Mixed Integer Programming Models

Portfolios with Combinatorial Constraints

Sirong Luo

Faculty of Statistics and Data Science
Shanghai University of Finance and Economics

Table of Contents

Combinatorial Auctions

- A *combinatorial auction* is an auction that involves the concurrent sale of multiple items. Examples include Federal Communications Commission (FCC) spectrum auctions, electricity markets, pollution right auctions, and auctions for airport landing slots. In these kinds of auctions, bidders have preferences for sets of items usually called *bundles*. The value that a bidder has for a bundle may not necessarily be equal to the sum of the values that the bidder has for individual items in the bundle. To take the bidders' preferences into consideration, combinatorial auctions allow bidders to submit bids on combinations of items.

Combinatorial Auctions

- Specifically, let M be the set of items that the auctioneer has to sell and N the set of bidders. A *bid* is a pair $(S, b_j(S))$ where $S \subseteq M$, for $j \in N$, and $b_j(S)$ is the price that bidder j is willing to pay for the bundle S. The *combinatorial auction problem or winner selection problem* is the problem of identifying which bids should be accepted to maximize the auctioneer's revenue. This problem can be formulated as a binary linear program.

Combinatorial Auctions

- *Binary linear programming model for the combinatorial auction problem*

  Variables:

$$x(S,j) = \begin{cases} 1 & \text{if bundle } S \text{ is allocated to bidder } j \\ 0 & \text{otherwise} \end{cases}$$

  for $S \subseteq M, j \in N$.

  Objective:

$$\max \sum_{S \subseteq M} \sum_{j \in N} b_j(S) x(S,j).$$

Combinatorial Auctions

- *Binary linear programming model for the combinatorial auction problem*
  Constraints:

$$\sum_{S \subseteq M : i \in S} \sum_{j \in N} x(S,j) \leq 1 \text{ for } i \in M$$

$$x(S,j) \in \{0,1\} \text{ (allocated bundles do not overlap).}$$

$$\text{for } S \subseteq M, j \in N$$

$$\text{(binary variables).}$$

Combinatorial Auctions

- In some combinatorial auctions, bidders are awarded at most one of the bundles that they bid on, even when these bundles are disjoint. This is easy to model by adding the constraint

$$\sum_{S \subseteq M} x(S, j) \leq 1 \quad \text{for } j \in N \text{ (each bidder receives at most one bundle).}$$

Combinatorial Auctions

- For example, if there are four items for sale and the following bids have been received:
  $B_1 = (\{1\}, 6), B_2 = (\{2\}, 3), B_3 = (\{3, 4\}, 12), B_4 = (\{1, 3\}, 12)$,
  $B_5 = (\{2, 4\}, 8), B_6 = (\{1, 3, 4\}, 16)$, the winners can be determined by the following integer program:

$$
\begin{array}{lll}
\max & 6x_1 + 3x_2 + 12x_3 + 12x_4 + 8x_5 + 16x_6 & \\
\text{s.t.} & x_1 + x_4 + x_6 \leq 1 & \text{(item 1 is allocated at most once)} \\
 & x_2 + x_5 \leq 1 & \text{(item 2 is allocated at most once)} \\
 & x_3 + x_4 + x_6 \leq 1 & \text{(item 3 is allocated at most once)} \\
 & x_3 + x_5 + x_6 \leq 1 & \text{(item 4 is allocated at most once)} \\
 & x_j \in \{0, 1\} \text{ for } j = 1, \ldots, 6. &
\end{array}
$$

- If bids B_4 and B_5 come from the same bidder who wants at most one of these two bundles, it suffices to add the constraint

$$x_4 + x_5 \leq 1.$$

Combinatorial Auctions

- If there are multiple units u_i of each item $i \in M$, then a bid can be more broadly defined as a pair $(\lambda, b_j(\lambda))$ where λ is an M-vector with entries $\lambda_i \in \{0, 1, \ldots, u_i\}$, $i \in M$, that indicates the desired number of units λ_i of each item $i \in M$. Let Λ denote the set of all these M-vectors. The previous model is replaced by

$$\max \sum_{\lambda \in \Lambda} \sum_{j \in N} b_j(\lambda) x(\lambda, j)$$

$$\text{s.t.} \quad \sum_{\lambda \in \Lambda} \sum_{j \in N} \lambda_i x(\lambda, j) \leq u_i \text{ for } i \in M$$

$$x(\lambda, j) \in \{0, 1\} \text{ for } \quad \lambda \in \Lambda, j \in N$$

Combinatorial Auctions

- There are further variations of the above formulations that incorporate additional features such as constraints on the kinds of bids the auctioneer accepts and constraints on the kinds of bundles that can be allocated to bidders (De Vries and Vohra, 2003). The gist of these models is essentially the same as those discussed above.

