \mathbb{R}^{d\times v} \to \mathbb{R}^d$, defined over multisets of v d-vectors, functions $g_j : \mathbb{R}^d \to \mathbb{R}^m$, $j = 1, \ldots, p$ and a function $f_p : \mathbb{R}^{d\times p} \to \mathbb{R}^d$. Then the value $F(\pi)$ associated with partition π having shape $(n_1, \ldots, n_p)$ is given by

$$F(\pi) = f_p\left(g_1[h_{n_1}(\overline{A^{\pi_1}})], \ldots, g_p[h_{n_p}(\overline{A^{\pi_p}})]\right) \ . \tag{1.2.1}$$

The functions h_{n_j} can, in fact, depend on the location within the variables of f_p, that is, on the index j; also, the functions g_j may depend on n_j. In many common applications, each of the functions h_{n_j} is the summation function, in which we refer to the corresponding problems as *sum-partition problems*. When h_v or g_j is independent of the indexing parameter, we drop the index. Also, when referring to partitions of common size we drop the subscript "p" of f_p. We call f_p *additive* if f_p is the sum function. It is also possible to consider partition problems where the domain of the functions h_{n_j} consists of ordered lists (and the functions h_{n_j} are not symmetric).

For a more concise form of sum-partition problems, we introduce some notation. For a $d \times n$ real matrix A and a p-partition $\pi = (\pi_1, \ldots, \pi_p)$ of

$\mathcal{N}$, we define the *π-summation-matrix of A*, denoted A_π, by

$$A_\pi \equiv \left[\sum_{t\in\pi_1} A^t, \ldots, \sum_{t\in\pi_p} A^t\right] \in \mathbb{R}^{d\times p}, \qquad (1.2.2)$$

where the empty sum is defined to be 0 (here and elsewhere in this book). When each of the g_j's is the identity over $\mathbb{R}^d$, the objective function F associates with partition π the value $F(\pi)$ with the representation

$$F(\pi) = f_p(A_\pi) \qquad (1.2.3)$$

(as was already mentioned, when the optimization over partitions concerns only partitions of fixed size p, the dependence of the functions f_p on p is suppressed and we refer only to a real-valued function f on $\mathbb{R}^p$).

Of particular interest is the case where all A^i's have a common coordinate, say the first one, and it equals 1 for each A^i. In this case row 1 of A_π is the shape of π. It follows that (1.2.3) allows for the objective function F to depend on the shape of partitions. Of course, for single-shape problems, the part-sizes (that is, the coefficients of the shape) are fixed and can be viewed as parameters of the objective function.

In summary, the three major characteristics by which partition problems are classified are:

(1) The family of partitions Π over which the function F (with representation as in (1.2.1) or (1.2.3)) is considered and optimized: Using the classification of families of partitions provided in Table 1.1.1, we shall refer to *open, constrained-size, bounded-size, single-size, constrained-shape, bounded-shape, and single-shape partition problems*; of course, there is a natural hierarchy of this characteristic: single-shape and single-size are instances of bounded-shape which is an instance of constrained-shape. In addition, we refer to *relaxed-size problems* as single-size problems which allow for empty parts. Further, the description of the set Π of (feasible) partitions has to specify whether empty parts are allowed or are prohibited.

(2) The number of parameters associated with each of the partitioned elements: We shall refer to *single-parameter problems*, *two-parameter problems* and *multi-parameter problems*.

(3) The objective (cost) function F as expressed by (1.2.1): Adjectives like "sum-," "max-" or "mean-" of partition problems reflect properties of h_v while properties of f, like "linear," "convex," "Schur convex" and "separable," reflect properties of f_p; e.g., we refer to sum-partition problems with f Schur convex.

A partition that maximizes/minimizes $F(\cdot)$ over a prescribed family of partitions Π is called *optimal over* Π.

For single-, bounded- and constrained-shape families of partitions, the inclusion or exclusion of empty parts is implicit in the description of the set of feasible shapes – empty parts are prohibited when the set of feasible shapes consists only of positive vectors, whereas the inclusion of nonnegative shapes that are not (strictly) positive indicates that empty parts are allowed. In particular, results that apply to all single-, bounded- or constrained-shape partition problems with empty parts allowed, extend to the corresponding problems with empty parts prohibited (by restricting attention to the corresponding sets of shapes that consist only of positive vectors). Still, there are results in this book that apply to constrained-shape problem that require the exclusion of empty parts. For sets of partitions that are determined by restrictions on size, e.g., the case of single-size, the inclusion or exclusion of empty parts must be made explicitly. As neither variant captures the other – see the results of Section 5.4 (demonstrated in Table 5.4.3) about the optimality of monopolistic partitions for size problems with empty parts allowed and the optimality of extremal partitions for size problems with empty parts prohibited. In different parts of this book we set default positions that allow for empty parts, exclude them or allow either way (to be determined through the specification of the set of feasible shapes). When needed, exceptions are then made to such default positions. See Section 1.3 where the default position is set to exclude empty parts, but a discussion is included of what happens when this restriction is relaxed.

The above classification refers to labeled partitions. We note that optimization problems over sets of unlabeled partitions can be embedded in the above framework, with Π and F being invariant under part-permutation.

A major goal of our development is the identification of properties that are present in optimal partitions, thereby allowing one to restrict attention to the corresponding subclasses of partitions. These properties are usually defined in terms of geometric/algebraic characteristics of the set of vectors associated with the indices assigned to the parts of the underlying partition. In particular, we consider properties that capture "clustering" of these vectors.

Formally, a *property* Q of a partition is a unitary relation over sets of partitions and it can be identified with the sets of partitions that satisfy it. The properties we consider when studying multi-parameter problems depend on the A^i's; as such, it seems natural to consider the partitioning of the A^i's rather than the partitioning of the underlying index-set. But, possible equality among the A^i's often obscures the partition property. This situation is easily handled when the partitions are of the set of distinct indices and not of the A^i's.

We refer to a *type 1 result* about a partition problem if a property is established for every optimal partition; we refer to a *type 2 result* if a property is established for some optimal partitions. The next lemma, observed by Golany, Hwang and Rothblum [2008], records a useful implication for the presence of properties in optimal solutions for single-shape/size and constrained-shape/size partition problems. For simplicity, we restrict attention to partitions whose size p is fixed.

Lemma 1.2.1. *Consider an objective function F over partitions, a set Γ of integer p-vectors whose coordinate-sum is n and a property Q of partitions such that each single-shape partition problem corresponding to a shape in Γ and the objective function F, Q is satisfied by some (every) optimal partition. Then, for each constrained-shape partition problem corresponding to a subset of Γ and the objective function F, Q is satisfied by some (every) optimal partition. Also, the above holds with "shape" replaced by "size".*

Proof. We consider only the "shape" version of the lemma, as the same arguments verify the "size" version. We first establish the result with "some" rather than "every". Consider a partition problem where F is to be maximized over the set of partitions whose shapes must lie in a prescribed subset of Γ. Let π be an optimal partition for this problem. Consider the single-shape partition problem where F is to be maximized over partitions with shape $\langle\pi\rangle$; π is obviously optimal for this problem. By assumption, Q is satisfied by some optimal partition for this single-shape partition problem, say π'. As both π and π' are optimal for the same partition problem, $F(\pi) = F(\pi')$. Now, as $\langle\pi'\rangle = \langle\pi\rangle \in \Gamma$ and $F(\pi) = F(\pi')$, π' is feasible and optimal for the underlying constrained-shape partition problem; so π' satisfies Q and is optimal for that problem. The case with "every" replacing "some" follows from the above arguments and the observation that π must satisfy Q as it is optimal for the corresponding single-shape partition problem. □

Remarks:

(1) Lemma 1.2.1 is an example where allowing or prohibiting empty parts is captured by the description of the set of feasible shapes – if Γ consists only of positive vectors, then empty parts are prohibited and if Γ

includes vectors that have 0 coordinates, then empty parts are allowed.

