Integer Programming and Combinatorial Optimization

Katta G. Murty Lecture slides

Integer Programming (IP) deals with LPs with additional constraints that some variables can only have values

- 0 or 1
- integer values
- or values in some specified discrete set

0−1 variables, also called *binary* or *Boolean variables* used to select one of two alternatives.

Example: Binary variables In automobile design, need to decide whether to use cast iron or aluminium engine block. Introduce a binary variable with definition:

$$y = \begin{cases} 0 & \text{if cast iron block used} \\ 1 & \text{if al. block used} \end{cases}$$

In this model need to restrict y to $0-1$ values only, because other values for y have no meaning. Such $0-1$ variables called *combinatorial choice variables.*

Example: Integer Variables: Army decides to use combat simulators to train soldiers. Each costs \$ 5 million US. Let

y = no. of combat simulators purchased by Army.

Then $y \geq 0$ is an integer variable.

Example: Discrete Variables: In designing water distribution system for a city, diameter of pipe to be used for a particular link needs to be decided. Pipe available only in diameters 16", 20", 24", 30". So, if

y = diameter of pipe used on this link

y can only take a value from set $\{16, 20, 24, 30\}$. A discrete valued variable.

Each discrete variable can be replaced by binary variables in the model.

Types of IP Models

If all variables required to take integer values only, model called a *Pure IP Model*. In addition, if they are all required to be 0 or 1, model called a *0−1 Pure IP Model*.

If some variables are required to be integer, and others can be continuous, model called *Mixed IP Model*, or *MIP*. If all integer decision variables are binary, model called *0−1 MIP*.

Integer Feasibility Problem refers to one with no obj. func. to optimize, but aim is to find an integer solution to a given system of linear constraints. In such model, if all variables binary, it is called *0−1 Feasibility Problem*.

Examples: Subset sum problem, Equal Partial sums problem.

Many puzzles from recreational math. can be posed as $0-1$ feasibility problems. Here is one, from Shakespeare's *Merchant of Venice*, which we solve by *Total Enumeration.*

The portrait is in this casket	The portrait is not in this casket	The portrait is not in the gold casket
1 = Gold	2 = Silver	3 = Lead

Figure 9.1

Combinatorial Optimization (CO) deals with finding best arrangement s. to specified constraints. Most CO models involve following components.

		Useful Models
Location	Where to put the plants?	p-median model, set covering model
Partition	Divide a set into subsets	Set partitioning, 0–1 IP, Assignment
Allocation	Allot jobs to machines	Assignment, 0–1 IP
Routing	Find optimal routes	TSP, Nonbipartite perfect matching
Sequencing	Find optimal order for jobs etc.	TSP, Permutation models
Scheduling	Arrange events over time	DP, Heuristics.

Formulation Examples

The One Dimensional Knapsack Problem: is a single constraint pure IP.

n types of objects are available. For $i = 1$ to n, ith type has weight w_i kg and value v_i \$.

Knapsack has weight capacity of w kg .

Objects cannot be broken. Only a nonnegative integer no. of them can be loaded into knapsack.

Determine which subset of objects (and how many of each) to load into knapsack to maximize total value loaded subject to its weight capacity.

Two versions; *nonnegative integer knapsack problem, 0–1 knapsack problem*.

Simplest pure IP. Many applications. Appears as a subproblem in algorithms for cutting stock problem.

Example: $n = 9$. $w = 35$ kg.

Type	Weight	Value
1	3	21
2	4	24
3	3	12
4	21	168
5	15	135
6	13	26
7	16	192
8	20	200
9	40	800

Application: *Journal Subscription Problem:* Project carried out at UM-COE library in 1970's. For sample problem, subscription budget is \$650.

Journal	Subscription	Readership
1	80	7840
2	95	6175
3	115	8510
4	165	15015
5	125	7375
6	78	1794
7	69	897
8	99	8316

Capital Budgeting Problem: Available budget is \$ 23 million.

Project	Cost (\$mil.)	Annual return (\$$10^4$units)
1	3	12
2	4	12
3	3	9
4	3	15
5	15	90
6	13	26
7	16	112

Multidimensional Knapsack Problem: You get this if no. of constraints is > 1

Multiperiod Capital Budgeting Problem: Determine which subset of projects to invest in to maximize total expected amount obtained when projects sold at end of 4th year. Money unit = US $10,000.