The Lockbox Problem

- Consider a national firm that receives checks from all over the United States. Due to the vagaries of the Postal Service, as well as the banking system, there is a variable delay from when the check is postmarked (and hence the customer has met her obligation) and when the check clears (and when the firm can use the money). For instance, a check mailed in Pittsburgh sent to a Pittsburgh address might clear in just two days. A similar check sent to Los Angeles (L.A.) might take four days to clear. It is in the interest of the firm to have the check clear as quickly as possible since then the firm can use the money. In order to speed up this clearing, firms open offices (called lockboxes) in different cities to handle the checks.

The Lockbox Problem

- **Example 9.1** Suppose we receive payments from four regions (West, Midwest, East, and South). The average daily value from each region is as follows: $\$300,000$ from the West, $\$120,000$ from the Midwest, $\$360,000$ from the East, and $\$180,000$ from the South. We are considering opening lockboxes in L.A., Cincinnati, Boston, and/or Houston. Operating a lockbox costs $\$90,000$ per year. The average days from mailing to clearing is given in Table 9.1. Which lockboxes should we open?

From	L.A.	Cincinnati	Boston	Houston
West	2	4	6	6
Midwest	4	2	5	5
East	6	5	2	5
South	7	5	6	3

Table 9.1 Clearing times

The Lockbox Problem

- First we must calculate the losses due to lost interest for each possible assignment. For example, if the West sends to Boston, then on average there will be $\$1,800,000 (= 6 \times \$300,000)$ in process on any given day. Assuming an investment rate of 10%, this corresponds to a yearly loss of $\$180,000$. We can calculate the losses for the other possibilities in a similar fashion to get Table 9.2.

From	L.A.	Cincinnati	Boston	Houston
West	60	120	180	180
Midwest	48	24	60	60
East	216	180	72	180
South	126	90	108	54

Table 9.2 Lost interest ('000)

- The intuition for the formulation of the lockbox problem is similar to that of the formulation of the K-median model for clustering discussed in Chapter 8. We use a set of binary variables to model the lockboxes to open and another set of binary variables to model what lockbox serves each region.

The Lockbox Problem

- *Binary linear programming model for the lockbox problem*
  Variables:

$$y_j = \begin{cases} 1 & \text{if lockbox } j \text{ is opened} \\ 0 & \text{otherwise} \end{cases} \quad \text{for } j = 1, \ldots, 4$$

$$x_{ij} = \begin{cases} 1 & \text{if region } i \text{ is served by lockbox } j \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i, j = 1, \ldots, 4.$$

Objective: minimize total yearly costs

$$\min \sum_{i,j=1}^{4} c_{ij} x_{ij} + 90 \sum_{j=1}^{4} y_j,$$

where c_{ij} is the (i, j) entry in Table 9.2 .

- *Binary linear programming model for the lockbox problem*

Constraints:

$$\sum_{j=1}^{N} x_{ij} = 1, \quad \text{for } i = 1, \ldots, 4 \quad \text{(each region must be assigned to one lockbox)}$$

$$x_{ij} \leq y_j, \quad \text{for } i = 1, \ldots, 4 \quad \text{(region } i \text{ is assigned to lockbox } j \text{ only if } j \text{ is opened)}$$

$$x_{ij}, y_j \in \{0, 1\} \quad \text{for } i = 1, \ldots, 4 \quad \text{(binary variables).}$$

The Lockbox Problem

- As we observed for the binary programming formulation for clustering discussed in Example 8.2, the above formulation would also be correct if we replaced the $4^2 = 16$ constraints

$$x_{ij} \leq y_j, i, j = 1, \ldots, 4$$

with the four constraints

$$\sum_{i=1}^{4} x_{ij} \leq 4y_j, j = 1, \ldots, 4$$

- However, we should note that the solution to the linear programming relaxation of the first formulation (with more constraints) is

$$x_{11} = x_{21} = x_{33} = x_{43} = y_1 = y_3 = 1, \text{ and all other variables zero,}$$

which has binary entries and hence is an optimal solution to the binary linear programming model. Therefore the firm should open two lockboxes, one in the Eastern region and one in the West.