(2) Lemma 1.2.1 shows that in order to verify properties of optimal solution of constrained-shape/size problems, it suffices to consider the case of single-shape/size. But, from a computation point of view, there may be a difference, e.g., bounded-shape problems may display simpler computational procedures (see Chapter 2 and Section 8.1).

(3) Lemma 1.2.1 is of particular interest when the objective function $F(\cdot)$ over partitions depends on the shape of the underlying partition and the sums of vectors corresponding to elements assigned to its parts. For single-shape problems, the dependence of the objective function on the shape is degenerated (as the shape of all considered partitions is the same and can be viewed as a constant integer vector); but, for bounded- and constrained-shape problems the dependence of the objective function on the shape is not trivial any more (for example, see Lemma 7.3.1 and Example 7.3.1).

For single-parameter problems we usually denote the n elements in the single row of A by $\theta^1, \ldots, \theta^n$. In particular, for single-shape sum-partition problems, the π-summation matrix associated with a partition π is then a p-vector, which we denote θ_π and to which we refer as π-*summation vector*, that is

$$\theta_\pi \equiv \left(\sum_{i \in \pi_1} \theta^i, \ldots, \sum_{i \in \pi_p} \theta^i \right) \in \mathbb{R}^{1 \times p}. \tag{1.2.4}$$

In this special case, the partitioned elements can be renumbered so that θ^i is monotone, that is,

$$\theta^1 \leq \cdots \leq \theta^n \tag{1.2.5}$$

(still, when analyzing the complexity of algorithm that solve partition problems, we account for time required to sorting θ^i's). As observed for single-parameter problems, when $\theta^i = 1$ for each i the vector θ_π is the shape of π.

The next lemma provides a tool for deducing type 2 conclusions from type 1 conclusions in the case of single-parameter problems. Note that type 1 results usually require that the θ^i's are distinct (e.g., if all θ^i's coincide and F satisfies (1.2.1), then all partitions with the same shape have the same F-value and not all can be expected to satisfy Q). Express the dependence of an objective function $F(\cdot)$ over partitions on the parameter vector θ by $F^\theta(\cdot)$. We say that $F(\cdot)$ is *continuous* if for every sequence $\theta(k) \in \mathbb{R}^n$ that converges to $\theta \in \mathbb{R}^n$ and for every partition π, $lim_{k\to\infty}F^{\theta(k)}(\pi) = F^{\theta}(\pi)$. Of course, if F is given by (1.2.1), then F is continuous whenever f_p, the g_j's and h_v's are continuous in the conventional sense.

Lemma 1.2.2. *Assume that $d = 1$, F is continuous and Π is a set of partitions for which it is known that when the θ^i's are distinct, every optimal partition satisfies Q. Then without restrictions on the θ^i's, every partition problem over Π has an optimal partition satisfying Q.*

Proof. Let $\theta \in \mathbb{R}^n$. For each positive ϵ consider a perturbation $\theta(\epsilon)^i \equiv \theta^i + \epsilon^i$ for $i = 1, \dots, n$. Observe that for sufficiently small positive ϵ, the $\theta(\epsilon)^i$'s are distinct; in fact, if $\theta^1 \leq \cdots \leq \theta^n$ it can be assured that $\theta(\epsilon)^1 < \cdots < \theta(\epsilon)^n$. For each such $\epsilon > 0$ there is an $F^{\theta(\epsilon)}$-optimal partition, and by assumption, each such partition satisfies Q. Thus, for all sufficiently small positive ϵ, there exists a partition $\pi^\epsilon \in \Pi^*_{F^{\theta(\epsilon)}}$ that satisfies Q, in particular, $F^{\theta(\epsilon)}(\pi^\epsilon) \geq F^{\theta(\epsilon)}(\pi)$ for each partition $\pi \in \Pi$. As the number of partitions in Π is finite, there is a partition π^* that is uniformly $F^{\theta(\epsilon)}$-optimal for a decreasing sequence of values ϵ that has 0 as its limit point, in particular, π^* satisfies Q. Let $\epsilon_1, \epsilon_2, \dots$ be a corresponding sequence. As $F^{\theta(\epsilon_k)}(\pi^*) \geq F^{\theta(\epsilon_k)}(\pi)$ for each $\pi \in \Pi$, it follows from the continuity assumption that $F^{\theta}(\pi^*) \geq F^{\theta}(\pi)$. So, the partition π^* satisfies Q and is F^{θ}-optimal. □

In the case of multi-parameter problems, having the A^i's distinct is usually not enough for a type 1 result. Still, perturbation techniques that generalize those used in the proof of Lemma 1.2.2 are developed and used in Chapter 9 to deduce type 2 results from type 1 results for multi-parameter problems.

Much of our analysis focuses on partition problems with fixed size (single-shape, bounded-shape, constrained-shape and single-size problems). When considering problems that allow for varying partition-sizes, the objective function is, effectively, parametrized by an integer p representing the partition-sizes, see (1.2.1). In particular, results about optimal partitions will then depend on some structure that expresses a connection between the objective function of p-partitions for different values of p. The most natural structure is the "*reduction assumption*" that asserts that F is invariant under the permutation/elimination of empty parts. For example, this is the case when f_p in (1.2.1) is the sum- (max/min)-function and the value assigned to empty parts is 0 ($-\infty/\infty$). A natural assumption that accompanies the "reduction assumption" is *symmetry* of F under part-permutations.

Open problems that do not satisfy the reduction assumption can be difficult to analyze: they may have infinitely many feasible partitions and there needs not be an optimal one; this situation is demonstrated in the next example.

Example 1.2.1. Let $\mathcal{N} = \{1, 2\}$ and consider the open problem where empty sets are allowed and objective function is given by $F(\pi) = \sum_{j=1}^{p} j(\sum_{i\in\pi_j} i)$ for partitions π of size p. Then every single-size problem will assign both elements of $\mathcal{N}$ to the part with the highest index. But, the open problem does not have an optimal partition as every p-partition is dominated by the best $(p + 1)$-partition. □

When the "reduction assumption" is satisfied, each partition with p nonempty parts can, effectively, be viewed as (one of several) p'-partitions (with empty sets allowed) for any $p' \geq p$. In particular, upper bounded-size problems with upper bound p and open problem can be embedded in single-size problems with, respectively, p or n as the prescribed size (and with empty parts allowed). The amount of computation of solving n-size problems may be high, but the reduction may be useful in establishing properties of optimal solutions. Further, in cases where an optimal solution for a single-size problem with empty parts excluded is simple, optimizing over the number of nonempty parts may be computationally tractable (for example, see Theorem 7.3.10). When the reduction assumption is in effect, allowing for empty parts is a technical notion that helps us address bounded-size and open problems.

