1. Introduction to Discrete Optimization

In finite dimensional optimization we are interested in locating solutions to the problem

$$\begin{array}{lll}\mathcal{P}: & \underset{x\in X}{\text{minimize}} & f_0(x) \\ & \text{subject to} & x \in \Omega.\end{array}$$

where X is the variable space (or decision space), $f_0 : X \to \mathbb{R} \cup \{\pm\infty\}$ is called the objective function, and the set $\Omega \subset X$ is called the constraint region. The techniques that one employs in the study of $\mathcal{P}$ are determined by the nature of the space X, the function f_0, and the set Ω. Some of the basic problem categories are as follows:

(1) Variable Type:
- (a) continuous variable: $X = \mathbb{R}^n$
- (b) discrete variable: $X = \mathbb{Z}^n$
- (c) zero-one variables: $X = \{0, 1\}^n$
- (d) mixed variable: $X = \mathbb{R}^r \times \mathbb{Z}^s \times \{0, 1\}^t$

(2) Constraint Type:
- (a) unconstrained: $\Omega = X$
- (b) constrained: $\Omega \neq X$

(3) Problem Type:
- (a) Convex Programming:

  f_0 is a convex function and Ω is a convex set.

  Definition 1.1. *The set $\Omega \subset \mathbb{R}^n$ is said to be convex if for every $x, y \in \Omega$ one has $[x, y] \subset \Omega$ where $[x, y]$ denotes the line segment connecting x and y:*

  $$[x, y] = \{\lambda x + (1 - \lambda)y : 0 \leq \lambda \leq 1\}.$$

  Definition 1.2. *The function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$ is said to be convex if the set epi$(f) = \{(x, \mu) : f(x) \leq \mu\}$ is a convex set in $\mathbb{R}^{n+1}$. In particular,*

  $$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

  for all $0 \leq \lambda \leq 1$ and points x, y for which not both $f(x)$ and $f(y)$ are infinite.

- (b) Linear Programming:

  The minimization or maximization of a linear functional subject to a finite number of linear inequality and/or equality constraints. $f_0(x) := c^T x$ for some $c \in \mathbb{R}^n$ and

  $$\Omega := \left\{ x \left| a_i^T x \begin{array}{ll} \leq b_i & i = 1, \ldots, s \\ = b_i & i = s + 1, \ldots, m \end{array} \right. \right\}.$$

Linear programming is a special case of convex programming. In this case the constraint region Ω is called a polyhedral convex set. Polyhedra have a very special geometric structure.

(c) Quadratic Programming:

The minimization or maximization of a quadratic objective functions over a convex polyhedron:

$$f_0(x) = \frac{1}{2}x^TQx + b^Tx + \alpha$$

Exercise: Show that $\nabla f_0(x) = \frac{1}{2}(Q+Q^T)x + b$ and $\nabla^2 f_0(x) = \frac{1}{2}(Q+Q^T)$.

Fact: f_0 is convex if and only if Q is positive semi-definite.

(d) Mini-Max:

$$f_0(x) = \max\{f_i(x) : \, i = 1, \dots, s\}.$$

(e) Nonlinear Programming

$$\Omega := \{x \in X : f_i(x) \le 0, i = 1, \dots, s, f_i(x) = 0, i = j+1, \dots m\}$$

(i) Differentiable: f_i is smooth $i = 0, \dots, m$
(ii) Nonsmooth: at least one f_i is not smooth
(iii) Semi-Infinite: $m = +\infty$.

(f) Box Constraints:

$$\begin{array}{rcl} \Omega & := & \{x \in \mathbb{R}^n : l_i \le x_i \le u_i, i = 1, \dots, n\} \\ l_i & \in & \mathbb{R} \cup \{-\infty\}, \quad u_i \in \mathbb{R} \cup \{+\infty\}, \quad l_i \le u_i \end{array}$$

This course is devoted to discrete optimization and so our focus is on the development of numerical methods for solving the general nonlinear programming problem under the assumption that all of the underlying functions are smooth and the variables are either integer or zero-one. After developing a few ideas associated with the general case, we refine our study to linear programming over the integers.

