Optimization Techniques

Laboratory 4

Variable neighborhood search

UNIVERSITY OF TRENTO - Italy
Information Engineering
and Computer Science Department

0-1 Knapsack Problem

Assume

$$N = \text{number of items}$$

$$X = \{x_1, x_2, \ldots, x_N\} = \text{set of items}$$

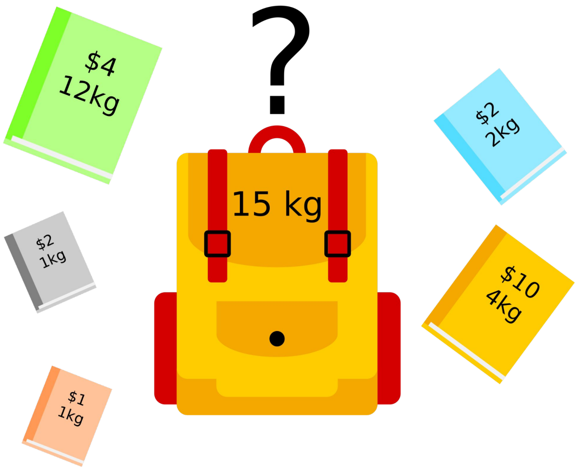
$$v_i = \text{value of item } x_i$$

$$w_i = \text{weight of item } x_i$$

$$W = \text{maximum weight the knapsack can hold}$$

Then the 0/1 knapsack problem can be formulated as follows:

$$\begin{array}{ll} \text{maximize} & \sum_{x_i \in X} v_i x_i \\ \text{subject to} & \sum_{x_i \in X} w_i x_i \leq W \\ & x_i \in \{0,1\} \forall x_i \in X \end{array}$$



Variable Neighborhood Search

Systematically change the neighbourhood during search

Procedure Basic VNS
- select $\{N_k\}$, k = 1, ..., k_{max}
- find an initial solution x
- choose a stopping condition
- repeat until stopping condition is met:
  - k ← 1
  - repeat until k = k_{max}
    - x' ← RandomSolution(N_k (x))
    - x'' ← LocalSearch(x')
    - if f(x'') < f(x) then
      - x ← x''
      - k ← 1
    - else
      - k ← k + 1

End

Function `BestImprovement`(x)

repeat
- $x' \leftarrow x$
- $x \leftarrow arg\min_{y \in N(x)} f(y)$

until $(f(x) \geq f(x'))$

return x

Function `FirstImprovement`(x)

repeat
- $x' \leftarrow x;\quad i \leftarrow 0$
- **repeat**
  - $i \leftarrow i + 1$
  - $x \leftarrow arg\min\{f(x), f(x^i)\},\quad x^i \in N(x)$
- **until** $(f(x) < f(x')$ or $i = |N(x)|)$

until $(f(x) \geq f(x'))$

return x

Reduces VNS (RVNS)

No local search to save time

Random solutions are generated from increasingly larger neighborhoods of x

- Can converge way faster
- Suitable when local search is expensive

```
Procedure Reduced VNS
    select {N_k}, k = 1, ..., k_max
    find an initial solution x
    choose a stopping condition
    k ← 1
    repeat until k = k_max
        x' ← RandomSolution(N_k (x))
        if f(x') < f(x) then
            x ← x'
            k ← 1
        else
            k ← k + 1
End
```