Balanced Assignment of Experimental Units in the Analysis of Covariance through Optimization

Robert Howley
Lehigh University

Robert H. Storer
Lehigh University

Report: 11T-010

Balanced Assignment of Experimental Units in the Analysis of Covariance through Optimization

Robert Howley[1], Robert H. Storer[2*]

Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015, USA
[1]E-mail: roh210@lehigh.edu
[2]E-mail: rhs2@lehigh.edu

Abstract

In a designed experiment with covariates, experimental units are typically assigned to treatments randomly and analysis of covariance is used to account for the covariate. In cases where the covariate is known beforehand, the possibility exists to assign experimental units systematically to achieve a "balanced" covariate distribution in each treatment. This balance can be accomplished by solving a multi-criteria number partitioning problem. We discuss approximate methods for solving this NP hard optimization problem and present simulation results quantifying increases in the power of the statistical test for differences in treatment means.

Keywords: number partitioning, Karmarkar-Karp, analysis of covariance, nonrandom assignment, alternate ranks

1 Introduction

The number partitioning problem is well known, and an elegant heuristic proposed by Karmarkar and Karp (KK [1]) has been proposed for finding good solutions quickly. Korf extended the KK heuristic to an anytime complete algorithm [2] that can find improved solutions at the expense of increased computation. While these algorithms appear to be very effective, few if any practical applications have been found for the problem [3]. In this paper we study generalizations of NP and develop algorithms for these problems by extending the ideas behind the KK algorithm. In particular we introduce the multi-criteria number partitioning problem (MCNP) and an equal cardinality constrained MCNP.

MCNP has many potential applications, particularly in statistics. The problem of finding a representative subset of a dataset, or partitioning a dataset into two representative subsets can be modeled as a MCNP. We apply this method to small sample experimental design in the Analysis of Covariance (ANCOVA). We assign experimental units to treatments using MCNP to create a balanced assignment with respect to covariate distribution. We show that this results in enhanced statistical power over random assignment and over alternate ranks designs. We then extend the method to the case of multiple covariates.

*Corresponding author

2 Number Partitioning

2.1 Problem Formulation

The Number Partitioning problem (NP) is well known and easily described: partition a set of n non-negative numbers into two sets so that the sums of the numbers in the two sets are as close as possible in absolute value. Let w_i, $i = 1, \ldots, n$ be n non-negative numbers we seek to partition, and let $W = \sum_{i=1}^{n} w_i$. Define $x_i = 1$ if item i is assigned to set 1 and $x_i = 0$ if item i is assigned to set 2. The the problem can be formulated as

$$\begin{array}{ll} \min & \left| \sum_{i=1}^{n} w_i x_i - w_i(1 - x_i) \right| = \min \left| -\frac{1}{2} W + \sum_{i=1}^{n} w_i x_i \right| \\ s/t & x_i \in \{0, 1\} \quad \forall\, i \end{array} \tag{1}$$

NP is one of six original NP-complete problems cited by Garey and Johnson [4]. In [1], Karmarkar and Karp introduced a simple and elegant heuristic (which we refer to as the KK heuristic) for NP problem which we discuss later. Johnson et.al. [**?**] compared simulated annealing local search algorithms for NP to the KK heuristic and concluded that KK was more effective despite requiring orders of magnitude less computing time. Flanders and Storer [5] and Arguello and Feo [6] both proposed randomized versions of the KK heuristic which obtained better objective function values at the expense of greater computation time. Korf developed a branch and bound procedure based on the KK algorithm [2]. Choi and Storer [7] describe an effective heuristic for the turbine rotor blade balancing problem based on finding embedded NPs in the problem. James and Storer [8] propose algorithms for the subset sum problem, one of which is based on the KK algorithm. However, the KK based algorithm was inferior to other subset sum algorithms proposed in the same paper.

An interesting and well known property of NP is that many instances can be easily solved. Indeed the problem has been termed the easiest hard problem by Hayes [9]. At the risk of over simplification, the problem becomes easier to solve as n increases, but more difficult to solve as the number of digits required to represent the numbers, w_i, increases. With large n, it is usually possible to get partitions that are extremely close to balanced.

2.2 The Karmarkar-Karp Heuristic

The Karmarkar-Karp (KK) heuristic represents the problem with n nodes, each labeled with one of the numbers w_i. An iteration of the algorithm consists of joining with an arc the two nodes with the largest labels, removing one node from live and relabeling the other node with the residual imbalance thus reducing the problem to a similar problem with one less node.

Algorithm 1: KK

1. Label nodes with the numbers w_i and place all nodes in set "live"
2. Among nodes in set "live", find the node with the largest label (call it node u) and the node with the second largest label (node v).
3. Join u and v with an arc. Relabel node u with the difference $w_u - w_v$. Remove node v from set "live".
4. Repeat steps 2 and 3 until set "live" contains only one node.
5. Two color the resulting tree to obtain the partition. The label on the last node in set "live" will give the objective function value.

When two nodes are joined with an arc, they are placed on opposite sides of the partition. After iteration i of the algorithm, the graph will be a forest of exactly $(n - i)$ trees. Each tree in the forest will have exactly one "live" node. The current label of the live node in a tree is the residual imbalance of the partial partition represented by the tree. A key insight is that while the algorithm places nodes on opposite sides at each step, it does not decide which side until the end. Another insight is that at each iteration, the algorithm reduces the problem on k nodes to a problem on $k - 1$ nodes, while reducing the magnitude of numbers by replacing the two largest numbers with their difference.

3 Number Partitioning Extensions

In this section we introduce two extensions to the basic NP, multi-criteria number partitioning (MCNP) and a variant on MCNP that enforces equal cardinality of the subsets. For each new problem introduced we propose heuristics based on direct extensions of the basic KK procedure. Each heuristic may also be converted to branch and bound methods following Korfs procedure.

