Integer Programming and Combinatorial Optimization

Katta G. Murty Lecture slides

Integer Programming deals with linear programs 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 *Bolean variables* used whenever we have 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 $\{16, 20, 24, 30\}$. This is a discrete valued variable.

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

Types of IP Models

If all variables in model are 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 a problem in which there is no objective functions to be optimized, but aim is to find an integer solution to a given system of linear constraints. In such a model, if all variables binary, it is called *0−1 Feasibility Problem*.

Examples: Subset sum problem with data $\{d_1 \text{ to } d_{10}\} = \{317, 89, 463, 572, 59, 311, 484, 786, 898, 944\}$, $d_0 = 2206$.

Equal Partial sums problem with data a $\{26, 97, 84, 30, 78, 112, 9, 65, 54\}$, $b = \{39, 7, 8, 58, 27, 46, 73\}$.

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 10.1

Combinatorial Optimization deals with the problem of finding the best arrangement subject to specified constraints. Most combinatorial optimization 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

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	

Set Partitioning, Set Covering, and Set Packing Problems

Let $A_{m\times n}$ be a 0–1 matrix, $e = (1, \ldots, 1)^T$ 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.

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.

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, 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.

Differences Between LP and IP Models:

LP	IP
1. Theoretically proven nec. and suff. optimality conditions exist. Useful to check whether a given feasible solution optimal	No known opt. conds. to check whether a given feasible sol. is opt., other than to compare it with every other feasible solution implicitly or explicitly.
2. Algos. are algebriac methods based on opt. conds.	All existing methods are enumerative methods based on partial enumeration.
3. Excellent software packages available. Very large models can be solved within reasonable times using them.	Performance of algorithms is very highly dependent on problem data. For most models, only moderate sized problems can be solved within reasonable times.

The Branch and Bound Approach:

Assume original problem minimization problem. Let K_0 = its set of feasible solutions.

During B&B K_0 is partitioned into many simpler subsets, each subset is set of feasible sols. of a problem called a Candidate Problem or CP.

Each CP is the original problem, augmented with additional constraints called Branching Constraints.

Branching constraints are simple constraints generated by an operation called Branching.

Whenever a new CP is generated, an

LB = Lower Bound for min. obj. value in it

is computed by a procedure called Lower bounding strategy.

For some CPs, the LB strategy may actually produce a minimum cost feasible sol. in it. In this case, that CP is said to be Fathomed, it need not be processed any further, so is taken out from further consideration.

Among the optimum solutions of fathomed CPs, the best is called the incumbent at this stage, and it is stored and updated. So, the objective value of incumbent is an Upper Bound for the min obj. value in original problem.

The incumbent and upper bound change whenever a new and better feasible sol. appears in method due to fathoming.

In each stage, method selects one CP to examine, called Current CP.

- If LB for current CP $\geq$ current Upper Bound, this CP is Pruned, i.e., discarded. The Partial enumeration property of method comes from this.
- Otherwise, set of feasible solutions of this CP is partitioned into 2 or more subsets by applying branching strategy on it.

Main Steps in B&B

Bounding: B&B uses both :

Upper Bound for min objective value in original problem: Changes whenever incumbent does, and decreases when it changes.

Lower Bound for min obj. value in each CP: Calculated by applying LB strategy on it.

Pruning: Deleting some CPs from further consideration. A CP is pruned

- if its LB $\geq$ Current UB
- if it is fathomed
- if it is found infeasible

Branching: This operation on a CP (Called Parent Node, generates two or more new CPs (called its Children).

The various steps:

The LB strategy: Most commonly used LB strategy is based on solving a relaxed problem.

To find LB for a CP, this strategy relaxes (i.e., ignores) difficult constraints in it until remaining problem can be solved by an efficient algo. Opt. sol. of relaxed problem called Relaxed Optimum. Objective value of relaxed opt. is a LB for the CP.

Fathoming Criterion: If relaxed opt. satisfies the relaxed constraints,it is in fact an opt. sol. for that CP.

Examples: TSP

0–1 Knapsack

Pure 0–1 IP

MIP.

The Braching Strategy:

Usually carried out by selecting a **Branching Variable**, one that is likely to make LBs for children as high as possible.

If branching variable is a 0–1 variable x_1, branching constraints are:

If branching variable is an integer variable x_1 whose value in present relaxed optimum is the nonintegral $\bar{x}_1$, branching constraints are:

1 Union of sets of feasible solutions of child problems is always the set of feasible solutions of parent.

2 Every child always inherits all branching constraints in its parent. So always, LB for child $\geq$ LB for parent.

The Search Strategy:

List refers to the set of all unexplored CPs in the present stage, i.e., set of all Live nodes, those not yet branched, fathomed or pruned.