Project	Investment needed in year			Expected selling price in 4th year
	1	2	3	
1	20	30	10	70
2	40	20	0	75
3	50	30	10	110
4	25	25	35	105
5	15	25	30	85
6	7	22	23	65
7	23	23	23	82
8	13	28	15	70
Funds available to invest	95	70	65	

$0 - 1$ Knapsack with Multiple choice constraints

Consider $0 - 1$ multidimensional knapsack in n articles. $\forall j = 1$ to n: $x_j = 1$ if article j selected, 0 otherwise.

Articles partitioned into p disjoint sets: $\mathbf{A}_1 = \{1, \ldots, n_1\}$, $\ldots$, $\mathbf{A}_p = \{n_{p-1} + 1, \ldots, n\}$.

Can select exactly 1 article from each $\mathbf{A}_r$. System of constraints representing this type constraint in $0-1$ variables called **System of Multiple choice constraints**.

Example: In previous example, projects 1, 2 (fertilizer mfg.); projects 3, 4 (tractor leasing); projects 5, 6, 7, 8 (miscellaneous). Select exactly 1 fertilizer mfg., exactly 1 tractor leasing; & at least one miscellaneous projects.

Set Partitioning, Set Covering, and Set Packing Problems

Let $A_{m \times n}$ be a 0–1 matrix, $e = (1, \dots, 1)^T$ a column vector of all 1's in R^n; and c a general integer cost vector.

These 3 models are very important 0–1 pure IPs with many applications. They are:

Set Covering Problem: $\min z = cx$ subject to $Ax \geq e$, and x is 0–1.

Set Partitioning Problem: $\min z = cx$ subject to $Ax = e$, and x is 0–1.

Set Packing Problem: $\min z = cx$ subject to $Ax \leq e$, and x is 0–1.

Example: US Senate Simplified Problem: Select smallest size committee in which senators 1 to 10 are eligible to be included, subject to constraint that each of following groups must have at least one member on committee.

Group	Eligible senators in this group
Southerners	{1, 2, 3, 4, 5}
Northerners	{6, 7, 8, 9, 10}
Liberals	{2, 3, 8, 9, 10}
Conservatives	{1, 5, 6, 7}
Democrats	{3, 4, 5, 6, 7, 9}
Republicans	{1, 2, 8, 10}

Facility Location Problem: Area divided into 8 zones. Average Driving time (minutes) between zones given below. Blank entries indicate that time is too high. Need to set up facilities (like fire stations, etc.) in a subset of zones. Constraint: every zone must be within critical time (25 minutes) of a zone with a facility. Find best locations for smallest no. of facilities.

	Average driving time							
	to $j = 1$	2	3	4	5	6	7	8
from $i = 1$	10		25		40			30
2		8	60	35		60	20	
3	30		5	15	30	60	20	
4	25		30	15	30	60	25	
5	40		60	35	10		32	23
6		50	40	70		20		25
7	60	20		20	35		14	24
8	30		25		25	30	25	9

Fire Hydrant Location Problem: Street network with traffic centers 1 to 6, and street segments (1, 2), (1, 5), (1, 7), (2, 3), (2, 5), (3, 4), (4, 5), (4, 6), (6, 7). Find locations for smallest no. of fire hydrants so that there is one on every street segment. **Node covering problem**.

Delivery problems:

A Delivery Problem Depot, 0, to make deliveries to locations 1 to 8. 9 good routes for delivery vehicles given. Cost is expected driving time (hours). Determine which of these 9 routes should be implemented to minimize total driving time of all vehicles.

Route no.	Route	Cost
R_1	0-3-8-0	6
R_2	0-1-3-7-0	8
R_3	0-2-4-1-5-0	9
R_4	0-4-6-8-0	10
R_5	0-5-7-6-0	7
R_6	0-8-2-7-0	11
R_7	0-1-8-6-0	8
R_8	0-8-4-2-0	7
R_9	0-3-5-0	7

Airline Crew Scheduling: Very important large scale application for set covering. Basic elements are **flight legs**, a flight between two cities, departing at one city at a specified time and landing next at the second city at a specified time, in an airline's timetable. **Duty period** for a crew is a continuous block of time during which the crew is on duty, consisting of a sequence of flight legs each one following the other in chronological order. **Pairing** for a crew is a sequence of duty periods that begins and ends at the same domicile.