3.1 Multi-criteria Number Partitioning

Multi-criteria number partitioning (MCNP) extends the number partitioning problem to the case where each of the nodes to be partitioned is labeled with multiple dimensions of numbers. We seek to balance all dimensions simultaneously. For illustration, consider dividing a set of jobs between two workers. Each job (node) is characterized by two dimensions, the time needed to do the job, and the amount of physical exertion the job requires. We seek to divide the jobs evenly so that both time and exertion are evenly balanced between workers. This problem can be formulated as follows.

Let $x_i = 1$ if node i is assigned to set 1 and $x_i = 0$ if node i is assigned to set 2. Let w_{ij} be the j^{th} number on node i. The problem may be formulated as

$$\begin{aligned} \min \quad & \sum_{j=1}^{m} \left| \sum_{i=1}^{n} w_{ij} x_i - w_{ij}(1 - x_i) \right| \\ s/t \quad & x_i \in \{0, 1\} \end{aligned} \tag{2}$$

Letting $W_j = \sum_{i=1}^{n} w_{ij}$, the objective may be rewritten as follows:

$$\begin{aligned}
\min \quad & \sum_{j=1}^{m} \left| \sum_{i=1}^{n} w_{ij}x_i - w_{ij}(1-x_i) \right| = \sum_{j=1}^{m} \left| \sum_{i=1}^{n} w_{ij}x_i - w_{ij} + w_{ij}x_i \right| \\
& = \sum_{j=1}^{m} \left| \sum_{i=1}^{n} 2w_{ij}x_i - w_{ij} \right| \\
& = \sum_{j=1}^{m} \left| -W_j + \sum_{i=1}^{n} 2w_{ij}x_i \right|
\end{aligned}$$

or equivalently

$$\min \quad \sum_{j=1}^{m} \left| -\frac{1}{2}W_j + \sum_{i=1}^{n} w_{ij}x_i \right|$$

We propose a modification to the Karmarkar-Karp differencing algorithm for the multi-criteria case. An intuitive view of the basic KK differencing algorithm is that at each step two "big" numbers are removed from the "live" list and replaced with a smaller number. Our multi-criteria implementation operates on the same principle. We begin by creating n nodes. Each node, i, is labeled with the m values $w_{i1}, \ldots, w_{im}$. In addition we label each node with a "cost" quantity defined by:

$$c_i = \sum_{i=1}^{m} |w_{ij}| \tag{3}$$

We call the label c_i the "node cost" as it is a measure of the overall magnitude of the numbers on the node. We proceed as in the differencing algorithm by reducing the number of "live" nodes by one in each iteration. Our goal is to remove two nodes with large c_i values and replace them with a node with a smaller c_i value. At the end of the procedure, one node remains in "live", and its c_i cost value is the objective function of the partition.

The basic operation in the KK differencing algorithm is to place two nodes on opposite sides of the partition and relabel the larger node with the difference. This new label represents a "remainder" needing to be accommodated. By always relabeling the larger valued node, we assure that all node labels remain positive. In the multi-criteria case, we cannot assure that all node labels w_{ij} will remain positive throughout the algorithm. Thus we must consider placing nodes on the same side of the partition as well as on opposite sides. Consider the following example with two nodes:

$w_{u1} = 12$	$w_{v1} = 5$
$w_{u2} = 4$	$w_{v2} = 10$
$c_u = 16$	$c_v = 15$

If placed on opposite sides of the partition, their "remainder node", i.e. the relabeled live node u, will be:

$w_{u1} = 7$
$w_{u2} = -6$
$c_u = 13$

representing the residual imbalance. This example illustrates that intermediate "remainder" node labels may become negative. Next suppose that nodes u and v are labeled as follows:

$$\boxed{\begin{array}{l} w_{u1} = 7 \\ w_{u2} = -6 \\ c_u = 13 \end{array}} \qquad \boxed{\begin{array}{l} w_{v1} = -5 \\ w_{v2} = 7 \\ c_v = 12 \end{array}}$$

If we place these nodes on opposite sides of the partition, the remainder node replacing them would be:

$$\boxed{\begin{array}{l} w_{u1} = 12 \\ w_{u2} = 13 \\ c_u = 25 \end{array}}$$

This would be a poor move as the residual imbalance (as measured by c_u) is unimproved. If, however, we placed u and v on the same side of the partition, the remainder node replacing them would be labeled:

$$\boxed{\begin{array}{l} w_{u1} = 2 \\ w_{u2} = 1 \\ c_u = 3 \end{array}}$$

This example illustrates that we must consider placing nodes both on the opposite and on the same side of the partition. In general, the remainder node u resulting from placing nodes u and v on opposite sides of the partition will be labeled with values:

$$\begin{aligned} w_{r1} &= w_{u1} - w_{v1} \\ w_{r2} &= w_{u2} - w_{v2} \\ \vdots &= \vdots \\ w_{rm} &= w_{um} - w_{vm} \\ c_r &= \sum_{j=1}^{m} |w_{rj}| \end{aligned}$$

If nodes u and v are placed on the same side of the partition, the remainder node will be labeled:

$$\begin{aligned} w_{r1} &= w_{u1} + w_{v1} \\ w_{r2} &= w_{u2} + w_{v2} \\ \vdots &= \vdots \\ w_{rm} &= w_{um} + w_{vm} \\ c_r &= \sum_{j=1}^{m} |w_{rj}| \end{aligned}$$

We now describe the differencing algorithm extended to the case of multi-criteria number partitioning:

Algorithm 2: MCKK

1. Label each node i, $i = 1, \ldots, n$ with the m values $w_{i1}, \ldots, w_{im}$ and, in addition, with the node cost label $c_i = \sum_{j=1}^{m} |w_{ij}|$. Place all nodes in the "live" list sorted from largest to smallest c_i values.

2. Let u be the top node on the list of "live" nodes. For all other nodes, v, on the "live" list, calculate:

   - the remainder node, uv_s, and its cost, $c(uv_s)$, resulting from placing u and v on the *same* side of the partition
   - the remainder node, uv_o, and its cost, $c(uv_o)$, resulting from placing u and v on the *opposite* side of the partition

3. Let v^* be the node yielding the smallest $c(uv_s)$ or $c(uv_o)$ value among all remainder nodes computed in step one. Let uv_x^* be the remainder node with the smallest $c()$ value, i.e. $c(uw_x^*)$ from step 2 (either uv_s^* or uv_o^*). Remove nodes u and v^* from the "live" list.