A different class of partition problems concerns situations where each partition consists of $t > 1$ parallel partitions, each on a distinct type of elements. In such problems we have p *multiparts* (referred to as *modules* in some applications) and t distinct sets of elements $\mathcal{N}^u$, with $u = 1, \ldots, t$, and we refer to $\mathcal{N}^u$ as the set of elements of *type* u. A *multipartition* then consists of (parallel) partitions of each of the sets $\mathcal{N}^u$ into p parts, say $(\pi_{11}, \ldots, \pi_{1p}), \ldots, (\pi_{t1}, \ldots, \pi_{tp})$ which are partitions of $\mathcal{N}^1, \ldots, \mathcal{N}^t$, respectively. Each multipart j is then assigned the parts $\{\pi_{uj} : u = 1, \ldots, t\}$. This formulation allows for allocating items of distinct nature to a module in complex systems, for example, to perform distinct tasks. Formally, a multipartition is the array $\{\pi_{uj} : j = 1, \ldots, p \text{ and } u = 1, \ldots, t\}$. We refer to the array of integers $\{|\pi_{uj}| : u = 1, \ldots, t,\ j = 1, \ldots, p\}$ as the *multishape* of the given multipartition and to p as its *size*. We can now consider classes of multipartitions that are determined by a single multishape, bounded multishape and constrained multishape and we can consider a variety of objective functions over multipartitions.

In the multipartition problems we consider, the partitioned elements of each type are associated with a single parameter, e.g., element i of type u is associated with θ^{ui}. As for partition problems, we consider an objective function F which associates each multipartition $\pi = \{\pi_{uj} : j = 1, \ldots, p \text{ and } u = 1, \ldots, t\}$ with a real number $F(\pi)$; the goal is to maximize $F(\pi)$ over a given set of multipartitions. A simple structure for such a function F is:

$$F(\pi) = f_p \left(\sum_{u=1}^{t} \sum_{i \in \pi_{u1}} \theta^{ui}, \ldots, \sum_{u=1}^{t} \sum_{i \in \pi_{up}} \theta^{ui} \right)$$

with f_p as a real-valued function on $\mathbb{R}^p$; such F is called a *multi-sum function*. The classification of partitions problems applies to multi-partitions with corresponding modifications.

1.3 Counting Partitions

In this section we count the number of partitions in various classes, following Hwang and Mallows [1995] and Yeh, Liao, Hwang and Chang [1998]. We focus first on the case where empty parts are prohibited, but at the end of the section we obtain corresponding expressions for the number of partitions that allow for empty parts. Further, we explain that in forthcoming counting of partitions that satisfy prescribed properties, it is sufficient to focus on the case where empty parts are prohibited.

For positive integers n and p, let $\#(n,p)$ denote the number of labeled p-partitions of a subset of $\mathcal{N}$ consisting of n elements. Since each element can be assigned to any of the p-parts, we obtain an estimate that there are p^n partitions. However, this expression allows for the parts to be empty. For each $j = 1, \ldots, p$, the number of partitions with part j empty (these are not partitions as per the above formal definition) is $(p-1)^n$. Subtracting $p(p-1)^n$ from p^n, would subtract too many partitions since those partitions with $k > 1$ empty parts are then subtracted k times. The correct number is obtained by using the principle of inclusion and exclusion.

$$\binom{\#(n,p) = \sum_{j=0}^{p-1} (-1)^j \; p}{j(p-j)^n \; .}$$

This number is bounded from below by the number of partitions obtained by assigning any specific p of the n elements to individual parts and assigning the remaining $n-p$ "elements" to the parts, allowing for empty allocations. Note that the p specific elements identify the p parts (as distinct). Hence, there are p^{n-p} ways of assigning the remaining elements yielding a lower bound of p^{n-p} on $\#(n,p)$, and thereby showing that $\#(n,p)$ is exponential in n for each fixed $p > 1$. Let $\overline{\#}(n,p)$ denote the corresponding number of unlabeled p-partitions. Since any part permutation of a labeled p-partition yields the same unlabeled partition,

$$\overline{\#}(n,p) = \frac{\#(n,p)}{p!};$$

$\overline{\#}(n,p)$ are known in the literature as "the Sterling numbers of the second kind" (e.g., Fort [1945], Hall [1986]); they satisfy the recursion $\overline{\#}(n,p) = \overline{\#}(n-1,p-1) + p\#(n-1,p)$.

For positive integer n, let $\#(n)$ denote the number of labeled partitions of a subset of $\mathcal{N}$ consisting of n elements. Then,

$$\#(n) = \sum_{p=1}^{n} \#(n,p) = \sum_{p=1}^{n} \sum_{j=0}^{p-1} (-1)^j \binom{p}{j} (p-j)^n \text{ , and}$$

$$\overline{\#}(n) = \sum_{p=1}^{n} \overline{\#}(n,p).$$

$\overline{\#}(n)$ are known as the *Bell Numbers* in the combinatorics literature. The first ten numbers of the sequence are: 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975. The above lower bound on the $\#(n,p)$ yields a lower bound $\sum_{p=1}^{n} p^{n-p}$. Selecting $p \approx n^\alpha$ for $0 < \alpha < 1$ yields a lower bound of $n^{\alpha(n-n^\alpha)}$, showing that $\#(n)$ is super-exponential in n. Further, as $\#(n) = \sum_{p=1}^{n} \#(n,p)$, the average size of $\#(n,p)$ is $\#(n)/n$, hence, this average is also super-exponential in n.

For positive integers $n_1, \ldots, n_p$, let $\#(n_1, \ldots, n_p)$ denote the number of labeled $(n_1, \ldots, n_p)$-partitions of a subset of $\mathcal{N}$ consisting of n elements, where $\sum_{j=1}^{p} n_j = n$. Part 1 can be assigned any n_1 elements from the available n elements, so there are $\binom{n}{n_1}$ possible selections. Part 2 can then be assigned any n_2 elements of the remaining $n - n_1$ elements, so there are $\binom{n-n_1}{n_2}$ possible selections. Inductively, we conclude that there are $\binom{n - \sum_{j=1}^{i-1} n_j}{n_i}$ possible selections for part i. Hence,

$$\#(n_1, \ldots, n_p) = \prod_{j=1}^{p} \binom{n - \sum_{k=j}^{p} n_k}{n_j} = \frac{n!}{\prod_{j=1}^{p} (n_j!)} \ .$$

Interchanging two parts of the same size in a labeled shape partition leaves the corresponding unlabeled shape partition invariant. For each integer $m = 1, \ldots, n-p+1$, let $p_m \equiv |\{j \in N : n_j = m\}|$. Then

$$\overline{\#}(n_1, \ldots, n_p) = \frac{\#(n, \ldots, n_p)}{\prod_{m=1}^{n-p+1} (p_m!)}$$

(where 0! is defined, as usual, to be 1). This number is reduced to approximately

$$\frac{n!}{\left(\frac{n}{p}!\right)^p p!}$$

for uniform shape partitions. Note that this number is exponential in n for any fixed $p > 1$.

The number of constrained-shape (labeled) partitions, including bounded-shape (labeled) partitions, can be obtained by summing up the numbers of shaped (labeled) partitions over all shapes contained. The number of labeled shapes is the coefficient of the x^n term in the expansion of $\prod_{j=1}^{p} \sum_{k=L_j}^{U_j} x^k$. For the unlabeled case, it is the number of distinct coefficient terms of the term x^n in the expansion $\prod_{j=1}^{p} \sum_{k=L_j}^{U_j} a_k x^k$. Note that $a_i a_k$ and $a_k a_i$ are considered the same coefficient term. For example, in the expansion of $(a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5)(a_2 x^2 + a_3 x^3 + a_4 x^4)$, the coefficient of x^6 has two distinct terms, $a_2 a_4$ and a_3^2.

Similarly, the number of constrained (labeled) partitions, including bounded-size (labeled) partitions, can be obtained by summing up the numbers of size (labeled) partitions over all sizes contained.