Forming Sales Districts: Region consists of sales areas 1, ..., m. These may be towns, villages, etc. To group them into **Sales districts**, each to be managed by one director.

Generate various subsets of sales areas, LIST = $\{S_1, \ldots, S_n\}$ each forming a good district. Use some heuristic, or manually using a screen display. Select a subset of LIST s. th. it forms a partition of set of areas.

Example: Consider following LIST generated:

Subset	Cost	Subset	Cost
$S_1 = \{1, 2, 3, 4, 5\}$	39	$S_8 = \{3, 5, 9, 11\}$	48
$S_2 = \{2, 7, 8, 9, 10, 11\}$	83	$S_9 = \{4, 7, 11\}$	32
$S_3 = \{3, 4, 6, 7, 10\}$	64	$S_{10} = \{1, 2, 3, 7\}$	39
$S_4 = \{1, 2, 5, 9, 10, 11\}$	85	$S_{11} = \{3, 4, 10, 11\}$	56
$S_5 = \{2, 4, 5, 6, 10\}$	93	$S_{12} = \{2, 4, 8, 11\}$	62
$S_6 = \{3, 4, 7, 9\}$	50	$S_{13} = \{3, 5, 7, 10\}$	84
$S_7 = \{1, 5, 7, 10\}$	77		

Meeting Scheduling Problem: $A = (a_{ij})$ an $k \times n$ $0-1$ matrix, where:

n	Meetings to be scheduled
k	administrators, each to attend a subset of meetings
T	Hour time slots available
a_{ij}	1 if administrator j has to attend meeting i; 0 otherwise

Problem 1: Find maximum no. meetings that can be scheduled without conflicts. A **Set Packing problem**.

: Find min no. time slots needed to schedule all meetings without conflicts. A **Graph coloring problem**.

Example: 11 administrators, 13 meetings. A given below. Blank entries are 0.

	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	1							
2	1				1						
3				1		1	1				
4		1	1	1							
5		1			1			1			
6							1		1	1	
7				1			1				1
8			1			1		1	1		
9										1	1
10									1	1	
11	1				1			1		1	
12	1				1					1	
13	1	1					1				

Uncapacitated Plant Location

n sites in a region need a product. For $i = 1$ to n:

$d_i =$	demand at site i over plannig horizon (could be life of the plants), in units.
$m =$	max. no. plants that can be set up.
$f_i + s_i\alpha =$	cost of setting up a plant of prod. capacity α (over planning horizon) at site i.
$c_{ij} =$	unit cost of shipping product from site i to site j.

In practice $m << n$. Assumplion implies that at any site, plant of any prod. capacity.

Capacitated Plant Location

Previous model made unrealistic assumption that at any site, any size plant can be set up. Assume:

$k_i =$ Prod. capacity (over planning horizon) of a plant set up at site i.

$f_i =$ Cost of setting up a plant (of above capacity) at site i.

Other data same as in uncapacitated problem.

Disjunctive Constraints

Disjunctive refers to system of 2 constraints of which ≥ 1 must hold (or in general, $\geq k$ constraints out of a given set of m must hold).

To model using $0-1$ variables, we need an upper bound , $\alpha > 0$ say, for all constraint functions in the feasible region.

Batch Size Variables: In a LP model suppose we have a variable x_j which is required to be:

$$\text{either } 0 \text{ (i.e., } x_j = 0) \\ \text{or } \geq \ell_j$$

where ℓ_j is some specified positive lower bound if x_j is positive.

i.e., here we want one of the two constraints $x_j = 0 \quad x_j \geq \ell_j$ to hold. This situation arises if x_j represents material purchased from a supplier who will supply only in lot sizes $\geq \ell_j$.

Can be formulated using one combinatorial choice variable.

Example Either $x_1 = 0$, or $x_1 \geq 10$

Either $x_2 = 0$, or $x_2 \geq 15$

Suppose $\alpha = 100$ is an upper bound for both x_1, x_2 in feasible region.

Example: 1 out of 2 disjunctive constraints:

Feasible region defined by: $x_1, x_2 \geq 0$, $x_1, x_2 \leq 10$; and either $x_1 \leq 5$ or $x_2 \leq 5$.