One strategy picks current CP to branch to be the one in list with least lower bound.

Another is a backtrack search strategy based on depth first search.

Search terminates when list becomes $\emptyset$. Incumbent then is an optimum solution.

B&B for MIP Based on LP Relaxation:

Example: Consider following MIP.

y_1	y_2	x_1	x_2	x_3	x_4	$-z$	b
1	0	0	1	-2	1	0	3/2
0	1	0	2	1	-1	0	5/2
0	0	1	-1	1	1	0	4
0	0	0	3	4	5	1	-20

$y_1, y_2 \geq 0$, and integer; x_1 to $x_4 \geq 0$; z to be minimized

B & B for 0−1 knapsack problem

Object	Wt. w_j	Value v_j	Density v_j/w_j
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	

Wt. capacity = 35

The formulation of the problem is:

Simplified method to solve LP relaxation of 0−1 knapsack

Load knapsack with available objects in decreasing order of density. In end, if a full object won't fit, load it at fractional value that will fit.

HWs. 1. Given set of integers {80, 66, 23, 17, 19, 9, 21, 32}, need to find a subset of them s. th. their sum is as close to 142 as possible, without exceeding it. Formulate.

2. Object of weight $w = 3437$, a balance, and multiple copies of stones with weights 1, 5, 15, 25, 57, 117, are available.

Put object in right pan of balance. Determine how many stones of each wt. to put in left and/or right pans to balance, using smallest no. of stones. Formulate.

3. Company considering opening plants to make a product. 4 sites (S_1 to S_4) available, with following data. Demand for product in markets M_1, M_2, M_3 has to be met. In following table:

[1] Fixed cost is cost that must be paid to keep plant at site open/day.

[2] Capacity is the production capacity (tons/day) of plant at site, if it is kept open on that day.

[3] Cost/ton including production and shipping costs, from site to the market.

Site	Fixed cost[1]	Capacity[2]	Cost/unit to ship to[3]		
			M_1	M_2	M_3
S_1	\$400	120	\$25	37	48
S_2	600	80	38	15	29
S_3	350	130	32	37	21
S_4	500	110	20	42	38
Daily demand			80	70	40

At most two plants can be left open daily. Plant at S_1 can be left open only if plant at S_2 is also opened. Plants at either S_2 or S_3 or both must be left open daily. Formulate to decide which plants to open, and the shipping pattern, to minimize total cost. Do not solve numerically.

4. Letter A is worth 1 point, B is worth 2 points, etc. Consider following *words* (these words may have no meaning in English): DBA, DEG, CFG, AID, FFD, IGB, AGC, BDF, EAE.

You need to select exactly 4 words among these to: maximize sum of their third letter values, subject to constraint that sum of their 1st letter values is $\geq$ sum of their second letter values + 5. Formulate, do not solve numerically.

5. 5 projects being considered. Table gives data on AR = expected annual return, FI = investment needed in first year, WC = working capital expenses, and SE = expected safety and accident expenses, on each project in money units.

Project	AR	FI	WC	SE
1	49.3	150	105	1.09
2	39.5	120	83	1.64
3	52.6	90	92	0.95
4	35.7	20	47	0.37
5	38.2	80	54	0.44
Constraint on total	≥ 100	≤ 250	≤ 300	≤ 3.8

To determine which projects to approve to max expected annual return from approved projects, s. to constraints. Formulate.

6. Solve MIP by B & B: max $4y_1 + 5x_1 + x_2$ subject to
$3y_1 + 2x_1 \leq 10$

$y_1 + 4x_1 \leq 11$
$3y_1 + 3x_1 + x_2 \leq 13$
$y_1, x_1, x_2 \geq 0$, x_1, x_2 integer.

7. Solve MIP by B & B: max $4y_1 + 3x_1 + x_2$ s. to
$3y_1 + 2x_1 + x_2 \leq 7$
$2y_1 + x_1 + 2x_2 \leq 11$
$y_1, x_1, x_2 \geq 0$, x_1, x_2 integer.

8. Solve 0−1 knapsack problem with following data using B & B: Knapsacks weight capacity = 15.

Weight	6	8	5	4
Value	17	23	13	9

9. Solve 0−1 knapsack problem with following data using B & B: Knapsacks weight capacity = 40.

Weight	19	15	20	8	5	7	3	2	4
Value	380	225	320	96	70	126	30	22	68