The Lockbox Problem

- By contrast, the solution to the linear programming relaxation of the second formulation (with fewer constraints) is

$$x_{11} = x_{22} = x_{33} = x_{44} = 1, \quad y_1 = y_2 = y_3 = y_4 = 0.25,$$

and all other variables zero,

which does not give any useful information about the binary linear programming model.
This example highlights how different equivalent integer programming formulations can have very different properties with respect to their associated linear program.

Constructing an Index Fund

An old and recurring debate about investing lies in the merits of active versus passive management of a portfolio. Active portfolio management tries to achieve superior performance by using technical and fundamental analysis. On the other hand, passive portfolio management relies entirely on diversification to achieve a desired performance. There are two types of passive management strategies: "buy and hold" or "indexing". In the first one, assets are selected on the basis of some fundamental criteria and there is no active selling or buying of these stocks afterwards (see the chapters on dedication (Chapter 3) and portfolio optimization (Chapter 6)). In the second approach, absolutely no attempt is made to identify mispriced securities. The goal is to choose a portfolio that mirrors the movements of a broad market population or a market index. Such a portfolio is called an index fund. Given a target population of n stocks, one selects K stocks (and their weights in the index fund) to represent the target population as closely as possible.

Constructing an Index Fund

In the last 30 years, an increasing number of investors, both large and small, have established index funds. Simply defined, an index fund is a portfolio designed to track the movement of the market as a whole or some selected broad market segment. The rising popularity of index funds can be justified both theoretically and empirically.

- **Market efficiency:** If the market is efficient, no superior risk-adjusted returns can be achieved by stock picking strategies since the prices reflect all the information available in the marketplace. Additionally, since the market portfolio provides the best possible return per unit of risk, to the extent that it captures the efficiency of the market via diversification, one may argue that the best theoretical approach to fund management is to invest in an index fund.

Constructing an Index Fund

- **Empirical performance:** Considerable empirical literature provides strong evidence that, on average, money managers have consistently underperformed the major indices. In addition, studies show that, in most cases, top performing funds for a year are no longer amongst the top performers in the following years, leaving room for the intervention of luck as an explanation for good performance.
- **Transaction cost:** Actively managed funds incur transaction costs, which reduce the overall performance of these funds. In addition, active management implies significant research costs. Finally, fund managers may have costly compensation packages that can be avoided to a large extent with index funds.

Constructing an Index Fund

- Here we take the point of view of a fund manager who wants to construct an index fund. Strategies for forming index funds involve choosing a broad market index as a proxy for an entire market, e.g. the Standard & Poor's list of 500 stocks (S&P 500). A pure indexing approach consists in purchasing all the issues in the index, with the same exact weights as in the index. In most instances, this approach is impractical (many small positions) and expensive (rebalancing costs may be incurred frequently). An index fund with K stocks, where K is substantially smaller than the size n of the target population, seems desirable. The clustering approach introduced in Chapter 8 can be used to aggregate the stocks in a broad market into a smaller more manageable index fund. This approach will not necessarily yield mean/variance-efficient portfolios but will produce a portfolio that closely replicates the underlying market population.

- We describe a two-step heuristic approach for constructing an index fund. First, select K stocks to be included in the portfolio. Second, determine weights for these stocks so that the portfolio is as close as possible to the benchmark. The motivation for this two-step approach is that each of the stocks selected in the portfolio is a proxy for a portion of stocks in the index.

  The first step, that is, the selection of the K stocks to be included in the portfolio, can be formulated as the binary linear programming formulation for clustering for Example 8.2 in Chapter 8. Recall that the model is based on the following data:

$$\rho_{ij} = \text{ similarity between stock } i \text{ and stock } j.$$

  An example of this is the correlation between the returns of stocks i and j. But one could choose other similarity measures ρ_{ij}.

