问题描述：它是带有一组约束（重量 尺寸 喜欢度 可买性 可靠性等）的背包问题。需要从约束条件中求出符合约束条件中的最大价值之和。比如有N个物品，价值为比、数组v[i]存起来。那么约束条件吗M，即w1、w2····wm。总重为C[i];约束条件在比C[i]小于等于情况下寻找最大价值bestp。可借鉴0-1背包问题的回溯法解决，只是把一维变成多维度。

MAXVi<= C;

Xi ={0,1} i=1,2---n;

复杂性分析：

核心代码：

for (int j = 0; j < m; j++)

put[j] += w[j][i];

cp += v[i];

backtrack(i + 1);

for (int j = 0; j < m; j++)

put[j] -= w[j][i];//此处回溯

cp -= v[i];

因为在0-1背包中是一维的，所以在约束条件个数m中需要比较m次，当符合条件的时候就把价值给加进来，如果条件不符合就会还原数据，把原来加的属性和价值给减去。回溯回去。并继续下一个数据进行调用backtrack(i + 1)，循坏下去，以此比较出最优价值。

(cw - c[j] <= 0)；限界函数，根据当前的背包是否超出范围从而决定去取不取。

在最坏情况下会有O（2n）个右儿子结点。所以在背包问题回溯法中算法回溯函数backtrack需要的计算时间复杂度为O（n2n）。