Suppose an objective function is defined over the corresponding family of partitions. It follows from the above counting of the number of partitions, that a brute force approach to find an optimal open partition, an optimal size partition or an optimal shape partition by enumeration of all partitions in the relevant space requires exponential time. Furthermore, it is unlikely that efficient algorithms for determining optimal size partitions can be found since Hwang [1981] observed that the problem of partitioning a set of numbers into two parts under a certain objective function can be reduced to the well-known "number partition" problem which is known to be NP-hard; see Garey and Johnson [1977], for example. Also, Chakravarty, Orlin and Rothblum [1982] showed that even the uniform shape problem with $p = 3$ can be reduced to the 3-partition problem which is also known to be NP-hard; again, see Garey and Johnson [1979]. The focus of this book is to find conditions under which the search for an optimal partition can be conducted efficiently; for example, by confining the search to a set smaller than the set of all partitions, hopefully to one whose size is polynomial in the parameters of the problem.

So far, the countings in this section were done under the assumption that parts are required to be nonempty. But, it is easy to derive the corresponding counts of partitions that allow for empty parts. For size problems, we already noted that the number of labeled p-partitions that allow for empty parts is p^n. The number of unlabeled p-partitions that allow for empty parts is $\sum_{q=0}^{p-1} \bar{\#}(n, p-q)$; here q can be thought of as the number of empty parts. Also, the formula we derived for single shape problems with empty parts prohibited apply to the case where empty parts are allowed – here $n_j = 0$ indicates that part j must be empty and for the unlabeled case, the range of m in the corresponding product is $m = 0, \ldots, n$. Most importantly, the complexity-order – polynomial or exponential – is the same when empty parts are allowed or excluded.

In our development we introduce partition properties and demonstrate that for particular partition problems, the search for optimal partitions can be restricted to partitions that satisfy these properties. Let Q denote such a property. The number of partitions that satisfy Q (or a bound on this number) is then a major element in the usefulness of results that allow one to restrict attention to partitions satisfying Q. We shall therefore be interested in computing the number of partitions that satisfy Q and have a prescribed shape or a prescribed size. Let $\#_Q(n_1, ..., n_p)$ and $\#_Q(n, p)$ denote these numbers when parts are required to be nonempty.

We now give some general formulas which express the corresponding numbers when empty parts are allowed in terms of those that prohibit them. Let "$\#^*_Q$" and "$\bar{\#}^*_Q$" replace, respectively, "$\#_Q$" and "$\bar{\#}_Q$" when allowing for empty parts. Then $\#^*_Q(n_1, ..., n_p)$ and $\bar{\#}^*_Q(n_1, ..., n_p)$ for nonnegative n_j's are usually expressed by the same formulas as for the case where the n_j's are positive (with the substitution of $n_j = 0$ whenever applicable). Also, for partition properties that satisfy the reduction assumption (which is true for most properties that we consider), we have that

$$\#^*_Q(n, p) = \sum_{q=1}^{p} \binom{p}{q} \#_Q(q), \text{ and}$$

$$\bar{\#}^*_Q(n, p) = \sum_{q=1}^{p} \bar{\#}_Q(q).$$

These formulas are significant as polynomiality in n is invariant of whether or not empty parts are allowed.

The formulas of the above paragraph apply to partition properties that are defined in terms of the partitioned indices; this is the case for most partition properties that apply to single-parameter spaces, studied in Chapter 6. But, properties of partitions over multi-parameter spaces are defined in terms of the associated vectors (the A^i's). In this case, the number of partitions that satisfy a property Q depends on the partitioned vectors (the A^i's) and their dimension. Still, when the reduction assumption holds, the above formulas, expressing the number of partitions that satisfy a property Q and allow for empty parts in terms of the expressions that apply to the case where empty parts are prohibited, extend to multi-dimensional vector-partitions. In particular, polynomiality in the number n of partitioned elements is invariant of whether or not empty parts are allowed.

When counting the number of partitions (in Sections 6.2 and 10.2), we shall consider only the case where empty parts are prohibited and we shall not repeat the conversion formulas.

1.4 Examples

In the current section, we demonstrate the expressive power of the framework of partition problems introduced in the previous section through 15 examples. In particular, we demonstrate (in Subsections 1.4.13 and 1.4.14) that even when either d or p is fixed, sum-partition problems may be NP-hard, even with a convex objective function. So, polynomial time algorithms for solving such problems are expected to exist for (convex) objective functions only when both d and p are bounded.

1.4.1 *Assembly of Systems*

Consider a system having p modules as components. Each of these modules can be either *operative* or *inoperative*. The *state* of the system is determined by the set of operative modules and is represented by a vector $s \in \{0,1\}^p$, where $s_i = 0$ if module i is inoperative and $s_i = 1$ if module i is operative. The operativeness of the system is determined by a *structure function* $J; \{0,1\}^p \to \{0,1\}$, i.e., the system is *inoperative* if it is in a state s with $J(s) = 0$ and the system is *operative* if it is in a state s with $J(s) = 1$. The system is called *coherent* if the structure function is *monotone*, that is, if $J(s) \leq J(s')$ for $s,\ s' \in \{0,1\}^p$ with $s \leq s'$. (We note that the standard definition of coherence has an added requirement which we do not need herein.)

The modules are assumed to be composed of parts which are functionally interchangeable, with module $j \in \{1,\dots,p\}$ requiring exactly $n_j > 0$ parts. The modules are constructed in series, that is, a module is operative if and only if each of its parts is operative. All needed $n = \sum_j n_j$ parts are assumed to be available. An *assembly* for the system is an assignment of parts to the modules in a way that matches the requirements of each module; it corresponds to a partition with shape $(n_1,\dots,n_p)$ and we identify assemblies with partitions.

The *reliability* of a part, a module and the system as a whole is the probability of being operative. We assume that positive reliabilities of the parts are given and that operativeness of the parts are stochastically independent. Also, the parts are enumerated in a weakly increasing order of their reliabilities. So, with θ^i as the log of the reliability of the i-th part, we have that (1.2.5) is satisfied.

The reliability of a module depends on its composition. Given an assembly $\pi = (\pi_1,\dots,\pi_p)$, the series structure of the modules implies that the reliability of module j is given by

$$r(\pi)_j \equiv \prod_{i \in \pi_j} \exp(\theta^i) = \exp(\theta_{\pi_j}) , \quad \text{for } j = 1, \ldots, p \ .$$

The reliability of the system as a whole depends on the way it is constructed. Let r be a vector whose coordinates $r_1, \ldots r_p$ are, respectively, the reliabilities of the modules. Then the system's reliability is the expectation of $J(\underline{s})$ where $\underline{s}$ is a random vector whose components have independent Binomial distributions with coefficients $r_1, r_2, \ldots, r_p$ and is given by

$$h_p(r) = \sum_{s \in \{0,1\}^p} J(s) \left[\prod_{\{j : s_j = 0\}} (1 - r_j) \right] \left[\left(\prod_{\{j : s_j = 1\}} r_j \right) \right] \ .$$

With $h_p : \mathbb{R}^p \to \mathbb{R}$ as the function defined for $\xi \in \mathbb{R}^p$ by $f_p(\xi) = h_p(e^{\xi_1}, \ldots, e^{\xi_p})$, the system's reliability under assembly $\pi = (\pi_1, \ldots, \pi_p)$ is then given by

$$F(\pi) \equiv h_p[r(\pi)_1, \ldots, r(\pi)_p] = f_p(\theta_{\pi_1}, \ldots, \theta_{\pi_p}) = f_p(\theta_\pi) \ .$$

Thus, the problem of finding an assembly π that maximizes the system-reliability function $F(\cdot)$ is a single-parameter single-shape sum-partition problem. This optimal assembly problem has been widely studied in the literature, e.g., Derman, Lieberman and Ross [1972], El-Neweibi, Proschan and Sethuraman [1986,1987], Du [1987], Du and Hwang [1990], Malon [1990], Hwang, Sun and Yao [1985] and Hwang and Rothblum [1994a] among others.