Constructing an Index Fund

- Recall the binary linear programming formulation for the clustering problem:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{j=1}^{n} y_j = K \\
& \sum_{j=1}^{n} x_{ij} = 1 && \text{for } i = 1, \ldots, n \\
& x_{ij} \leq y_j && \text{for } i, j = 1, \ldots, n \\
& x_{ij}, y_j \in \{0, 1\} && \text{for } i, j = 1, \ldots, n
\end{aligned}
$$

Constructing an Index Fund

- As discussed in Chapter 8, the variables y_j describe which stocks j are in the portfolio ($y_j = 1$ if j is selected in the portfolio, 0 otherwise). For each stock $i = 1, \ldots, n$, the variable x_{ij} indicates which stock j in the portfolio is most similar to i ($x_{ij} = 1$ if j is the most similar stock in the portfolio, 0 otherwise).

Constructing an Index Fund

- Once the set of K stocks has been selected, a simple approach to the second step of portfolio construction is as follows. Assume $j_1, \ldots, j_K$ are the selected stocks and $C_1, \ldots, C_K$ are the corresponding clusters. That is, C_ℓ is the set of stocks represented by stock j_ℓ for $\ell = 1, \ldots, K$. Set the weight of each selected stock $j\ell, \ell = 1, \ldots, K$, proportional to the total market capitalization of the stocks in C_ℓ :

$$x_{j_\ell} := \frac{\sum_{i \in C_\ell} V_i}{\sum_{i=1}^{n} V_i}, \qquad \ell = 1, \ldots, K,$$

where V_i is the market capitalization of stock i.

Constructing an Index Fund

- The second step can alternatively be tackled via a linear or a quadratic programming model. The variables in the model are the portfolio weights on the selected stocks. A reasonable objective is to minimize a measure associated with the quality of tracking such as active risk - this would lead to a quadratic programming problem. Alternatively, one can minimize mean absolute deviation and obtain a linear programming problem. The constraints could include bounds on beta, sector exposures, and other attributes to find weights of the selected stocks so that the portfolio is as close as possible to the benchmark.

Cardinality Constraints

- In this section, we present a different approach for tracking a basket of assets, e.g., an index, with a small group of stocks. In contrast to the two-step approach in the previous section, we model the index replication problem in one step, as a mixed integer programming problem with cardinality constraints.

Cardinality Constraints

- For concreteness, consider that case when we want to track a benchmark with a portfolio containing a predetermined maximum number of stocks. Assume $\mathbf{x}_B$ is the vector of holdings in the benchmark, and $\mathbf{x}$ is the vector of holdings in the portfolio. Suppose we want to include at most K stocks in the tracking portfolio. If we have an estimate of the covariance matrix of the universe of stocks in the index, then the problem can be informally stated as follows:

$$\begin{aligned} \min\, & (\mathbf{x} - \mathbf{x}_B)^\top \mathbf{V} (\mathbf{x} - \mathbf{x}_B) \\ \text{s.t. }\, & \mathbf{1}^\top \mathbf{x} = 1 \\ & \mathbf{x} \geq 0 \\ & x_j > 0 \quad \text{for at most } K \text{ distinct } j = 1, \ldots, n. \end{aligned} \tag{9.1}$$

Cardinality Constraints

- In order to model the problem formally, we introduce a new set of binary variables whose role is to model the logical condition of whether each particular stock is included in the portfolio:

$$y_j = \begin{cases} 1 & \text{if } x_j > 0 \\ 0 & \text{otherwise.} \end{cases}$$

- The problem (9.1) can be reformulated as the following mixed quadratic program:

$$
\begin{aligned}
\min\ & (\mathbf{x}-\mathbf{x}_B)\,\mathbf{V}\,(\mathbf{x}-\mathbf{x}_B) \\
\text{s.t. } & 1^\top \mathbf{x} = 1 \\
& x_j \le y_j \quad \text{for } j = 1, \dots, n \\
& \sum_{j=1}^{n} y_j \le K \\
& \mathbf{x} \ge 0 \\
& y_j \in \{0, 1\} \quad \text{for } j = 1, \dots, n.
\end{aligned}
\tag{9.2}
$$

Observe that the linking constraint $x_j \le y_j$ in (9.2) is a mathematical way of encoding the logical connection between x_j and y_j : the variable x_j is positive only when $y_j = 1$. Furthermore, the constraint $\sum_{j=1}^{n} y_j \le K$ enforces the condition that at most K of the $\mathbf{x}$ variables are positive.