1.1. **A Sampling of Problems.**

1.1.1. *Train and Bus Scheduling.* Bus and train schedules repeat on a 24 hour basis. For each route the travel times between stops is known, and time spent at each stop must lie in a given time interval. It is preferred that two trains and/or busses traveling the same route be separated by a fixed time interval. To make connections between bus/train A and bus/train B at a given station, the arrival time of A must precede the departure time of B by a fixed amount. The problem is to find a feasible schedule for the system that *minimizes* the travel times between all stations in the system.

1.1.2. *Work Crew Scheduling.* Given a schedule of tasks and a list of employees with a range of expertise and wages, the problem is to design the weekly schedules of the work crews. Each day a crew must be assigned a duty period consisting of a set of one or more linked tasks satisfying numerous constraints on time to completion, qualifications and training of the crew, breaks and rest periods, ... The duty periods, weekly schedules or crew pairings must satisfy further constraints on task completion times and sequential work flow. The task is to minimize the total wage costs required while maintaining adequate work flow and contract specifications.

1.1.3. *Layout and Cutting Problems.* Whether placing circuits of a chip or cutting manufacturing patterns/templates from sheet metal, plastic, or fabric, the problem in each case is to follow precisely determined layout and cutting rules to satisfy demand and minimize waste.

1.1.4. *Pizza Delivery.* With multiple delivery vehicles, the problem is to determine the assignment of deliveries and the delivery routes to minimize the maximum customer wait-time.

1.2. **Some Classical Binary Integer LPs (BIP).**

1.2.1. *The Assignment Problem.* There are n people to carry out n jobs. Each person is assigned to carry out exactly one job. Some individuals are better suited to some jobs than others, so there is an estimated cost c_{ij} if person i is assigned to job j. The problem is to find a minimum cost assignment.

Let $x_{ij} = 1$ if person i is assigned to job j; otherwise, set $x_{ij} = 0$. Since each person can do only one job, we have

$$\sum_{j=1}^{n} x_{ij} = 1 \; i = 1, 2, \ldots, n \; .$$

Since each job must be assigned,

$$\sum_{i=1}^{n} x_{ij} = 1 \; j = 1, 2, \ldots, n \; .$$

The cost of an assignment is

$$\sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} x_{ij}.$$

Hence the problem can be stated as the BIP

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} x_{ij} \\ \text{subject to} & \sum_{j=1}^{n} x_{ij} = 1, \; i = 1, 2, \ldots, n \\ & \sum_{i=1}^{n} x_{ij} = 1 \; j = 1, 2, \ldots, n \\ & x_{ij} \in \{0, 1\}, \; i, j = 1, 2, \ldots, n \; . \end{array}$$

1.2.2. *The 0-1 Knapsack Problem.* There is a budget b available for investment in n projects during the coming year.Project j requires an investment of a_j to participate with an expected return of c_j dollars. The goal is to choose a set of projects to participate in so that the budget is not exceeded and the expected return is maximized.

Let $x_j = 1$ if project j is selected for participation; otherwise, $x_j = 0$. Since the budget cannot be exceeded, we require that

$$\sum_{j=1}^{n} a_j x_j \leq b \ ,$$

with an expected return of

$$\sum_{j=1}^{n} c_j x_j$$

Hence this problem can be stated as the BIP

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{n} c_j x_j \\ \text{subject to} & \sum_{j=1}^{n} a_j x_{ij} \leq b \\ & x_j \in \{0,1\}, \ j = 1,2,\ldots,n \ . \end{array}$$

1.2.3. *The Set Covering Problem.* Given a m regions, the problem is to decide where to place $n < m$ service facilities. For each region, the cost of installation and the other regions that a facility in this region can cover are known. For example, the facilities may be a fire stations (or cell towers, satellites, ice-cream stores, ...), and each station can service those regions for which a fire engine is guaranteed to arrive on the scene within 8 minutes. The goal is to choose a minimum cost set of service facilities so that all regions are covered.

Let $x_j = 1$ if a facility is placed in region j; otherwise, $x_j = 0$. Set $a_{ij} = 1$ if a facility in region j can service region i, and let c_j be the cost of placing a facility in region j. Then the set covering problem can be stated as the following BIP:

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{n} c_j x_j \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \geq 1, \ i = 1,2,\ldots,n \\ & x_j \in \{0,1\}, \ j = 1,2,\ldots,n \ . \end{array}$$

1.2.4. *The Traveling Salesperson Problem (TSP).* This is perhaps the most well-known BIP. A salesperson must visit each of n cities exactly once and then return to their starting point. The time taken to travel from city i to city j is c_{ij}. Find the order in which the salesperson should make their tour in the least amount of time.

