Integer Programming Formulations

Author | Ishank Juneja

May 20, 2018

1 Overview

An IP problem is (usually, may be non linear) an LPP with the further restriction that variables are Integers.
IPPs are of types,
All Integers (0, 1, 2...), zero-one/Binary, Mixed discrete and continuous.

2 Methods to Solve LPPs

- **Rounding Off** : A common Assumption (May be incorrect): The feasible Integer point closest to the optimal point will be optimal.
  Requires the idea of Rounding off. Rounding off is bad since, if there are n variables, then each variable will have an upper and lower integer bound and the possible solution set will be $\mathcal{O}(2^n)$. Also some of the other problems are:
  1. The best feasible solution among the rounded set may not be optimal.
  2. It's possible that all rounded solutions are infeasible but an optimal point exists.
- **Exploring Integer points in Feasible Region** : We want to use ideas from LP but an issue is that the region is not Convex unlike LPP.

3 Grouping Problem with P-median formulation

i = 1, 2, ..., n, $X_{jj} = 1$ iff point j is a median and $X_{ij} = 1$ iff i is grouped to median j.
Minimize $\sum\sum d_{ij}X_{ij}$, where d_{ij} represents an element of the distance Mat. D
Subject to the constraints,

1. $\sum_{j=1}^{n} X_{jj} = p$ where p is given no. of groups
2. Any point that is not a median can be clustered only along an existing median, hence $X_{ij} \leq X_{jj}$
3. Only one group per node allowed, $\sum_{j=1}^{n} X_{ij} = 1$
4. $X_{ij} \quad \in \quad \{0, 1\}$

4 Fixed Charge/Expenditure Problem

i = 1, 2, 3, ...n are the locations of warehouses, $Y_i = 1$ if location i is chosen. X_{ij} is te qty. transported from warehouse i to node j. f_i is the cost of creating warehouse i and c_{ij} is the cost of transporting goods from centre i to node j. The limit of the link i->j is u_{ij}, called a capacitated problem.
Hence te problem is minimize $\sum f_i Y_i + \sum\sum c_{ij} X_{ij}$. The maximum storage of warehouse i is a_i. Requirement of Node j is d_j.
Subject to the constraints,

1. $\sum Y_i = p$
2. $X_{ij} \leq u_{ij}$
3. $\sum X_{ij} \leq a_i Y_i$
4. $\sum X_{ij} \geq d_j$
5. $X_{ij} \geq 0$ and $Y_i \quad \in \quad \{0, 1\}$

5 Travelling Salesman Problem - TSP

We Assume that the distance matrix $\{d_{ij}\}$ is given, D must be a square matrix, must be symmetric (Euclidean Space) and the elements d_{ij} satisfy triangle inequality : $d_{ij} + d_{ik} > d_{jk}$. This ensures that no intermediate city/node will be visited twice. Further $X_{ij} = 1$ if the salesman visits node j immediately after node i.
The aim of TSP is to complete a tour. A tour is a path that visits all cities and comes back home.
Sub-tours are a collection of independent paths that start from different homes and cover all nodes when considered together but that is not desirable since we want a single continuous path tour as a solution.
A solution to TSP cannot not include sub-tours. Sub-tours of L = 1 can be avoided by setting diagonal elements of D to infinity.
Formulation:

1. $\sum X_{ij} = 1 \quad \forall i$
2. $\sum X_{ij} = 1 \quad \forall j$
3. To eliminate sub-tours of length 2 $X_{ij} + X_{ji} \leq 1 \quad \forall i, j$. There will be ${}^n\mathcal{C}_2$ such constraints.
4. To eliminate sub-tours of length 2 $X_{ij} + X_{jk} + X_{jk} \leq 2 \quad \forall i, j, k$. There are ${}^n\mathcal{C}_3$ such.