4. Merge the remainder node uv_x^* back into the "live" list by going up the list of "live" nodes sorted by c_i, compare $c(uv_x^*)$ to c_i until $c(uv_x^*) \leq c_i$. Merge uv_x^* just below node i in the list.

5. Repeat steps 2, 3, and 4 until only one node remains on the list. The label c_i of the last remaining node will be the value of the objective function.

At each iteration we place an arc between nodes u and v^*, and remove node v^* from the set "live". In this case we need to specify "opposite side" arcs and "same side" arcs. The partition can be found by two-coloring the resultant tree as in the KK algorithm, except following the rule that nodes joined by a "same side" arc must be the same color.

More computation is required by algorithm MCKK than that of KK since, at each iteration, we compute remainders between node u and all other nodes on the "live" list. Nonetheless, the complexity of the algorithm remains $\mathcal{O}(mn^2)$.

We can improve the efficiency of this algorithm somewhat by curtailing the search for node v^* using the following relation:

$$c(uv_x) \geq c(u) - c(v) \quad \text{for } x = s,\ o \tag{4}$$

This relationship can be shown quite easily. First consider the case where u and v are placed on opposite sides of the partition. We have:

$$c(u) = \sum_{j=1}^{m} |w_{uj}| \qquad c(v) = \sum_{j=1}^{m} |w_{vj}| \qquad c(uv_o) = \sum_{j=1}^{m} |w_{uj} - w_{vj}| \tag{5}$$

Comparing individual terms of each sum, we see that:

$$|w_{uj} - w_{vj}| \geq |w_{uj}| - |w_{vj}| \tag{6}$$

If w_{uj} and w_{vj} have the same sign, then the inequality becomes an equality, otherwise it holds as a strict inequality. For example, $|7 - 3| = |7| - |3|$ and $|7 - (-3)| > |7| - |-3|$. Since the inequality

holds for each term in the sum, it must hold for the sum. When u and v are on the same side of the partition we have:

$$c(u) = \sum_{j=1}^{m} |w_{uj}| \qquad c(v) = \sum_{j=1}^{m} |w_{vj}| \qquad c(uv_o) = \sum_{j=1}^{m} |w_{uj} + w_{vj}| \tag{7}$$

Again, comparing individual terms of each sum, we see that:

$$|w_{uj} + w_{vj}| \geq |w_{uj}| - |w_{vj}| \tag{8}$$

For example, $|7+3| > |7| - |3|$ and $|7+(-3)| = |7| - |-3|$. In step 2 of Algorithm 2 we are looking for the "best" node, v^*, to combine with node u. Suppose we have found a node, v^c, yielding a remainder with cost value $c(uv^c)$. Then we need not consider any additional node, v, satisfying

$$c(v) \leq c(u) - c(uv^c) \tag{9}$$

since combining nodes u and v cannot produce a remainder node with value lower than $c(uv^c)$. Since we maintain the "live" list sorted by decreasing $c()$, once a node meets this criteria, we can stop searching and declare that $v^* = v^c$. We have observed reasonably significant reductions in CPU times by curtailing the search for the best v^* to combine with u in this fashion.

3.2 Subsets of Equal Cardinality with MCNP

Consider the m dimensional MCNP problem of Section 3.1 and assume that the number of nodes, n, is even. The objective is to minimize the difference in absolute componentwise sums of the two resulting subsets. Now suppose an additional constraint is added to the problem requiring that the cardinality of each of the subsets is equal, i.e. a new constraint on the binary variables of the form

$$\sum_{i=1}^{n} x_i = \frac{n}{2} \tag{10}$$

In order to accommodate the constraint defined in (10), a simple adjustment is made in the node construction. For each of the n nodes include an $(m+1)^{\text{st}}$ dimension containing some value $w_{i(m+1)}$ such that

$$w_{i(m+1)} \gg \max_i c_i \tag{11}$$

To see how $w_{i(m+1)}$ would lead to equal sized subsets recall the MCNP objective function from (2) with the new dimension.

$$\min \quad \sum_{j=1}^{m+1} \left| \sum_{i=1}^{n} w_{ij} x_i - w_{ij}(1 - x_i) \right| \tag{12}$$

$$= \sum_{j=1}^{m} \left| \sum_{i=1}^{n} w_{ij} x_i - w_{ij}(1 - x_i) \right| + \left| \sum_{i=1}^{n} w_{i(m+1)} x_i - w_{i(m+1)}(1 - x_i) \right| \tag{13}$$

Observe that by defining $w_{i(m+1)}$ to be sufficiently large, the second term of the (13) will dominate the objective value. The minimum of (12) will therefore be achived when (10) is satisfied; we refer to this as the "BigM" approach to MCNP. If n is odd, "BigM" is still used, but will obviously lead to subsets of sizes $\lceil \frac{n}{2} \rceil$ and $\lfloor \frac{n}{2} \rfloor$.

4 Analysis of Covariance

4.1 General Model Form

Consider an experiment with response variable, y, and an uncontrollable, but measurable, variable, x, such that y has a linear relationship with x. The latter variable, x, is said to be a covariate or concomitant variable that, if unaccounted for, can inflate errors in hypothesis tests [10]. In order to adjust the response variable for covariates the analysis of covariance (ANCOVA) method can be applied.

Let y_{ij} be the j^{th} response to the i^{th} treatment and x_{ij} be the corresponding covariate measurement. For treatment effects τ_i, $i = 1, \ldots, a$ the standard general linear model used in ANCOVA is

$$y_{ij} = \mu + \tau_i + \beta(x_{ij} - \bar{x}) + \epsilon_{ij} \quad j = 1, \ldots, n \tag{14}$$

where errors $\epsilon_{ij} \sim N(0, \sigma^2)$. It is further assumed that $\sum_{i=1}^{a} \tau_i = 0$, that the covariate is not affected by the treatments, and that the linear relationship is parallel across treatment groups. Without the covariate, this model reduces to the standard one way analysis of variance (ANOVA). Note that model (14) is obviously not restricted to only a single covariate.

