Optimization Techniques

Laboratory 5

Iterated Local Search, Simulated Annealing

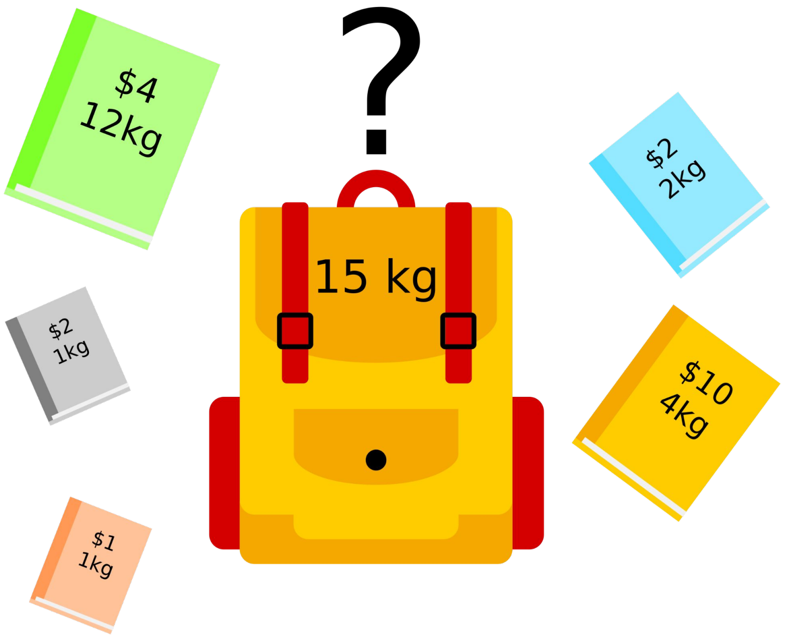
cc.rambaldimigliore@unitn.it
elia.cunegatti@unitn.it
mvincze@fbk.eu

UNIVERSITY OF TRENTO - Italy
Information Engineering
and Computer Science Department

0-1 Knapsack Problem

Maximize the sum of values in the bag bounded by the total weight

- Bag has capacity *W*
- Each item has
  - volume *w*
  - value *v*



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:

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

Iterated Local Search

<u>Stochastic Local Search method</u> based on an embedded heuristic

Given solution ***s****:

1. Perturb ***s**** leading to intermediate state ***s'***
1. Perform Local Search around ***s'***, arriving at solution ***s'****
1. If **s'*** passes the acceptance test, then **s*** = **s'***
1. Back to step 1 until convergence

procedure *Iterated Local Search*
$s_0 \leftarrow$ GenerateInitialSolution
$s^* \leftarrow$ LocalSearch(s_0)
repeat
$s' \leftarrow$ Perturbation$(s^*, \textit{history})$
$s^{*\prime} \leftarrow$ LocalSearch(s')
$s^* \leftarrow$ AcceptanceCriterion$(s^*, s^{*\prime}, \textit{history})$
until termination condition met
end

Better than random restarts as we do not completely give up on the current solution

Simulated Annealing

Improves greedy search
by allowing step to a worse neighbour

- **Neighbor selection**
  transform the given solution
  to introduce a new solution

- **Temperature scheduling**
  willingness to accept a worse state
  decreased over time

- **Acceptance probability**
  probability of going from one state to a neighbor state

- $X \leftarrow$ Initial configuration
- $T \leftarrow$ Initial high temperature
- Iterate:

1. Do K times:
   1.1 $E \leftarrow Eval(X)$
   1.2 $X' \leftarrow$ one configuration randomly selected in $Neighbors(X)$
   1.3 $E' \leftarrow Eval(X')$
   1.4 If $E' \leq E$
   $X \leftarrow X'$; $E \leftarrow E'$;
   Else accept the move with probability $p = e^{-(E'-E)/T}$
   $X \leftarrow X'$; $E \leftarrow E'$;
2. $T \leftarrow \alpha T$