Cardinality Constraints

- Consider now a more general mean-variance model with cardinality constraints where we now allow short positions:

$$\begin{aligned} &\min \mathbf{x}^\top \mathbf{V}\mathbf{x} \\ \text{s.t.}\ &\boldsymbol{\mu}^\top \mathbf{x} \geq \bar{\mu} \\ &\mathbf{A}\mathbf{x} = \mathbf{b} \\ &\mathbf{C}\mathbf{x} \geq \mathbf{d} \\ &x_j \neq 0 \quad \text{for at most } K \text{ distinct } j = 1, \ldots, n. \end{aligned} \tag{9.3}$$

The above approach extends provided that there is a lower bound ℓ_j and an upper bound u_j on the value of each holding x_j for $j = 1, \ldots, n$. In this case the cardinality constraint can be formulated via a new set of binary variables $y_j, j = 1, \ldots, n$, together with the linking constraints

$$\ell_j y_j \leq x_j \leq u_j y_j \quad \text{for } j = 1, \ldots, n.$$

- The problem (9.3) can be reformulated as the following mixed quadratic program:

$$
\begin{aligned}
& \min \mathbf{x}^\top \mathbf{V}\mathbf{x} \\
\text{s.t.}\ & \boldsymbol{\mu}^\top \mathbf{x} \geq \bar{\mu} \\
& \mathbf{A}\mathbf{x} = \mathbf{b} \\
& \mathbf{C}\mathbf{x} \geq \mathbf{d} \\
& \ell_j y_j \leq x_j \leq u_j y_j \quad \text{for } j = 1, \ldots, n \\
& \sum_{j=1}^{n} y_j \leq K \\
& y_j \in \{0, 1\} \quad \text{for } j = 1, \ldots, n
\end{aligned}
\tag{9.4}
$$

Minimum Position Constraints

- The same kind of linking constraints $\ell_j y_j \leq x_j \leq u_j y_j$ used in the mixed binary programming formulation (9.4) can be used to enforce another common practical consideration: minimum position constraints. Although diversification into a broad universe of assets generally has merits, there is a potential downside: some positions may become very small. Too many small positions typically generate higher research and monitoring costs. Consequently, investment managers enforce minimum position constraints. This means that if a stock j is included in the portfolio, then the holding x_j in that stock must surpass a minimum threshold $\ell_j > 0$.

Minimum Position Constraints

- Consider a general mean-variance model with minimum position constraints:

$$
\begin{aligned}
& \min \mathbf{x}^\top \mathbf{V} \mathbf{x} \\
\text{s.t. } & \boldsymbol{\mu}^\top \mathbf{x} \geq \bar{\mu} \\
& \mathbf{A}\mathbf{x} = \mathbf{b} \\
& \mathbf{C}\mathbf{x} \geq \mathbf{d} \\
& x_j > 0 \Rightarrow x_j \geq \ell_j \quad \text{for } j = 1, \ldots, n.
\end{aligned}
\tag{9.5}
$$

Minimum Position Constraints

- Provided there is an upper bound u_j on the value of each holding x_j, for $j = 1, \ldots, n$, and proceeding as above, the problem (9.5) can be reformulated as the following mixed quadratic program:

$$\begin{aligned} \min & \mathbf{x}^\top \mathbf{V} \mathbf{x} \\ \text{s.t. } & \boldsymbol{\mu}^\top \mathbf{x} \geq \bar{\mu} \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{C}\mathbf{x} \geq \mathbf{d} \\ & \ell_j y_j \leq x_j \leq u_j y_j \quad \text{for } j = 1, \ldots, n \\ & y_j \in \{0, 1\} \quad \text{for } j = 1, \ldots, n. \end{aligned} \tag{9.6}$$

Risk-Parity Portfolios and Clustering

- Consider a situation where you are trying to construct a risk-parity portfolio as explained in Section 7.5. We would like to allocate equal risk to a set of assets, but several of them may be very similar. By using the risk-parity strategy directly, you end up overweighting the characteristics those assets share. Instead, you can cluster the assets first, and then allocate risk evenly to each cluster. This is more consistent with the spirit of "risk parity". This first step can be accomplished using the clustering approach suggested in Example 8.2.