If the items that are to be assigned to the modules are of m different types, we have that m parallel partitions have to be executed and we are in the framework of multipartitions. If the parts of all types are set in series within the modules, that is, the module fails when any one of the parts that are assigned to it fails, then we still have a summation function for the multipartition problem. Specification of the number of parts of each type that is to be assigned to each module results in a single-multishape multipartition problem.

1.4.2 *Group Testing*

Consider a set $\mathcal{N}$ of items each of which can be either good or defective and the states of the items are random with independent Bernoulli distributions; the probability of any one item being good is assumed to be ρ. The problem is to identify all defective items by using a minimum expected number of group tests. A *group test* takes an (arbitrary) subset S of $\mathcal{N}$ as input and outputs either a *negative* or a *positive* outcome: A negative outcome indicates that all items in the group are good and a positive outcome indicates the reverse, namely, that there exists at least one (unspecified) defective item in the group.

Dorfman [1943] first proposed using group testing to screen blood samples of military recruits for syphilis (the modern-day modification would be for AIDS or SARS). He proposed a procedure which partitions the n samples into p disjoint groups each receiving a group test. Once a group of items is tested and the outcome is negative, all items in the group can be readily classified as good, while items in a group with positive outcome are further tested, individually, to determine their states (with degenerate group tests in which single items are tested). The expected number of group tests required for a group of size x is given by

$$g(x) = \begin{cases} 1 & \text{if } x = 1\,, \text{ and} \\ 1 + x(1 - \rho^x) & \text{if } x > 1\ . \end{cases}$$

The total expected number of group tests for a procedure with p groups of sizes $n_1, \ldots, n_p$ is then $\sum_{j=1}^{p} g(n_j)$. The problem is to determine p and $n_1, \ldots, n_p$ so as to minimize this expression. This is an open sum-partition problem with $\theta^i = 1$ and with

$$F(\pi) = f(n_1, \ldots, n_p) = \sum_{j=1}^{p} g(n_j)\ ;$$

in particular, the function f is additive. The symmetry of f and the fact that the labeling of the tests are irrelevant, suggest that the problem can be viewed as an optimization over open unlabeled partitions.

1.4.3 *Circuit Card Library*

Designs of many types of electronic equipments incorporate a wide variety of basic circuits. The cost of individually manufacturing and stocking all these basic circuits can be considerable. The concept of a *standard library* of replaceable units has been introduced by Kodes [1972]. The library consists of cards on which several basic circuits are printed. Whenever a particular basic circuit is required, a card containing that circuit is used to supply it, with all additional circuits on the card discarded. The cards which contain multiple circuits are called *circuit cards.* The use of circuit cards reduces the total number of distinct replaceable units that must be manufactured and stocked; on the other hand, their production is more expensive than that of individual circuits and their use results in some waste.

In the design of a standard library, the choice of which circuits to place on which cards can be of considerable economic importance. Consider the circuit partition problem with the following characteristics:

(i) A set of n circuits has to be available. Circuit i is identified by a pair of parameters (c_i, r_i) where c_i is the cost of a unit of a circuit of type i and r_i is the periodic (say annual) requirement for circuits of that type.
(ii) An upper bound U is given on the number of circuits allowed per card.
(iii) A stocking cost s_j per card of type j is known; it captures the annual fixed cost associated with the stocking and production of any given type of circuit card.

Upon request of a given circuit, a card that contains it has to be produced. Since one will always select the card with minimum associated cost, it will be assumed that no circuit appears on more than one card. Thus the library design is a partition problem of the n circuit types into a set of, say, p disjoint card types. Assuming that $s_j = s$ for all j, if $\pi_1, \ldots, \pi_p$ are the sets of circuits that are to be produced jointly, the associated cost is

$$F(\pi) = \sum_{j=1}^{p} \left(\sum_{i \in \pi_j} c_i \right) \left(\sum_{i \in \pi_j} r_i \right) + sp \ . \tag{1.4.1}$$

So, the above library design problem is formulated as a two-parameter, open, sum-partition problem with $f_p(X) = \sum_{j=1}^{p} X_{1j} X_{2j} + sp$ for $X \in \mathbb{R}^{2 \times p}$.

1.4.4 *Clustering*

Given a set of n points $A^1, \ldots, A^n$ in the d-space, the clustering problem asks for a partition of these n points into "clusters" such that points in the same cluster are "close" relative to points in different clusters under a given distance function. The number of clusters can be either fixed or set as a decision variable. So the clustering problem corresponds to either a size-partition problem or an open-partition problem.

In general, it is very difficult to obtain an exact optimal clustering. Most procedures proposed in the statistical literature are numerical. One exception is the single-parameter clustering problem solved by Fisher [1958] in which the number of clusters is fixed at p and point i is assigned a weight w_i. The distance function to be minimized is

$$\sum_{j=1}^{p} \sum_{i \in \pi_j} w_i(\theta_i - c_j)^2$$

where c_j is the weighted mean of the points in π_j. Fisher proved the existence of an optimal clustering which has the "consecutive" property, i.e., items in the same cluster have consecutive indices, and thus can be efficiently identified. He expressed a desire to see a generalization of this result to points in d-space (the notion of "consecutiveness" needs to be generalized too), that is, a d-parameter size-partition problem.

1.4.5 *Abstraction of Finite State Machines*

Oikonomou and Kain [1983] introduced the concept of "abstraction" to model the situation in which an observer is monitoring a complex system subject to malfunction. Let the system be represented by a finite state machine M with state-set Q, input-set X, output-set Y and two functions δ and λ which map $(Q \times X)$ into Q and Y, respectively; the interpretation of these functions is that δ assigns to each state-input pair the next state that will be observed while λ assigns to such a pair the corresponding output. Due to possible interface limitations, or the desire for a simplified model, or simply because of cost consideration, the observer often has to construct a nondeterministic finite state machine M_A by lumping each of Q, X and Y into classes, specified by partitions π_Q, π_X and π_Y, respectively; a part of π_Q, π_X and π_Y then represents a state, an input or an output of M_A, respectively. It is assumed that the number of parts in these partitions is prescribed. Let $(\pi_Q)_j, (\pi_X)_j, (\pi_Y)_j$ denote the j^{th} part of π_Q, π_X, π_Y, respectively. The output function λ_A of M_A is defined such that a state-input pair of M_A, say $((\pi_Q)_i, (\pi_X)_j)$, is mapped into a set $\lambda_A((\pi_Q)_i, (\pi_X)_j)$, called an *admissible set of outputs*, consisting of all parts $(\pi_Y)_k$ of π_Y such that there exist $q \in (\pi_Q)_i$, $x \in (\pi_X)_j$ and $y \in (\pi_Y)_k$ with $\lambda(q, x) = y$. The next-state function δ_A and its admissible sets are defined similarly. Henceforth, whenever π is given (without ambiguity about its identity), we simplify notation and refer to $(\pi_Q)_i, (\pi_X)_j$ and $(\pi_Y)_k$ by Q_i, X_j and Y_k.

