Three dimensional Knapsack problem with Orthogonal Packing .

Student Project

June 12, 2019

Given list of requested products (referred to as items) that come in their box of given height, width, and length (referred to as dimensions A, B and C), the problem is to pack as subset of them (possibly all) in a large box (refereed to as the knapsack) which dimensions are given by its height, width, and length.

The items can be rotated in any position, permutatting the role of their height, width, and length: i.e. each item has six positions : 3 ways to select the height, and for each way, there are 2 ways to select which is the width.

The items have a value associate to them. Hence, if all the items cannot fit into the knapsack, the goal is to select a subset of them that can fit and that maximises the total value of the selected items.

Hence a solution to the problem specifies the total value of the selected items, the list of the selected items and a description of the way to pack them into the knapsack.

The way to describe the packing is open, provided one can easily test that indeed this packing is feasible. A classical way to specify a packing is to designate a corner of the knapsack as being the origin - say the bottom left corner, and the dimensions of knapsack as being the (x, y, z) coordinates axis. Then each of the selected items receives a position : the (x,y,z) coordinates of its bottom left corner and an orientation. The orientation is defined by which dimension (among A, B and C) is the heigh and which dimension is the width.

1 Notes

There are many symmetric placements for the same selection of items. Make sure that if you proceed by enumeration, you avoid to enumerate all alternative placements of the same subset of items.

The goal is to output a "good" solution; the best selection that you can find, in a very short computing time. Hence it is recommended that you use parallel processing of several trial solutions so as to identify good solution very quickly.

The best method is the one that offers the best curve of solution quality over computing-time.

Test problems can be generated randomly with say a list of 5 to 20 items, 10 of which can fit in the knapsack on average.

The placement must be easy to deliver by a human operator; so you can restrict you placement consideration to the ones that a human being can easily implement.

2 Input

(H, W, L) are the dimensions of the knapsack
I is the set of items, with for each $i \in I$
v_i is the value of item i
(a_i, b_i, c_i) are the dimensions of item i

3 Output

V is the value of the solution.
$J \subset I$ is the set of select items, with for each $i \in J$,
(x_i, y, z_i) the coordinates of its bottom left corner;
(h_i, w_i) the dimension selected as hight and width .