The goal of ANCOVA is to test for a significant nonzero effect for any of the a treatments in the experiment. This hypothesis is formally stated as

$$\begin{aligned} H_0 &: \tau_i = 0 \quad \forall\, i \\ H_1 &: \tau_i \neq 0 \quad \text{for some } i \end{aligned} \tag{15}$$

To test (15) an extra sum of squares approach is utilized. Under the null hypothesis the reduced model is

$$y_{ij} = \mu + \beta(x_{ij} - \bar{x}) + \epsilon_{ij} \tag{16}$$

4.2 Experimental Design for ANCOVA

The traditional experimental design methodologies assign subjects to treatment groups through randomization. In [11], however, it was shown that with two treatments and one covariate in an ANCOVA experiment nonrandom assignment was not a necessity. The method of [11] assigned treatments using "alternate ranks" of the concomitant variable and was shown to maintain the null distribution alpha levels. In [12], alternate ranks design was shown to slightly increase statistical power in the presence of highly skewed response variable distributions. We present a generalization of the alternate ranks method to account for multiple covariates as well as a MCNP based nonrandom assignment design.

4.2.1 Alternate Ranks Design

Consider a single covariate ANCOVA experiment in which the covariate is observed before treatment groups are assigned to either of the two treatment groups. Dalton and Overall [11] proposed ranking each subject on the basis of the covariate measurement. The highest value is assignmed to treatment one, second to treatment two, third to treatment two and fourth to treatment one. This pattern is 1221... and is continued until all of the subjects are assigned to a treatment. As discussed in [12], the extension for three treatments is straightforward with an assignment pattern of 123321..., but only the two treatment case is considered here.

The primary concern with nonrandom assignment is with regards to creating a bias in the resulting treatment effect estimates, but [11] verified through Monte Carlo experiments that the alternate ranks design yielded unbiased results. Maxwell et al. [13] further investigated the power and Type I error of the design confirming that the simulated test statistic distribution is in line with that generated from random assignment. It was also concluded in [13] that if information regarding the concomitant variable is available, then the alternative ranks design would yield the most powerful results. This result was confirmed and shown to be even more prevalent in [12] in the case of skewed response distributions.

All discussed results focused upon alternate ranks design with a single covariate, but this concept can be applied to the multiple covariate case. Suppose that the following general model is considered

$$y_{ij} = \mu + \tau_i + \beta_1(x_{ij1} - \bar{x}_1) + \cdots + \beta_m(x_{ijm} - \bar{x}_m) + \epsilon_{ij} \quad \begin{cases} i = 1, \ldots, a \\ j = 1, \ldots, n \end{cases} \tag{17}$$

In general, ANCOVA with multiple covariates assumes that there is no interaction between the concomitant variables. Also, as in the standard alternate ranks case, we will include the additional assumption that all covariate information can be measured before assignment to treatment groups. The covariate measurements can be viewed as m dimensional points, allowing for each subject to be assigned to either of the two treatment groups based upon an iterative closest pairs algorithm.

Algorithm 3: Nonrandom Assignment with Closest Pairs

1. Label each m dimensional covariate observation as points x_i
2. Select the closest unassigned points by Euclidean distance
3. Assign the point with the *larger* of Euclidean norms to treatment one, the other to treatment two.
4. Repeat step 2
5. Assign the point with the *smaller* of the Euclidean norms to treatment one, the other to two
6. Repeat 2 - 5 until no points remain

Note that this algorithm does not have the alternate ranks design as the one dimensional special case. With alternate ranks, the covariate measurements are initially sorted in descending order; this concept does not have a clear parallel in higher dimensions. So, the above algorithm should only be applied to multiple covariate experiments and standard alternate ranks design for the single covariate case. Despite this issue, nonrandom assignment with closest pairs still maintains the general ideas of the alternate ranks design by using Euclidean norms to determine final group assignment amongst the closest pair of available points.

4.2.2 MCNP Design

Dalton and Overall [11] note that alternate ranks design achieves increases in power by "equating not only group means but the entire pretest distribution within the treatmeant groups". While this is true, the convergence in distribution still may be slow, limiting the ability for the rank based design to improve the ANCOVA procedure. We propose an equal cardinality MCNP based method

for efficiently creating treatment groups with approximately equal covariate distributions. This goal is accomplished by constructing nodes for each covariate measurement containing information regarding the deviations from the population's central moments.

Consider the single factor ANCOVA experiment in which the covariate measurements, x_i, are available before the n subjects are assigned to either of the two treatment groups. The subjects will be partitioned so that the distributions of the subsets are approximately equal insofar as the first k central moments will be closely matched. For each observation create a k dimensional node with the j^{th} element of the i^{th} node determined as follows

$$x_{ij} = \text{sign}(x_i - \bar{x})|x_i - \bar{x}|^j \tag{18}$$

The use of the sign function and dependence upon the central moments will force a balance between measurements on each side of the distribution.

Each of the k node values will also have a weight applied to it that will serve as a scaling factor. The scaling factor for the j^{th} element of each node is determined by the corresponding empirical moment of the full set as

$$w_j = \frac{n}{\sum_{i=1}^{n} x_i^j} \tag{19}$$

Scaling vector w acts as a normalizing term to prevent higher moments from dominating node cost. With these nodes, the MCKK algorithm can be applied and will yield two subsets that have k approximately equal central moments; subsequently, these subsets are equal in distribution.