Oikonomou and Kain assumed that if π_Q coincides with the decomposition of the state set Q into irreducible closed subsets, that is, each $(\pi_Q)_i$ is an irreducible closed set, and the steady-state distribution of the probabilities of $y \in Y_k$ given $q \in Q_i$ and $x \in X_j$ are known.

A λ-fault of M with domain $D \subseteq Q \times X$ is a mapping $\epsilon_D : D \to Y$ such that $\epsilon(q,x) \neq \lambda(q,x)$ for every pair $(q,x) \in D$; the multiplicity of such a λ-fault is then $|D|$. In particular, a λ-fault with multiplicity 1 is called a "single-fault". A δ-fault and its multiplicity can be similarly defined. The existence of both types of faults, λ-faults and δ-faults, renders the analysis of the model extremely difficult since when an error-output occurs, we don't even know whether it is due to an δ-fault or a λ-fault. Oikonomou [1987] simplified the matter by assuming no δ-fault can occur which we will follow in the rest of this section.

While some faults of M can be immediately detected, e.g., when the output is not in the admissible set, most faults are detected by comparing the steady-state distributions of the faultless M and fault-allowed M through long-term observations. However, there are faults which cannot be detected inherently by lumping the data. This is the case when $\epsilon(q,x) \neq \lambda(q,x)$, but both $\epsilon(q,x)$ and $\lambda(q,x)$ are in the same part of π_Y. Oikonomou studied the minimization of the number of undetected λ-faults, but found this multipartition problem (not covered by the sum-multipartition problem in Chapter 11) too difficult. Thus he focused on the minimization the number of single undetectable λ-faults, i.e., the number of pairs $(q,x) \in Q \times X$ for which $\epsilon(q,x) \neq \lambda(q,x)$, but both are in $(\pi_Y)_j$ for some j.

Consider a fixed pair (q,x). Let $Y_{(q,x)}$ denote the part of π_Y containing (q,x). Then any λ-fault $\epsilon(q,x)$ which maps to an element in $Y_{(q,x)}$ that is different from $\lambda(q,x)$ is a single undetectable λ-fault. This number is simply $|Y_{(q,x)}| - 1$, and the total number over all pairs (q,x) is

$$\sum_{(q,x)\in Q\times X} (|Y_{(q,x)}| - 1) = \sum_{(q,x)\in Q\times X} |Y_{(q,x)}| - |\pi_Q||\pi_X|.$$

Since the last term is a constant, we can set the objective function to be

$$F(\pi) = \sum_{(q,x)\in Q\times X} |Y_{(q,x)}|.$$

For $y \in Y$, define $m(y)$ as the number of (q,x) pairs such that $\lambda(q,x) = y$. Then the objective function can be written as

$$F(\pi) = \sum_{k=1}^{|\pi_Y|} |Y_k| \sum_{y\in Y_k} m(y).$$

Note that $F(\pi)$ depends on π_Y only; hence the problem is reduced to a single-partition problem. Further, set $\theta^y = m(y)$ for $y = 1, ..., |Y|$. Then $F(\pi)$ presents a size sum-partition problem with dependence on part-sizes.

1.4.6 *Multischeduling*

In a multischeduling problem, a set of n jobs is to be assigned to p machines, but the cost of running these jobs not only depends on the assignment (which is a partition problem), but also on the particular order the jobs are run on a machine. The general multischeduling problem will not be covered in this book. However, in some cases, the ordering part is trivial; so the multischeduling problem is essentially reduced to a partition problem. For example, suppose the n jobs are linearly ordered in some ways, perhaps by their arrival times, and each machine must run its assigned jobs not violating this linear order.

Mehta, Chandrasckaran and Emmons [1974] considered a 2-machine multischeduling problem with a set of n jobs $J_1, \ldots, J_n$, where $i > j$ implies J_i will be worked on before J_j. The time required to do job i is t_i on either machine. The goal is to minimize the *total flow time*, the sum of time each job spends in the system (from time 0 to the time the job is done). Let $\pi = (\pi_1, \pi_2)$ be a partition of $\mathcal{N} = \{1, \ldots, n\}$ with π_1 as the indices of the jobs assigned to the first machine and π_2 as the jobs assigned to the second one. Order the job assigned to the same machine according to their indices. The *position* of a job in π_j is its relative order in π_j, counting reversely. For example, if $\pi_1 = (8, 7, 5, 3, 1)$, then the position J_3 is 2. Let $T_\pi(i)$ denote the contribution of t_i to the total flow time. Suppose i is in position $k_\pi(i)$ of a part of π. Then

$$T_\pi(i) = k_\pi(i) t_i,$$

since each of $k - 1$ jobs after J_i also waits t_i units of time along with J_i. Finally,

$$F(\pi) = \sum_{i=1}^{n} T_\pi(i) = \sum_{i=1}^{n} k_\pi(i) t_i.$$

Note that we can also write

$$F(\pi) = \sum_{i \in \pi_1} k_{\pi_1}(i) t_i + \sum_{i \in \pi_2} k_{\pi_2}(i) t_i$$

where $k_{\pi_1}(i)$ and $k_{\pi_2}(i)$ are derived from $k_\pi(i)$. Thus, this multischeduling problem is reduced to a 2-size partition problem, but not a sum-partition problem due to the existence of the weight $k_\pi(i)$ which depends on π.

1.4.7 *Cache Assignment*

Gal, Hollander and Itai [1994] considered the problem of assigning computer jobs to caches which are specially fast memory devices. Jobs are stored in memory pages, while a cache can store a limited amount of memory pages. If a job requested is not in the cache, then a cache miss occurs and the whole page containing that job is transferred to the cache to replace its current content. It is assumed that the probability that job i is the job requested is known to be ρ_i. The problem is to assign jobs to p memory pages to minimize cache misses.

Note that a cache hit (opposite of cache miss) occurs only when the last job request is on the same page π_j with the current job requested. The probability of that is

$$\sum_{j=1}^{p} \left(\sum_{i \in \pi_j} \rho_i \right)^2 . \tag{1.4.2}$$

Then $F(\pi) = f(\sum_{i \in \pi_1} \rho_i, \ldots, \sum_{i \in \pi_p} \rho_i)$ where f is the sum of square function.

1.4.8 *The Blood Analyzer Problem*

Nawijn [1988] studied the optimization of the production rate of an automated blood analyzer for blood testing. Usually, several tests have to be performed on a single blood specimen. The blood analyzer can perform all tests required by all blood specimens. To execute a test, blood is first supplied to a cup, and then an agent associated with the test is added. The $n = pm$ agents for the n tests are grouped into p columns, each having m (assuming an integer) agents, around a turntable. Consider the procedure of testing a given blood specimen. At each column the turntable stops, m new cups, one for each agent in the column, are supplied with blood from the specimen and agents for the required tests are added to the corresponding cups with tests performed simultaneously. The turntable starts at a position before column 1 and always stops at the next column containing an agent associated with a test in the required set (the testing ends if no such column exists before returning to the starting position). The time required for going from one stop to the next (including the stopping) is constant, regardless of how many (or any) columns are skipped, takes constant time. Thus, it is desirable to partition the n agents into p groups to minimize the number of columns containing a required set such that the total time and the number of cups needed for testing a given specimen are both minimized. Since each specimen may require a different set of tests, and information about which specimens are coming for testing is generally not available at the grouping stage, Nawijn considered the problem for a random specimen whose set of required tests may be inferred (in a probability sense) by past data on specimens. For a random specimen, define $Z_i = 1$ if test i is required and $Z_i = 0$ otherwise. Let "Pr" stand for the the probability function. Then the cost of $f(\pi)$ associated with a partition π is the expected number of columns containing a required set, i.e.,

$$F(\pi) = p - \sum_{j=1}^{p} Pr\left(\sum_{i\in\pi_j} Z_i = 0\right).$$

Under the assumption that the Z_i's are independent, then

$$Pr\left(\sum_{i\in\pi_j} Z_i = 0\right) = \prod_{i\in\pi_j} Pr(Z_i = 0).$$

Define for each $i = 1, \ldots, n$, let $\theta^i = \ln Pr(Z_i = 0)$. For a partition π we then have

$$\sum \theta_{\pi_j} = \sum_{i\in\pi_j} \ln Pr(Z_i = 0) \text{ for } j = 1, \ldots, p$$

and

$$F(\pi) = f\left(\sum \theta_{\pi_1}, \ldots, \sum \theta_{\pi_p}\right) = p - \sum_{j=1}^{p} e^{\Sigma\theta_{\pi_j}};$$

so the problem has been formalized as a p-size sum-partition problem.