But this way the number of constraints blows up.
We can reduce constraints by observing that, if n = 5 say, then eliminating L = 1 and L = 2 sub tours also removes L = 3 and L = 4 sub-tours. Since an L = k sub-tour leaves a total of n - k nodes unaccounted which can have only $L \leq n - k$ type sub-tours left.
Hence if n is even, then, the number of constraints is $\sum_{r=1}^{n/2} {}^n\mathcal{C}_r$ and if n is odd then $\sum_{r=1}^{(n-1)/2} {}^n\mathcal{C}_r$

6 Job Shop Scheduling Problem

Consider that there is a Job Shop that produces three jobs A, B and C. Each of them requires to be on Machines M1, M2 and M3 in different orders for different amounts of time.
Consider the flow of creation to be as follows for them

$$\begin{array}{cccc} A & M1 & M2 & M3 \\ B & M2 & M1 & - \\ C & M3 & M2 & - \end{array}$$

Formulation:
t_{ij} is the start time of job i on machine j. p_{ij} is the time spent by job i on machine j.
Then some constraints will be-

1. $t_{12} \geq t_{11} + p_{11}$
2. $t_{13} \geq t_{12} + p_{12}$. Similarly for Jobs 2 and 3.
3. There will also be Either - OR type constraints.
   (a) $t_{21} \geq t_{11} + p_{11}$ or $t_{11} \geq t_{21} + p_{21}$
   (b) $t_{13} \geq t_{33} + p_{33}$ or $t_{33} \geq t_{13} + p_{13}$
   (c) $t_{12} \geq t_{22} + p_{22}$ or $t_{22} \geq t_{12} + p_{12}$
   $t_{22} \geq t_{32} + p_{32}$ or $t_{32} \geq t_{22} + p_{22}$
   $t_{32} \geq t_{12} + p_{12}$ or $t_{12} \geq t_{32} + p_{32}$

Model using IP :
Let δ_{ijk} be decision variables, where k is machine number and i and j are competing jobs. Then constraint 3(a) can be modified as

$$t_{11} + p_{11} - t_{21} \leq M\delta_{121}$$

$$t_{21} + p_{21} - t_{11} \leq M(1 - \delta_{121})$$

Where M is large (like INF) and delta is a decision boolean variables variable.
Aim: Minimize time required to make all 3 : make Span
The completion time of A, B and C are $t_{13} + p_{13}, t_{21} + p_{21}, t_{32} + p_{32}$. Aim is to minimize the maximum of these 3.
An alternate aim could be to meet individual deadlines for all the jobs i, d_1, d_2, d_3 and to minimize the total delay (if any delay at all).
Tardiness for Job 1 is defined as

$$u_1 = Max(0, \quad t_{13} + p_{13} - d_1)$$

New Objective is to minimize $u_1 + u_2 + u_3$

7 Assembly Line balancing Problem

There are assembly lines that manufacture items A, B, C.... The time required to make one of these 'pieces' to the product is described below with any prerequisites. If there were only 1 workstation then every 51s the assembly line would produce a product.
The other extreme is to have 10 workstations with a steady state output of 8s/product.
The problem specifies a cap on production time and we need to use the minimum number of workstations to satisfy the criteria.

Item	Duration, Total = 51s	Precedence
A	8	-
B	6	-
C	5	A,B
D	4	...
E	6	...
F	3	...
G	7	...
H	3	...
I	4	...
J	5	...

In this example if $t_{max} = 15s$ then we will require at least 4 workstations.
Formulation : $S_j = 1$, if station j is chosen and $= 0$ otherwise. $X_{ij} = 1$ if operation i is performed at station j $= 0$ otherwise. T is the time constraint and p_i's are the durations

$$\sum_{j=1}^{10} X_{ij} = 1 \quad \forall i$$

$$\sum_{j=1}^{10} p_i X_{ij} \leq T \quad \forall j$$

To account for precedence: If operation i needs to be done before operation k. Then we have a set of constraints

1. $X_{i1} \geq X_{k1}$
2. $X_{i1} + X_{i2} \geq X_{k2}$ similarly many more..
3. $\sum_i X_{ij} \leq 10S_j \quad \forall j$ since the total number of items at station j must be less than 10 if the station is active.

We must minimize the number of workstations $\sum S_j$ subject to above constraints
Since there are large number of constraints we neglect options which we know we can do better than (using algos)

8 Knapsack problem

There is a hiker who wishes to carry a subset of certain items with him. Further each item has a value associated with it. Given this set of items, it is to be determined what the number/discretized quantity of each item should be so that the total weight of the knapsack is less than a given limit and the total value is as large as possible.

Formulation

Subject to

$$\sum_{j=1}^{n} a_{ij} \leq b_i \qquad i = 1, 2, ..., m$$

$$x_j \geq 0 \quad x_j \in \mathcal{Z}$$

We want to maximize

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

Solving discussed in part two of notes.

9 Bin Packing Problem BPP

Objects occupying different volumes are to be packed into a countable number of bins such that the number of bins used is minimized.
Variations are lattice closest packing.
Applications include