The MCNP based nonrandom assignment design easily scales to the multiple covariate case. Suppose m covariates are measured and are to be used as the criteria for assignment to either of the two treatment groups. ANCOVA assumptions dictate that these covariates are independent, so we partition the subjects by simultaneously balancing k marginal moments for each of the m dimensions. Nodes are constructed for each of the n subjects with the node containing mk elements. The node values are determined using (18) for the covariates. Therefore, the mk elements of the i^{th} node are calculated as

$$x_{ij\ell} = \text{sign}(x_{i\ell} - \bar{x}_\ell)|x_{i\ell} - \bar{x}_\ell|^j \quad \begin{cases} j = 1, \ldots, k \\ \ell = 1, \ldots, m \end{cases} \tag{20}$$

Just as was done in the single covariate case, we scale the node elements by their corresponding full set empirical moments. So, for the i^{th} node, the scaling factors are

$$w_{j\ell} = \frac{n}{\sum_{i=1}^{n} x_{i\ell}^j} \quad \begin{cases} j = 1, \ldots, k \\ \ell = 1, \ldots, m \end{cases} \tag{21}$$

i.e., the inverse of j^{th} empirical moment of covariate ℓ.

4.3 Numerical Results

The alternate ranks design, its multiple covariate extension, the moment matching MCNP and randomization were compared through use of simulated ANCOVA experiments. Alpha levels for each of the partitioning methods were tested to ensure that the selected Type I error rates were maintained when the nonrandom schemes were used. This process was repeated for one, two and three covariates. The simulation algorithms and a summary of the numerical results are presented.

4.3.1 Methodology

Recall the general form of the ANCOVA general linear model presented in (17); to simulate an experiment with this model there are a total of $m+1$ random variables that must be generated. The first m correspond to realizations of the independent covariate values and the last is the random error, $\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$. The value σ_ϵ^2 as well as the group values τ_i are taken to be constant; in the experiments here, only two groups are used. The response variable, y_{ij}, is calculated using these random variables and constants. Parameters μ and β_i are preset constants that represent a translation of the data and, as a result, do not change the analysis. Due to this fact the numerical experiments here will set them to zero and one respectively.

For each simulated experiment the covariates will be sampled from a prespecified distribution with cumulative density function, F, and partitioned by each of the three methods. For F, a skewed distribution was tested for two reasons. First, since real data is highly likely to have a nonzero kurtosis [12]. Secondly, the alternate ranks design was shown to be the most powerful test for the skewed distribution case, so this represents the appropriate setting for comparisons between the two nonrandom assignment techniques. The resulting groups will be used in model (17) along with the independent random errors, ϵ_{ij}, and fixed τ_i to generate the response variables. With the simulated responses, the ANCOVA procedure will be applied to determine if a group difference in means exists. The complete algorithm for n total subjects with m covariates is shown in the following.

Algorithm 4: Simulated ANCOVA Experiment

1. Set constants σ_ϵ, τ_i, μ, β_ℓ
2. Generate n realizations of the m covariates $x_{i\ell} = F^{-1}(u_i)$ where $u_i \sim U(0,1)$, $i = 1, \ldots, n$, $\ell = 1, \ldots, m$
3. Produce two equal sized groups using each partition method
   - $x_{ij\ell}^M$ = element (i, j, ℓ) from the MCNP partition
   - $x_{ij\ell}^A$ = element (i, j, ℓ) from the alternate ranks/closest pairs
   - $x_{ij\ell}^R$ = element (i, j, ℓ) from the random partition
4. Generate n realizations of the random error, $\epsilon_{ij} \sim N(0, 1)$
5. Calculate the response variables
$$y_{ij}^{(p)} = \mu + \tau_i + \sum_{\ell=1}^{m} \beta_\ell x_{ij\ell}^{(p)} + \epsilon_{ij} \quad \begin{cases} i = 1, 2 \\ j = 1, \ldots, n/2 \\ p \in \{M, A, R\} \end{cases}$$
6. Apply the ANCOVA procedure to all sets $\left(Y^{(p)}, X_1^{(p)}, \ldots, X_m^{(p)}\right)$ to test for a significant treatment affect

This algorithm was repeated $N = 100,000$ times so that rejection results can be tabulated for each partitioning method and aggregated to calculate the power of the test. The difference in group means were varied over $[0, 2]$ in increments of 0.125. When all $\tau_i = 0$ the statistical power simulation represents an estimate of the Type I error rate; in the experiments, this value was set to 5%. In

each experiment the random error parameter was $\sigma_\epsilon = 0.75$. As was previously mentioned, $\mu = 0$ and $\beta_\ell = 1$.

For each experiment, all covariates were assumed to have the same underlying distribution. The skewed distribution used had $x_{ij\ell}$ selected from the intersection of $N(0, \sigma_x = 3) \cap [0, 5.88]$. This set represents the nonnegative values of a $N(\mu = 0, \sigma_x = 3)$ distribution excluding the upper 2.5% tail to avoid having the experiment influenced by extreme outliers. These values were generated using the Box-Muller transform [15] and an accept-reject methodology to enforce to the intervals $[0, 5.88]$. The power was estimated for the one, two and three covariate cases using $n = \{10, 20, 30, 40\}$ total subjects.

4.3.2 Simulation Results

Consider the $n = 10$ subjects case in which the covariates were sampled from the skewed distribution described in Section 4.3.1 and partitioned into two equal sized groups. The total statistical power results are in Table 1 with random, alternate ranks and MCNP partitioning for one, two and three factors.

