Zero-One Integer Problems

- Zero-one linear programming involves problems in which the variables are restricted to be either 0 or 1.
- Note that any bounded integer variable can be expressed as a combination of binary variables.

Capital Budgeting

- A firm has n projects that it would like to undertake but because of budget limitations not all can be selected.
- In particular project j is expected to produce a revenue of c_j but requires an investment of a_{ij} in the time period i for $i \in 1 \ldots m$.
- The capital available in time period is b_i .

Capital Budgeting

- The problem of maximising revenue subject to i the budget constraints can be formulated as follows: let $x_j = 0$ or 1 correspond to not proceeding or respectively proceeding with project j then we have to

$$
\begin{aligned}
&\max && \sum_{j=1}^{n} c_j x_j \\
&\text{subject to} && \sum_{j=1}^{n} a_{ij} x_j \leq b_i; \; i = 1, \ldots, m \\
& && 0 \leq x_j \leq 1 \quad x_j \text{ integer} \quad j = 1, \ldots, n
\end{aligned}
$$

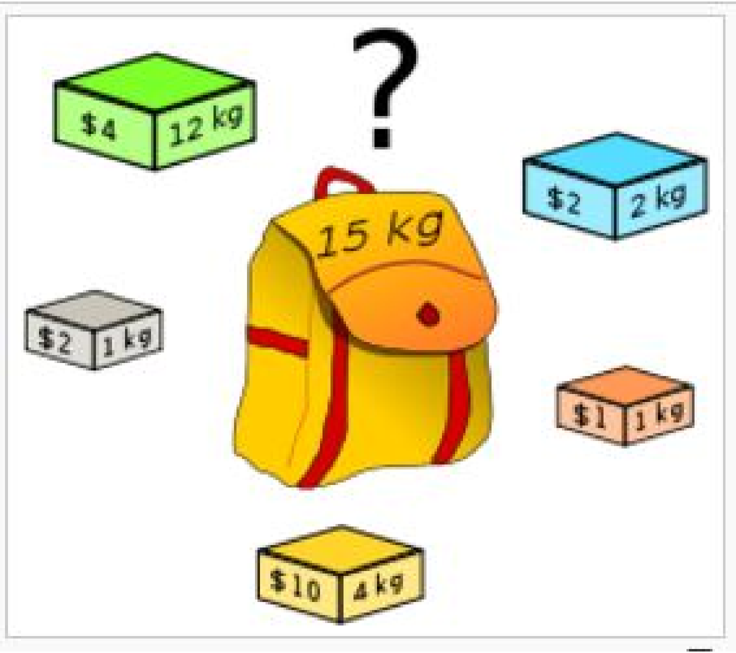

Figure:

The Knapsack Problem

- The knapsack problem or rucksack problem is a problem in combinatorial optimization
- Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

Knapsack Problems

- Remark: The 0-1 knapsack problems is an optimization problem.
- Brute force: Try all 2^n possible subsets (Recall definition of Power Set) .
- Question: Any solution better than the brute-force?

Knapsack Problems

- The most common problem being solved is the 0-1 knapsack problem, which restricts the number xi of copies of each kind of item to zero or one.
- Given a set of n items numbered from 1 up to n, each with a weight wi and a value vi, along with a maximum weight capacity W,

$$\text{maximize} \sum_{i=1}^{n} v_i x_i$$
$$\text{subject to} \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \in \{0,1\}.$$

Figure:

Knapsack Problems

Here xi represents the number of instances of item i to include in the knapsack. Informally, the problem is to maximize the sum of the values of the items in the knapsack so that the sum of the weights is less than or equal to the knapsack's capacity.

Knapsack Problems

The bounded knapsack problem (BKP) removes the restriction that there is only one of each item, but restricts the number x_i of copies of each kind of item to an integer value c_i:

$$\text{maximize} \sum_{i=1}^{n} v_i x_i$$

$$\text{subject to} \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \in \{0, \ldots, c_i\}$$

Knapsack Problems

The unbounded knapsack problem (UKP) places no upper bound on the number of copies of each kind of item and can be formulated as above except for that the only restriction on x_i is that it is a non-negative integer.

$$\text{maximize} \sum_{i=1}^{n} v_i x_i$$

$$\text{subject to} \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \geq 0$$

Merrill Lynch is considering investments into 6 projects: A, B, C, D, E and F. Each project has an initial cost, an expected profit rate (one year from now) expressed as a percentage of the initial cost, and an associated risk of failure. These numbers are given in the table below:

	A	B	C	D	E	F
Initial cost (in M)	1.3	0.8	0.6	1.8	1.2	2.4
Profit rate	10%	20%	20%	10%	10%	10%
Failure risk	6%	4%	6%	5%	5%	4%

a) Provide a formulation to choose the projects that maximize total expected profit, such that Merrill Lynch does not invest more than 4M dollars and its average failure risk is not over 5%.
For example, if Merrill Lynch invests only into A and B, it invests only 2.1M dollars and its average failure risk is (6%+4%)/2=5%.

b) Suppose that if A is chosen, B must be chosen. Modify your formulation.

c) Suppose that if C and D are chosen, E must be chosen. Modify your formulation.

Branch and Bound

The branch and bound method is a solution approach that partitions the feasible solution space into smaller subsets of solutions

Depot Location Problem

Cutting Plane Algorithm

The rationale behind this approach is :

1. Solve the continuous problem as an LP i.e. ignore integrality
2. If by chance the optimal basic variables are all integer then the optimum solution has been found. Otherwise:
3. **Generate a cut** - i.e. a constraint which is satisfied by all integer solutions to the problem but not by the current L.P. solution.
4. Add this new constraint and go to 1 .

Fathoming

1. Pruning by Optimality
2. Pruning by Bound
3. Pruning by Infeasibility