Use $\alpha = 10$ as an upper bound for both x_1, x_2 in feasible region; and represent it by a mixed $0 - 1$ system.

Representing Union of Two Convex Polytopes

$P_t = \{x : A^t x \leq b^t, x \geq 0\}$; $t = 1, 2$.

Let $\beta > 0$ be such that $b^1 + \beta(1, 1, ...)^T$ $[b^2 + \beta(1, 1, ...)^T]$ is an upper bound for $A^1 x$ $[A^2 x]$ over P_2 $[P_1]$.

Assignment Problem: n machines, m jobs, where $n \geq m$. c_{ij} = cost of doing job j on machine i.

Each machine can do at most one job.

Each job must be carried out on exactly one machine.

Assign jobs to machines to minimize cost of completing all jobs.

By Integer Property of Transportation problems, this problem can be solved as an LP, because optimum solution of LP relaxation obtained by Simplex method will be integral; since every extreme point of LP relaxation is integral by **Total Unimodularity (TU)** of coeff. matrix.

TU and U properties

Let $A_{m \times n} = (a_{ij})$ be of rank r. A is said to be **TU** iff the determinant of every square submatrix of A is $\in \{0, +1, -1\}$.

A is said to be **U** iff the determinant of every square submatrix of A of order r is $\in \{0, +1, -1\}$.

Theorem: Coeff. matrix in min cost pure network flow problems are TU; and all BFS in these problems are integral if the

RHS constants vector is integral.

Theorem: (Heller and Tompkins) $A = (a_{ij})$ with all $a_{ij} \in \{0, +1, -1\}$, and every col. of A has at most 2 nonzero entries. A is TU iff rows of A can be partitioned into two sets s. th.

(a) if the two nonzero entries in a col. have same [different] signs, their rows are in different [same] sets.

Theorem: (Hoffman and Kruskal) Consider $Ax \leq b, \quad x \geq 0$ where A integral. Following are equivalent. (a) A is TU (b) for all b integral, all BFSs of above system are integral (c) every nonsingular square submatrix of A has integer inverse.

Theorem: Consider $Ax = b, \quad x \geq 0$ where A integral. Following are equivalent. (a) A is U (b) for all b integral, all BFSs of above system are integral (c) every basis for above system has integer inverse.

Matching and Edge Covering Problems in Undirected Networks

Let $G = (\mathcal{N}, \mathcal{A}, c)$ be an undirected network with c as vector of edge weights, or costs. $E \subset \mathcal{A}$ is said to be a

matching	if E contains ≤ 1 edge incident at each node
perfect matching	if E contains exactly one edge incident at each node
edge cover	if E contains ≥ 1 edge incident at each node.

Also, let $(\mathcal{N}^{\leq} \cup \mathcal{N}^{=} \cup \mathcal{N}^{\geq} \cup \mathcal{N}^{0})$ be a partition of $\mathcal{N}$. THen E is said to be a **1-M/EC (1-Matching/Edge Covering)** WRT this partition of $\mathcal{N}$ if it has

≤ 1 edge incident at all nodes	$i \in \mathcal{N}^{\leq}$
exactly 1 edge incident at all nodes	$i \in \mathcal{N}^{=}$
≥ 1 edge incident at all nodes	$i \in \mathcal{N}^{\geq}$

Finding min cost matchings, perfect matchings, edge covers, 1-M/ECs all have efficient algorithms.

Since node-edge incidence matrices of bipartite networks are TU, all these problems can be solved efficiently by LP techniques in bipartite networks.

In nonbipartite networks, all these problems are nontrivial IPs, but have efficient **blossom algorithms** pioneered by J. Edmonds. These algorithms initiated the subject of **Polyhedral Combinatorics** of IPs.

Assignment Problem interpreted as a bipartite matching problem

Let cost matrix be:

$$\begin{pmatrix} 3 & 7 & 9 \\ 11 & 5 & 14 \\ 6 & 8 & 10 \end{pmatrix}$$

A bipartite min cost perfect matching problem.

The Chinese Postman Problem (CPP)

Street network of a postman's beat is an undirected network. Find a min distance route starting at post-office-node, going thro' each edge $\geq$ once, & returning to post office at end.