τ	One Covariate			Two Covariates			Three Covariates		
	Rand	AltRank	MCNP	Rand	AltRank	MCNP	Rand	AltRank	MCNP
0.000	4.89%	4.85%	4.98%	5.01%	4.92%	5.01%	5.02%	5.03%	5.07%
0.125	5.58%	5.68%	5.67%	5.61%	5.50%	5.49%	5.40%	5.55%	5.46%
0.250	7.21%	7.48%	7.39%	6.81%	7.06%	7.21%	6.52%	6.80%	7.10%
0.375	9.87%	10.51%	10.48%	8.93%	9.93%	10.10%	8.36%	9.24%	9.76%
0.500	14.00%	15.04%	14.92%	12.45%	13.84%	14.37%	10.78%	12.41%	13.18%
0.625	19.00%	20.61%	20.61%	16.53%	18.86%	19.69%	14.24%	16.75%	18.09%
0.750	25.24%	27.55%	27.49%	21.99%	24.97%	26.07%	18.35%	21.86%	23.70%
0.875	32.18%	35.41%	35.47%	27.90%	32.03%	33.54%	23.17%	28.15%	30.67%
1.000	40.25%	44.08%	44.17%	34.65%	40.08%	41.72%	28.64%	34.58%	37.97%
1.125	48.52%	52.96%	52.88%	41.64%	48.11%	50.46%	34.55%	41.91%	45.53%
1.250	56.87%	61.82%	61.69%	48.97%	56.17%	58.63%	40.95%	49.24%	53.55%
1.375	64.59%	69.82%	69.66%	56.41%	64.12%	66.49%	47.29%	56.70%	61.56%
1.500	71.84%	77.17%	77.15%	63.39%	71.43%	74.02%	53.80%	63.77%	68.68%
1.625	78.30%	83.27%	83.18%	70.03%	77.98%	80.41%	59.95%	70.29%	75.27%
1.750	83.74%	88.11%	88.13%	75.52%	83.30%	85.46%	65.32%	76.10%	80.94%
1.875	88.22%	91.83%	91.96%	80.74%	88.10%	90.00%	70.70%	81.20%	85.74%
2.000	91.38%	94.67%	94.69%	84.87%	91.54%	93.10%	75.60%	85.72%	89.68%

Table 1: For $n = 10$, the statistical power using MCNP exceeds that of random and alternate ranks based partitioning. The power improvement with MCNP increases as additional covariates are included in the experiment

It is important to note that the Type I error rates found in the first row, $\tau = 0$, are maintained at the 95% confidence level for all three methods. The size of the confidence intervals range from $\pm 0.13\%$ for $\tau = 0$ up to $\pm 0.31\%$ for power values near 50%. As was observed in [11], the alternate ranks design produces Type I errors in line with that of random assignment. This holds true for MCNP and the closest pairs extension. As τ increases, the statistical power using alternate ranks is slightly higher than that of randomization, but the increase is larger still for MCNP. With the

addition of more covariates in the model, the power enhancement offered by the MCNP partitioning becomes more pronounced. Figure 1 shows the amount of power increase over randomization for each covariate count.

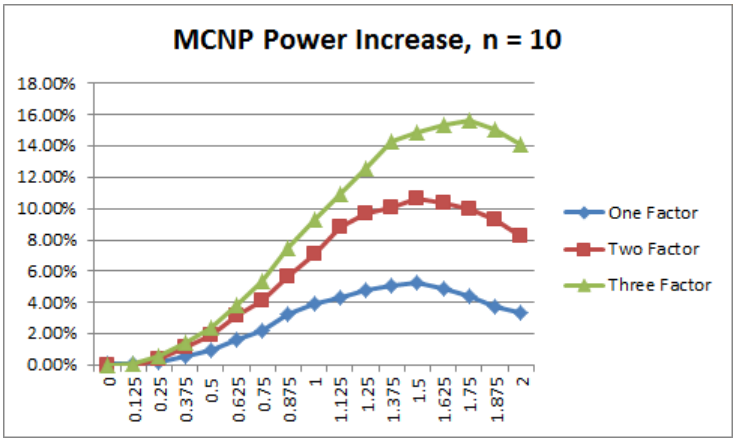

Figure 1: For $n = 10$, the increase in power offered by MCNP over standard randomization grows with the inclusion of additional covariates.

When the pool of subjects increases, the gap in power between methods decreases due to the asymptotic properties of randomization. This result is to be expected. Despite this, the outperformance of MCNP is still evident for larger sample sizes. For example, consider the MCNP power increase for $n = \{20, 30, 40\}$ with a the three factor experiment in Figure 2.

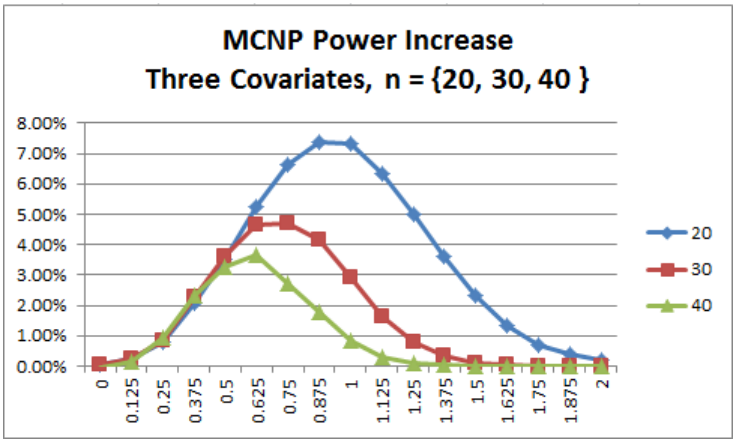

Figure 2: MCNP provides a persistent increase in statistical power, but, as is expected, this gain decreases as sample size, n, increases.

For a complete listing of power results for MCNP, alternate ranks and random design, see Appendix A. The power increase associated with MCNP shows large increases with additional covariates and smaller improvements with increases in n.

5 Conclusion

We have presented a generalization of the number partitioning problem (NP) that allows for the partitining of m dimensional nodes containing elements with no sign restrictions; this was referred to as the multi-criteria number partitioning problem (MCNP). To solve MCNP, a multi-criteria adaptation of the Karmarkar-Karp (KK) algorithm was developed (MCKK) that produces quality approximate solutions in $\mathcal{O}(mn^2)$. By adding an additional "BigM" dimension to each node, it was shown that subsets of equal cardinality can be produced using MCKK. Each of these results represents new extensions to the classical NP problem.

For observed random samples, it was demonstrated that MCNP can be used to partition the sample into subsets that are approximately equal in distribution. This was accomplished by defining k dimensional nodes for each of the observations. Each of the dimensions contained the scaled deviations of the corresponding datapoint from the central moments of the full sample.