1.4.9 *Joint Replenishment of Inventory*

Chakravarty, Orlin and Rothblum [1985] considered an economic order quantity model involving n items, where the i-th item has (deterministic) demand rate D_i, a unit inventory holding cost h_i per unit time, and a fixed cost K_i for placing an order. The problem is to partition the n items into p subgroups and choose order cycles for the groups out of a given set of allowable (joint) order cycles, so as to minimize the net average cost per unit time.

Assume that the demand occurs at constant rate of D_i for item i. Set $a_i = 2^{-1}h_iD_i$ and $b_i = K_i$ for each item i, and for a subset S of $\{1, \ldots, n\}$ let $a_S = \sum_{i\in S} a_i$ and $b_S = \sum_{i\in S} b_i$. As in the ordinary EOQ model (e.g., Wagner [1969, pp. 18–19]), if item i has order cycle τ, then its order quantity is τD_i and average inventory cost per unit time is $2^{-1}h_iD_i = a_i$.

In particular, the net cost over an order cycle of a group S of items having the same order cycle τ is

$$c(S,\tau) = \sum_{i\in S} 2^{-1}\tau D_i h_i \tau + \sum_{i\in S} K_i = \tau^2 a_S + b_S \ .$$

So, if the items are partitioned according to partition $\pi = (\pi_1, \ldots, \pi_p)$ and order cycles $t_1, \ldots, t_p$ are used, respectively, the total average cost per unit time is

$$c_p(\pi, t) = \sum_{j=1}^{p} t_j^{-1} c(\pi_j, t_j) = \sum_{j=1}^{p} (t_j a_{\pi_j} + t_j^{-1} b_{\pi_j}) \ ,$$

where $t = (t_1, \ldots, t_p)$. The p-vector of order cycles is assumed to be extracted from a set T of allowable p-vectors of order cycles. This formulation allows one to impose joint restrictions on the order cycles, e.g., the requirement that all order cycles are integer multiples of the smallest one, or individual restrictions, e.g., that each order cycle be an integer in the set $\{1, 7, 30\}$, representing daily, weekly and monthly deliveries. For a fixed partition $\pi = (\pi_1, \ldots, \pi_p)$ of the items, the minimum average cost per unit time is

$$F(\pi) = \inf_{t\in T} c_p(\pi, t) \ . \tag{1.4.3}$$

Evidently, since $c_p(\pi, t) \geq 0$, the infimum defining $F(\pi)$ is finite for each partition π. The problem is to find a partition of the items into subsets so as to minimize F.

We observe that $F(\pi)$ depends on the parts of the partition π through the sums of the a_i's and b_i's in each part. Thus, we have a two-parameter, size, sum-partition problem with

$$f_p(X) = \inf_{t\in T} \sum_{j=1}^{p} \left(t_j X_{1j} + t_j^{-1} X_{2j}\right) \ \text{for } X \in \mathbb{R}^{2\times p} \ .$$

In particular, as f_p is the infimum of linear functions in the variables of X, we have that f_p is concave. It is noted that while each partitioned element i is associated with three numbers h_i, D_i and K_i, the partition problem is a two-parameter one.

1.4.10 *Statistical Hypothesis Testing*

An experiment is conducted to test a null-hypothesis against a single alternative hypothesis. Outcome i of the experiment is associated with two parameters a_i and b_i, where a_i is the probability that outcome i occurs under the null-hypothesis and b_i is the probability that outcome i occurs under the alternative hypothesis. The problem is then to partition the set of outcomes into two sets – the *acceptance region* of outcomes under which the null-hypothesis is accepted, and the *rejection region* of outcomes under which the null-hypothesis is rejected. A type-1 error is then an error in accepting the null-hypothesis when it is false and a type-2 error is an error in rejecting the null-hypothesis when it is true. The goal is to define acceptance and rejection regions so as to minimize the cost of type-1 and type-2 errors.

Let p be the a priori probability of the decision maker that the null-hypothesis is true (before experiments are conducted). Suppose the acceptance region is π_A and the rejection region is π_R. Using Bayes Rule, the probabilities of first and second error are, respectively, $\sum_{i \in A} b_i(1-p)$ and $\sum_{i \in R} a_i p$. With cost C_1 for type-1 error and cost C_2 for type-2 error, the expected cost of adopting a partition $\pi = (\pi_A, \pi_R)$ is then $F(\pi) = C_1(1-p)(\sum_{i \in \pi_A} b_i) + C_2 p(\sum_{i \in \pi_R} a_i)$ and we see that this cost depends on the parts of the partition only through the sums of the parameters of the elements that are associated with them. We conclude that the problem is a linear two-dimensional, size 2, sum-partition problem. Of course, more general objective functions which depend on the probabilities of type-1 and type-2 problems can be considered, except that the associated function f will not be linear. Also, partition of the alternative hypothesis to multiple hypotheses converts the problem to size-partition problems with prescribed size with cardinality larger than 2.

We note that a limiting case of the linear case is the problem of maximizing the probability of one type of error subject to the other being fixed.

1.4.11 *Nearest Neighbor Assignment*

A city has p post offices (or refuse collection stations, police stations, motor vehicle inspection stations...) and wants to assign each of its n households to an office as its host office. It is natural to assign each household to its nearest post office except for the restriction that office j can serve at most U_j households and at least L_j households (the lower bound imposes a minimal efficiency requirement to justify the existence of the facility). Let $d(H_i, O_j)$ denote the planar distance from household i to office j. Then this assignment problem can be treated as a bounded-shape single-parameter sum-partition problem with the objective function

$$F(\pi) = \sum_{j=1}^{p} \sum_{i \in \pi_j} d(H_i, O_j).$$

Suppose that we decide to represent the location of either H_i or O_j as a two-dimensional vector, and the distance $d(H_i, O_j)$ also a two-dimensional vector, say, a function of $H_i - O_j$. Then the problem can be treated as a bounded shape 2-parameter sum partition problem

1.4.12 *Graph Partitions*

Consider a (directed) graph $G(\mathcal{N}, E)$ with $\mathcal{N}$ as the set of *vertices* and with $E \subseteq \mathcal{N} \times \mathcal{N}$ as the set of *edges*. For a subset $\mathcal{N}'$ of $\mathcal{N}$, let $E_{\mathcal{N}'}$ denote the subset of the set of edges E induced by $\mathcal{N}'$, that is, $E_{\mathcal{N}'}$ consists of all edges with both endpoints in $\mathcal{N}'$. The problem is to find a partition of $\mathcal{N}$ into p parts $\pi_1, \ldots, \pi_p$ such that

$$\sum_{\substack{u,v=1 \\ u \neq v}}^{p} |\{(s,t) : s \in \pi_u \text{ and } t \in \pi_v\}| = |E| - \sum_{j=1}^{p} \left|E_{\pi_j}\right|$$

is minimized, or equivalently, to maximize $\sum_{j=1}^{p} |E_{\pi_j}|$. For example, $\mathcal{N}$ can be the set of terminals to be distributed into p chips and E represents pairs of terminals which need to be connected (by a wire). Typically, an intra-chip connection (one within a chip) is relatively easy while an inter-chip connection (one between distinct chips) is much longer and takes up space and consumes the all-important "vias" which to a chip are like seaports to a continent. Under these circumstances, one would like to keep most connections internal to chips.