Posed by Guan Mei Go (1962). Can be posed as a min cost perfect matching problem in a nonbipartite network.

Approach for the CPP

1. Identify $J =$ Set of odd nodes in network. If $J = \emptyset$, network has **Euler route** which is optiman postman route.

2. If $J \neq \emptyset$, find $\mathcal{P}_{ij}$ = a shortest path between i and j for each $i \neq j \in J$, and let d_{ij} be its length. Let $H = (J, \{(i;j) : i \neq j \in J\}, (d_{ij}))$. Find a min cost perfect matching M in H.

3. In original network duplicate all edges on $\mathcal{P}_{ij}$ $\forall (i;j) \in M$. Opt. postman route is Euler route in resulting network.

Min cost perfect matching problem

$G = (\mathcal{N}, \mathcal{A}, c)$. Define $x_{ij} = 1$ if $(i; j)$ included as matching edge; 0 otherwiase.

$$\begin{aligned} \min \quad \sum c_{ij} x_{ij} & \\ \sum (x_{ij} : \text{ over } j \text{ s. th. } (i;j) \in \mathcal{A}) &= 1 \quad \forall i \in \mathcal{N} \\ x_{ij} &= 0 \text{ or } 1 \quad \forall (i;j) \in \mathcal{A} \end{aligned}$$

Example:

The LP relaxation is below. Its opt. BFS is $\bar{x} = (1/2, 1/2, 1/2)^T$, nonintegral. IP infeasible.:

$$\begin{array}{llllr} \min & -x_{12} & -x_{23} & -x_{31} & \\ \text{s. to} & x_{12} & +x_{23} & & = 1 \\ & & x_{23} & x_{31} & = 1 \\ & x_{12} & & x_{31} & = 1 \\ & x_{12} & x_{23} & x_{31} & \geq 0 \end{array}$$

Strategy Developed by J. Edmonds TO SOLVE PROBLEM:

Add additional constraints to LP rexation to remove all non-integral extreme points, while at same time not creating any new extreme points, or deleting integer extreme points.

Resulting problem can be solved by LP approaches to yield integer optimum. First nontrivial IP for which an efficient algo. developed based on this approach. Now called **Polyhedral approach to IP**.

Subnetwork formed by an odd subset of nodes Y_σ. Any perfect matching can have at most $(|Y_\sigma| - 1)/2$ edges from this subnetwork. So, every perfect matching satisfies:

$$\sum(x_{ij}: \quad \text{over } i, j \in Y_\sigma \text{ and } (i, ; j) \in \mathcal{A}) \leq (|Y_\sigma| - 1)/2$$

called **Valid inequality** for IP. This one known as **blossom**

ineq. corresponding to odd subset Y_σ. Edmonds showed that by adding blossom ineq. of all odd subsets, we get linear constraint representation of convex hull of all perfect matching incidence vectors.

Fixed charge problems

If cost of performing Activity j at level x_j is: $\quad 0 \quad$ if $x_j = 0$; $f_j + c_j x_j \quad$ if $x_j > 0$ we have a **Fixed charge problem**.

$f_j =$ setup cost, or fixed charge incurred to make $x_j > 0$.

$c_j =$ variable cost of activity j. Per unit cost of increasing x_j from 0, once fixed charge is paid.

Common in transportation, mfg. applications. Model of form: $\min \Sigma_{j=1}^{n} c_j x_j + \Sigma_{j \text{ s. th.} x_j > 0} f_j \quad$ s. to $\quad Ax = b, \ \ x \geq 0$.

Define: $\quad y_j = 1 \quad$ if $x_j > 0; \quad = 0$ if $x_j = 0$.

And let $\alpha > 0$ be a practical upper bound for all x_j. Can be modeled as an MIP using these.

The Traveling Salesman Problem (TSP) :

A salesperson's trip begins and ends in city 1, and must visit each of cities $2, \ldots, n$ exactly once in some order.

$c = (c_{ij})$, the $n \times n$ cost matrix for traveling between pairs of cities, is given.

If the cities visited in order are: $1, p_2, \ldots, p_n; 1$ this is called a Tour or Hamiltonian cycle or Node covering cycle, and its cost is: $c_{1,p_2} + c_{p_2,p_3} + \ldots + c_{p_{n-1},p_n} + c_{p_n,1}$.