The moment matching partitioning scheme proved to be extremely useful in the creation of treatment groups for analysis of covariance (ANCOVA) experiments. For experiments in which the covariate(s) could be measured before group assignment, MCNP based moment matching with equal subset sizes can be applied to generate nonrandom treatment groups. For the purpose of benchmarking with a well known nonrandom assignment method, the alternate ranks design of [11] was used. To compare multiple covariate experiments a generalization of the alternate ranks design that utilized an iterative closest pairs approach was presented.

The MCNP design yielded improvements in statistical power over both alternate ranks and random assignment that were particularly pronounced in small sample experiments as well as with multiple covariates. The gains in power dramatically increased with the number of covariates and decreased with larger sample sizes. Despite being nonrandom designs, both MCNP and alternate ranks/closest pairs maintained Type I error rates; this was verified for alternate ranks in [11] and again in [12], but only for a single covariate. We conclude that MCNP is an efficient nonrandom experimental design for ANCOVA that increases statistical power while maintaining desired Type I error levels.

Appendices

A Power Results for Section 4.3 Experiments

$n = 10$									
τ	One Covariate			Two Covariates			Three Covariates		
	Rand	AltRank	MCNP	Rand	AltRank	MCNP	Rand	AltRank	MCNP
0.000	4.89%	4.85%	4.98%	5.01%	4.92%	5.01%	5.02%	5.03%	5.07%
0.125	5.58%	5.68%	5.67%	5.61%	5.50%	5.49%	5.40%	5.55%	5.46%
0.250	7.21%	7.48%	7.39%	6.81%	7.06%	7.21%	6.52%	6.80%	7.10%
0.375	9.87%	10.51%	10.48%	8.93%	9.93%	10.10%	8.36%	9.24%	9.76%
0.500	14.00%	15.04%	14.92%	12.45%	13.84%	14.37%	10.78%	12.41%	13.18%
0.625	19.00%	20.61%	20.61%	16.53%	18.86%	19.69%	14.24%	16.75%	18.09%
0.750	25.24%	27.55%	27.49%	21.99%	24.97%	26.07%	18.35%	21.86%	23.70%
0.875	32.18%	35.41%	35.47%	27.90%	32.03%	33.54%	23.17%	28.15%	30.67%
1.000	40.25%	44.08%	44.17%	34.65%	40.08%	41.72%	28.64%	34.58%	37.97%
1.125	48.52%	52.96%	52.88%	41.64%	48.11%	50.46%	34.55%	41.91%	45.53%
1.250	56.87%	61.82%	61.69%	48.97%	56.17%	58.63%	40.95%	49.24%	53.55%
1.375	64.59%	69.82%	69.66%	56.41%	64.12%	66.49%	47.29%	56.70%	61.56%
1.500	71.84%	77.17%	77.15%	63.39%	71.43%	74.02%	53.80%	63.77%	68.68%
1.625	78.30%	83.27%	83.18%	70.03%	77.98%	80.41%	59.95%	70.29%	75.27%
1.750	83.74%	88.11%	88.13%	75.52%	83.30%	85.46%	65.32%	76.10%	80.94%
1.875	88.22%	91.83%	91.96%	80.74%	88.10%	90.00%	70.70%	81.20%	85.74%
2.000	91.38%	94.67%	94.69%	84.87%	91.54%	93.10%	75.60%	85.72%	89.68%
$n = 20$									
0.000	4.98%	5.06%	5.01%	4.98%	4.88%	4.94%	4.96%	5.02%	5.02%
0.125	6.33%	6.44%	6.46%	6.22%	6.27%	6.36%	6.02%	6.27%	6.28%
0.250	10.52%	10.83%	10.83%	10.18%	10.78%	10.84%	9.85%	10.48%	10.63%
0.375	17.51%	18.18%	18.19%	16.99%	17.99%	18.24%	16.02%	17.59%	18.07%
0.500	27.94%	29.02%	29.12%	26.41%	28.33%	28.88%	24.91%	27.40%	28.43%
0.625	40.17%	41.86%	41.84%	38.13%	41.12%	41.72%	36.07%	39.93%	41.31%
0.750	53.60%	55.94%	55.87%	51.13%	54.76%	55.55%	48.36%	53.35%	55.02%
0.875	66.87%	69.13%	69.14%	63.72%	67.77%	68.72%	60.58%	66.19%	67.95%
1.000	78.05%	80.30%	80.22%	75.07%	79.15%	79.80%	71.87%	77.52%	79.20%
1.125	86.41%	88.39%	88.37%	84.09%	87.53%	88.08%	81.44%	86.32%	87.77%
1.250	92.60%	93.94%	93.95%	90.61%	93.28%	93.71%	88.47%	92.38%	93.45%
1.375	96.20%	97.07%	97.04%	94.88%	96.76%	97.02%	93.09%	96.12%	96.73%
1.500	98.28%	98.83%	98.82%	97.36%	98.52%	98.69%	96.27%	98.27%	98.61%
1.625	99.21%	99.53%	99.53%	98.74%	99.43%	99.49%	98.12%	99.28%	99.46%
1.750	99.68%	99.83%	99.84%	99.40%	99.78%	99.80%	99.09%	99.72%	99.81%
1.875	99.89%	99.95%	99.95%	99.75%	99.94%	99.94%	99.56%	99.90%	99.93%
2.000	99.96%	99.99%	99.99%	99.92%	99.98%	99.98%	99.78%	99.97%	99.98%