With $n = |\mathcal{N}|$, enumerate the vertices by $1, \ldots, n$ and associate the vertex i with the i-unit vector in $\mathbb{R}^n$. The matrix whose columns are the vectors associated with the partitioned elements is then the identity I in $\mathbb{R}^{n \times n}$, and the matrix associated with a partition π of $\mathcal{N}$ the $n \times p$ matrices whose (i, j) element is 1 if i is assigned to π_j and 0 otherwise. We observe that the subsets of $\mathcal{N}$ are in one-to-one correspondence with the vectors in $\{0,1\}^n$ with subset $\mathcal{N}$ of $\mathcal{N}$ corresponding to its characteristic vector χ_V having $(\chi_V)_i = 1$ if $i \in V$ and $(\chi_V)_i = 0$ otherwise. Let g_n be the function on $\{0,1\}^n$ which associates to the characteristic vector of subset V the number $|E_V|$. We note that for $j = 1, \ldots, p$, the j-columns of I_π, namely, I_{π_j} is the characteristic vector χ_{π_j} of π_j. Thus, the objective function F

over partitions that is to be maximized has

$$F(\pi) = \sum_{j=1}^{p} \left| E_{\pi_j} \right| = \sum_{j=1}^{p} g_n \left(\chi_{\pi_j} \right) = \sum_{j=1}^{p} g_n \left[I_{\pi_j} \right] = f_p \left(I_\pi \right),$$

where f_p is the real-valued function on $\{0,1\}^{n\times p}$ which assigns to matrix $J \in \{0,1\}^{n\times p}$ the number $f_p(J) = \sum_{j=1}^{p} g_n(J^j)$. We note that f_p can be extended to cover the convex hull of the I_π's, for example, by letting

$$f_p(x) = \sup \left\{ \sum_{\pi} \beta_\pi f_p \left(I_\pi \right) : \beta_\pi \geq 0 \text{ for all } \pi, \sum_{\pi} \beta_\pi = 1 \text{ and } \sum_{j=1}^{q} \beta_\pi I_\pi = x \right\};$$

this particular extension is convex. So, the above graph-partition problem is formulated as a convex, multi-dimensional, size, sum-partition problem.

When $p = 2$, and the objective is to maximize, rather than minimize, the above problem is referred to in the literature as the min-cut problem which is known to be NP-complete (e.g., Garey and Johnson [1979]).

1.4.13 *Traveling Salesman Problem*

Find a shortest Hamiltonian path on n sites under a given symmetric non-negative matrix D, where D_{ij} represents the *distance* between sites i and j.

<u>FORMULATION</u>: $n = p$, $d = 1$, $A = (1, \ldots, n)$, $\Lambda = \{1^n = (1, \ldots, 1)\}$,

$$F(\pi) = -\sum_{j=1}^{n-1} D_{\pi_j, \pi_{j+1}},$$

where we regard a partition simply as the corresponding permutation. The matrices A^π in this case are simply all permutations of A. The Shaped Partition Polytope $\mathcal{P}_A^\Lambda$ has $n!$ vertices which stand in bijection with Λ-partitions, and is the so-called *Permutohedron*. Since each A^π is a distinct vertex of $\mathcal{P}_A^\Lambda$, there is again a convex function C on $\mathbb{R}^n$ such that $F(\pi) = C(A_\pi)$ for all π. This problem is known to be NP-hard.

1.4.14 *Vehicle Routing*

A school employs m buses to pick up n students. The problem is to partition the n students into m routes (each served by a bus) to minimize the total length of the routes. Note that each route is a "traveling salesman's tour" passing through the school and the students (actually, their houses) on the route.

Since each student is represented by a point in the plane, the vehicle routing problem is a parametric partition problem. However, the objective function is the sum of the lengths of m minimum traveling salesman's tours while the "traveling salesman problem" itself is NP-hard.

1.4.15 *Division of Property*

Consider the problem of a set partition of n items among p individuals. Such problems can arise in a divorce settlement, in the division of an estate among heirs, or in a liquidation of a partnership and the allocation of the assets of the partnership among the shareholders. In such problems, each individual j associates value A_j^t to item t and his/her utility from getting a bundle of items S is, in the simplest case, the sum $\sum_{t \in S} A_j^t$. It is further assumed that a function $f_p : \mathbb{R}^p \to \mathbb{R}$ is given, referred to as the *welfare function.* The welfare function of a p-partition π is then given by $F(\pi) = f_p(\sum_{t \in \pi_1} A_1^t, \ldots, \sum_{t \in \pi_p} A_p^t)$ and the goal is to maximize F over a given set Π of (feasible) partitions. In this example, the number p of parts of considered partitions equals the dimension d of the partitioned vectors and $F(\pi)$ depends only on the diagonal elements of A^π.

In this problem one frequently focuses on lotteries over bundles, using the criterion of expected welfare; attention then shifts to the polytope defined as the convex hull of the A_π's. Of particular interest are the Pareto-optimal points in this polytope which are points that are not dominated through the regular partial order $>$ over $\mathbb{R}^p$ by any other point in the set. Granot and Rothblum [1991] discuss objective functions which guarantee Pareto-optimal solutions and have interpretation in the game-theoretic literature; for example, the Nash Solution and the Kalai-Smorodinsky Solution for bargaining games are maximizers of such functions (over the polytopes) (see Nash [1950], Kalai-Smorodinsky [1975] and Roth [1979]).

We note that our formulation extends to situations where an individual's utilities are multiplicative, rather than linear, by taking the logarithm of the (positive) A_j^t's.

1.4.16 *The Consolidation of Farm Land*

In many rural communities, farmers cultivate a large number of small-sized lots while those lots belonging to a particular farmer can scatter over the whole area. This situation causes two main disadvantages to the farmers. The first is that a small-sized lot is not efficient in operating agricultural machines; the second is that scattered lots increase overhead in traveling expenses and time. Thus it is desirable to redistribute the lots so that each farmer can have all his lots forming a shape as contiguous as possible, under the condition that the market values of his new lots is comparable to what he owned before. This problem was first raised by Brieden and Gritzmann [2004], and further discussed in Borgwardt, Brieden and Gritzmann [2009].

This application extends our original partition model in several directions. First, while a lot is a 2-dimensional object, it is not a point but an area (could be odd-shaped). The above authors resolved this by representing a lot by its gravity center point. Secondly, a geometric property, contiguity, contributes to the objective function. We may study the extreme case that a solution must be contiguous. Note that contiguity is a property of a lot partition, which needs to be translated into property of a partition on points. Finally, in the real world, the market value of a lot is not just determined by its size, but many other factors which tend to result in a complicated objective function with multiple-parameters.