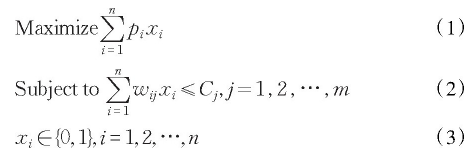
多维背包算法设计与分析

多维背包问题是将n个物品放入背包中，每个物品i有一个非负价值p i，并且占用m个维度的资源w i1，w i2，...，w im ，背包也有m个维度，并且每个维度有一个资源的限制C j，j = {1,2，...，m}。要求选择若干物品装入背包，使得物品在每个维度不超过背包限制的前提下装入物品的价值和最大。

用数学公式表示如公式（1）〜（3）所示：



算法设计：回溯法解多维背包问题，i= {1,2，...，n},j = {1,2，...，m},用数组cw[i]来存当前背包各维度的值，用变量cv来存当前背包价值，

用二维数组w[j][i]来表示物品i的各属性，用数组c[j]来存背包各维度的限制，v[i]来存物品的价值，用变量bestv表示最近最优的解，如果

cw[i]+w[j][i]<=c[j],即背包各维度当前的值分别加物品i的各属性的值小于或等于背包各维度的限界，则将物品放入背包，cv=cv+v[i],递归，进入下一层搜索。如果cw[i]+w[j][i]>c[j],则递归，进入下一层搜索。当i>=n时，cv与bestv比较，如果cv>bestv,则将cv的值赋给bestv。回溯，到没有活结点时，结束搜索，输出最优解bestv。

多维背包是01背包问题的一个变形，01背包问题的约束条件只有一个重量约束，而多维背包问题有多个约束条件。算法实现可以参照01背包问题的实现。

复杂度分析：

因为物品只有选与不选2个决策，而总共有n个物品，且每个物品有m个属性，所以用回溯法暴力搜索的时间复杂度为T（n）=O(m2n) ，

有n个物品,即最多递归n层,每个物品有m个属性，存储物品信息就是一个二维数组,即回溯法求解多维背包问题的空间复杂度为O(nm)。