$n = 30$									
τ	One Covariate			Two Covariates			Three Covariates		
	Rand	AltRank	MCNP	Rand	AltRank	MCNP	Rand	AltRank	MCNP
0.000	5.00%	4.98%	4.98%	5.07%	5.07%	5.06%	5.05%	5.16%	5.11%
0.125	7.16%	7.26%	7.27%	7.05%	7.19%	7.21%	7.06%	7.24%	7.30%
0.250	14.00%	14.30%	14.34%	13.65%	14.21%	14.41%	13.34%	14.11%	14.19%
0.375	25.52%	26.22%	26.24%	24.54%	25.95%	26.08%	23.72%	25.55%	25.98%
0.500	40.79%	41.99%	41.98%	39.45%	41.65%	41.98%	38.11%	41.04%	41.71%
0.625	58.18%	59.72%	59.76%	56.61%	59.42%	59.67%	54.55%	58.25%	59.23%
0.750	73.72%	75.27%	75.21%	71.99%	74.65%	75.05%	70.05%	73.94%	74.74%
0.875	85.64%	86.81%	86.83%	84.29%	86.48%	86.79%	82.52%	85.92%	86.68%
1.000	93.30%	94.17%	94.16%	92.23%	93.90%	94.08%	91.02%	93.47%	93.95%
1.125	97.30%	97.77%	97.77%	96.66%	97.63%	97.70%	95.98%	97.43%	97.60%
1.250	99.00%	99.23%	99.22%	98.78%	99.20%	99.24%	98.40%	99.06%	99.20%
1.375	99.73%	99.82%	99.81%	99.61%	99.78%	99.80%	99.43%	99.75%	99.79%
1.500	99.93%	99.96%	99.96%	99.89%	99.95%	99.95%	99.83%	99.95%	99.95%
1.625	99.99%	99.99%	99.99%	99.97%	99.99%	99.99%	99.95%	99.99%	99.99%
1.750	100.00%	100.00%	100.00%	99.99%	100.00%	100.00%	99.98%	100.00%	100.00%
1.875	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
2.000	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
$n = 40$									
0.000	4.91%	4.94%	4.93%	5.02%	4.97%	4.95%	4.91%	4.90%	4.85%
0.125	8.05%	8.14%	8.13%	8.01%	8.08%	8.15%	7.79%	7.98%	7.94%
0.250	17.29%	17.64%	17.62%	16.76%	17.58%	17.59%	16.67%	17.53%	17.63%
0.375	32.74%	33.58%	33.61%	32.40%	33.57%	33.74%	31.42%	33.43%	33.75%
0.500	52.65%	53.71%	53.70%	51.45%	53.52%	53.66%	50.15%	52.88%	53.42%
0.625	71.73%	72.89%	72.95%	70.27%	72.31%	72.53%	69.13%	72.04%	72.78%
0.750	86.09%	86.97%	86.96%	84.99%	86.65%	86.83%	83.87%	86.16%	86.59%
0.875	94.36%	94.92%	94.90%	93.70%	94.75%	94.81%	93.02%	94.55%	94.79%
1.000	98.16%	98.43%	98.44%	97.83%	98.32%	98.35%	97.52%	98.23%	98.38%
1.125	99.50%	99.60%	99.59%	99.39%	99.61%	99.63%	99.30%	99.57%	99.62%
1.250	99.90%	99.93%	99.92%	99.86%	99.92%	99.93%	99.83%	99.91%	99.93%
1.375	99.98%	99.99%	99.99%	99.98%	99.99%	99.99%	99.95%	99.99%	99.99%
1.500	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.99%	100.00%	100.00%
1.625	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
1.750	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
1.875	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
2.000	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

References

[1] Karmarkar, Narenda, and Richard Karp. "The Differencing Method of Set Partitioning." Technical Report UCB/CSD 81/113, Computer Science Division, University of California, Berkeley (1982).

[2] Korf, Richard E. "A Complete Anytime Algorithm for Number Partitioning." *Artificial Intelligence* 106.2 (1998): 181-203. Print.

[3] Aragon, C. R., D. S. Johnson, L. A. McGeoch, and C. Schevon. "Optimization by Simulated Annealing: An Experimental Evaluation; Part II, Graph Coloring and Number Partitioning." *Operations Research* 39.3 (1991): 378-406. Print.

[4] Garey, Michael R., and David S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-completeness*. San Francisco: W.H. Freeman, 1979. Print.

[5] Flanders, Seth W., Robert H. Storer, and S. David Wu. "Problem Space Local Search for Number Partitioning." *Annals of Operations Research* 63.4 (1996): 463-87. Print.

[6] Argüello, M., T. A. Feo, and O. Goldschmidt. "Randomized Methods for the Number Partitioning Problem." *Computers & Operations Research* 23.2 (1996): 103-11. Print.

[7] Choi, Wonjoon, and Robert H. Storer. "Heuristic Algorithms for a Turbine-blade-balancing Problem." *Computers & Operations Research* 31.8 (2004): 1245-258. Print.

[8] James, R., and R. H. Storer. "Techniques for Solving the Subset Sum Problem." *International Transactions in Operational Research* 12.4 (2005): 437-53. Print.

[9] Hayes, Brian. "The Easiest Hard Problem." *American Scientist* 90.2 (2002): 113-17. Print.

[10] Montgomery, Douglas C. "The Analysis of Covariance." *Design and Analysis of Experiments*. 5th ed. New York: John Wiley, 2001. 604-12. Print.

[11] Dalton, S., and J. E. Overall. "Nonrandom Assignment in ANCOVA: The Alternate Ranks Design." *The Journal of Experimental Education* 46: 58-62. Print.

[12] Klockars, Alan J., and Mary J. McAweeney. "Maximizing Power in Skewed Distributions: Analysis and Assignment." *Psychological Methods* 3.1 (1998): 117-22. Print.

[13] Delaney, Harold D., Charles A. Dill, and Scott E. Maxwell. "Another Look at ANCOVA versus Blocking." *Psychological Bulletin* 95.1 (1984): 136-47. Print.

[14] Glass, G. V., P. D. Peckham, and J. R. Sanders. "Consequences of Failure to Meet Assumptions Underlying the Analysis of Variance and Covariance." *Review of Educational Research* 42: 237-88.

[15] Box, G. E. P., and Mervin E. Muller. "A Note on the Generation of Random Normal Deviates." *The Annals of Mathematical Statistics* 29.2 (1958): 610-11. Print.