Find a minimum cost tour.

No. of tours in an n city problem is $(n-1)!$.

Let $\mathcal{N} = \{1, \ldots, n\}$, set of all cities in problem. Let $\mathcal{N}_1 \subset \mathcal{N}$, $\mathcal{N}_1 \neq \mathcal{N}$. A tour covering cities in $\mathcal{N}_1$ only, without touching any city in $\mathcal{N} \backslash \mathcal{N}_1$ is called a **subtour** spanning the subset of cities $\mathcal{N}_1$.

Define: $x_{ij} = 1$ if salesman goes from i to j, 0 otherwise. Then $x = (x_{ij})$ is a $0-1$ matrix satisfying the constraints in the assignment problem; & also $x_{ii} = 0 \forall i$. So, every tour is an assignment with 0-diagonal.

An assignment is called **tour assignment** if it has 0-diagonal & represents a tour. Make $c_{ii} = \infty\ \forall i$. So, TSP is:

$$\min \quad z(x) = \sum \sum c_{ij} x_{ij}$$

$$\begin{aligned} \text{s. to} \quad \sum_j x_{ij} &= 1, \forall i \\ \sum_i x_{ij} &= 1, \forall j \\ x_{ij} &\in \{0,1\} \ \ \forall i,j \\ \&\ x \text{ is a} \quad & \text{tour assignment} \end{aligned}$$

Tucker's MIP formulation of TSP

Uses new continuous variables $u = (u_2, \ldots, u_n)^T$.

$$\begin{array}{rrcl} \min & z(x) = \sum\sum c_{ij}x_{ij} & & \\ & \text{s. to} \quad \sum_j x_{ij} & = & 1, \forall i \\ & \sum_i x_{ij} & = & 1, \forall j \\ & u_i - u_j + nx_{ij} & \leq & n-1 \quad \forall i \neq j \in \{2, \ldots, n\} \\ & x_{ij} & \in & \{0,1\} \;\; \forall i, j \end{array}$$

The constraints make it impossible to have a subtour not containing node 1.

Other ways of imposing subtour elimination using the $0-1$ x_{ij} variables only are:

Either $\Sigma(x_{ij} : i \in S, j \notin S) \geq 1 \quad \forall S \subset \{1, \ldots, n\}$ with $2 \leq |S| \leq n-2$

or $\Sigma(x_{ij} : i \in S, j \in S) \leq |S| - 1 \quad \forall S \subset \{1, \ldots, n\}$ with $2 \leq |S| \leq n-2$

Production, Lot-Sizing Problem

To min sum of set-up, production, storage costs to meet known demands in T periods. For $t = 1$ to T:

f_t = set-up cost to produce in t

c_t = unit production cost in t

d_t = demand (to meet) in t

s_t = unit storage cost from t to $t+1$.

Formulation 1: Let $\omega = \Sigma_{t=1}^{T} d_t$, an upper bound for production in any period. Define decision variables:

x_t = production in t

I_t = units stored from t to $t+1$

$y_t = 1 \quad \text{if } x_t > 0; \quad 0 \quad \text{otherwise.}$

$$
\begin{aligned}
\min \sum_{t=1}^{T} \{f_t y_t + c_t x_t + s_t I_t\} & \\
\text{s. to} \quad x_1 &= d_1 + I_1 \\
I_{t-1} + x_t &= d_t + I_t, \quad 2 \le t \le T \\
x_t &\le \omega y_t, \quad 1 \le t \le T \\
I_t &= 0 \\
I_t, x_t &\ge 0, \quad \forall t \\
y_t &\in \{0,1\} \quad \forall t
\end{aligned}
$$

Formulation 2: Define decision variables:

q_{it} = Units produced in i to meet demand in $t \ge i$

$$
\begin{aligned}
\min \sum_{t=1}^{T} f_t y_t \;\; &+ \;\; \sum_{t=1}^{T} \sum_{i=1}^{t} (c_i + s_i + s_{i+1} + \ldots + s_{t-1}) q_{it} \\
\text{s. to} \quad \sum_{i=1}^{t} q_{it} &= d_t \;\; 1 \le t \le T \\
q_{it} &\le d_t y_t \;\; 1 \le t \le T, \;\; 1 \le i \le t \\
q_{it} &\ge 0 \;\; 1 \le i \le T, \;\; i \le t \le T \\
y_t &\in \{0,1\} \;\; \forall t
\end{aligned}
$$

Which formulation is better?