Let $x_{ij} = 1$ if the salesperson travels directly from city i to city j; otherwise, $x_{ij} = 0$ and x_{ii} is not defined. Since a tour leaves each city only once

$$\sum_{j:j\neq i} x_{ij} = 1, \ i = 1,2,\ldots,n \ .$$

Since a tour arrives in each city only once

$$\sum_{i:i\neq j} x_{ij} = 1,\ \ j = 1, 2, \dots, n\ .$$

The constraints so far only require that each city is visited only once. But they do not guarantee that for each pair of cities there is a portion of the tour that takes the sales person from one of these cities to the other. To guarantee that this occurs, a tour must pass from every subset of cities to the remaining cities at least once. Let $\mathcal{S}$ denote all possible non-empty subsets of the integers $\{1, 2, \dots, n\}$. It is well-known that $\mathcal{S}$ contains $2^n - 1$ elements. To guarantee that for each pair of cities there is a portion of the tour that takes the sales person from one of these cities to the other, we must add the constraints

$$\sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 1,\ \ \forall\, S \in \mathcal{S}.$$

The TSP can now be stated as the BIP

$$\begin{array}{ll}
\text{minimize} & \sum_{i=1}^{n}\sum_{j:j\neq i} c_{ij}x_{ij} \\[2ex]
\text{subject to} & \sum_{j:j\neq i} x_{ij} = 1,\ \ i = 1, 2, \dots, n \\[2ex]
 & \sum_{i:i\neq j} x_{ij} = 1,\ \ j = 1, 2, \dots, n \\[2ex]
 & \sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 1,\ \ \forall\, S \in \mathcal{S} \\[2ex]
 & x_{ij} \in \{0, 1\},\ \ i, j = 1, 2, \dots, n,\ \ i \neq j\ .
\end{array}$$

1.3. **A Mixed Integer Program.** Mixed integer programs often arise in the context of modeling fixed costs. In a typical scenario a fixed charge cost function takes the form

$$h(x) = \begin{cases} f + c(x) & \text{if } 0 \leq x \leq C, \text{ and} \\ 0 & \text{otherwise,} \end{cases}$$

where the fixed charge $f > 0$ and the running cost is $c(x) \geq 0$ with $c(0) = 0$. To accommodate this jump discontinuity in the objective function h we introduce a new variable y satisfying

$$y = \begin{cases} 1 & \text{if } x > 0, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

We then replace $h(x)$ by

$$h(x, y) = yf + c(x),$$

and add the constraints

$$0 \leq x \leq Cy,\ \ y \in \{0, 1\}.$$

1.3.1. *The Uncapacitated Facility Location Problem (UFLP).* In this problem we are given a set of potential depots $N = \{1, 2, \ldots, n\}$ and a set $M = \{1, 2, \ldots, m\}$ of clients. We suppose that there is a fixed cost f_j associated with the use of the depot j, and a transportation cost c_{ij} if all of client i's order is delivered from depot j. The problem is to decide which depots to open, and which depot serves which client in order to minimize the total sum of the fixed costs and the transportation costs.

As described above, for each $j = 1, 2, \ldots, n$, we introduce a fixed cost (or depot opening cost) y_j, with $y_j = 1$ if depot j is used and $y_j = 0$, otherwise. Let x_{ij} be the fraction of client i demand met from depot j so that $\sum_{j=1}^{n} x_{ij} = 1,\ i = 1, 2, \ldots, m$. To represent the constraint that nothing is shipped from depot j if it is not opened we write

$$\sum_{i=1}^{m} x_{ij} \leq m y_j,\ y_j \in \{0, 1\},\ j = 1, 2, \ldots, n.$$

The problem can now be written as the mixed integer LP

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{j=1}^{n} f_j y_j \\ \text{subject to} \quad & \textstyle\sum_{j=1}^{n} x_{ij} = 1,\ i = 1, 2, \ldots, m \\ & \textstyle\sum_{i=1}^{m} x_{ij} \leq m y_j, \quad j = 1, 2, \ldots, n \\ & 0 \leq x_{ij},\ y_j \in \{0, 1\},\ i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n. \end{aligned}$$