Importance of good formulation

To solve large IP or CO good formulation very critical. In min problems, algorithms use a **Lower bounding strategy** which computes a LB (lower bound) for min obj. value. A formulation that leads to highest LB is the best.

A Combinatorial optimization problem: The line haul problem in trucking

A trucking company that operates in the midwest region of USA has 15 terminals. During the day each terminal accepts packages for shipment until 6 PM. At 6 PM the office in the terminal closes and no more packages are accepted until next business day. During the night, each package is shipped to the terminal which is closest to its destination, and should arrive there before 6 AM next morning. From that terminal, the package is delivered to its destination address during next day by a separate division of the company. The line haul problem is concerned with the overnight transfer of each package from its origin-terminal to its destination-terminal, in the most efficient way possible. Line haul is carried out by trucks all of which can be assumed to be of the same size.

Each truck can travel upto 540 miles during a night, and can make any no. of intermediate stops for loading-unloading. The loading-unloading time can be ignored for solving the line haul problem. To keep problem simple, assume that each terminal has an unlimited supply of trucks to use. A route for a truck may consist of several cycles (i.e., a roundtrip from its home-terminal to a subset of terminals, returning back to its home terminal at end) and/or a chain to another terminal with break of journey and stay there next day; and the length of this route has to be ≤ 540 miles.

Following is data for one night. The first table gives the amount of material (in truck load units) to ship that night between every pair of terminals. The 2nd table gives the accepted driving distance between pairs of terminals.

Truckloads to ship between terminals

To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
from 1	0	1	1	2	4	1	1	3	1.2	1	.5	1	1	1	1
2	1.4	0	2.5	.8	2.25	.12	.37	.67	.04	.08	.04	.12	.33	.5	.25
3	.62	1	0	.6	2	.2	1.75	1.6	.6	1	.25	1.25	.66	.75	.75
4	1.33	1.5	1.5	0	.58	.16	.54	.5	.12	0	.43	0	.43	.92	1.4
5	2.5	2	2	1.33	0	2	5	2	4	2	1.5	6	3	2	1
6	.12	.12	.04	.08	1	0	.04	.87	0	.04	.08	.66	.54	.5	.04
7	.75	0	.25	.04	4	.85	0	2	.15	.12	1.9	.5	1.6	.5	.2
8	5	.78	.92	.24	1.4	.52	.4	0	.92	.2	.16	1.96	1.82	.86	.16
9	.4	.1	0	.88	2	.35	1	.25	0	.72	.43	.43	.62	.58	.04
10	1.1	.1	1	.12	1.5	0	.6	.3	.2	0	.7	.3	.5	.25	.12
11	.4	.1	0	.4	3.5	0	.5	0	.25	.25	0	1.75	.08	1.16	.04
12	1	.25	1.25	.25	3	.5	.58	.25	.08	.08	.5	0	.75	1.25	.33
13	1.42	0.37	.33	.37	1.54	.29	1.08	1.79	0	.33	.54	.92	0	.25	.04
14	1.2	.25	.25	.12	1.92	.33	1.25	.79	.12	.46	.58	1.04	.33	0	0
15	.16	.58	.2	.75	.54	.16	0	.08	.16	0	.29	.12	.16	.29	0

Distance (miles) between terminals

To	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
From 1	0	302	360	329	277	167	185	190	223	156	298	311	97	263	387
2		0	232	111	252	187	347	118	261	293	405	362	248	205	156
3			0	127	165	214	300	304	193	268	391	253	263	121	124
4				0	196	167	328	178	228	266	491	276	246	148	67
5					0	158	154	285	63	121	238	110	195	48	247
6						0	168	133	108	116	276	215	93	114	234
7							0	261	108	54	121	126	109	192	391
8								0	237	245	382	345	154	241	243
9									0	67	200	123	141	73	273
10										0	172	156	74	150	330
11											0	128	224	270	470
12												0	230	142	326
13													0	168	317
14														0	200
15															0

Assuming that a truck can be loaded with material to be shipped to any number of terminals; and that each load can be split between any number of trucks, find routes for the trucks to complete the shipments that minimizes the total truck mileage.