整个流程与队列式分支限界法的 01 背包问题一样，只是插入活结点的操作：由 insert()改为 push()，重新选择下一个扩展结点的操作：由 delete()改为 pop()。用一个 while循环，循环结束条件是活结点列表为空，即当前扩展结点为空，表 示所有可行的结点都已经得到遍历；并且输出最优解。 每一次循环作用是：在当前的扩展结点下，将以当前扩展结点作为父结点的下一 层子结点都加入到活结点列表里，并且选择一个新的扩展结点，进入下一层循环； 其中，剪枝函数包括左结点的容量约束（约束函数）和右结点的最大价值判断（界限函数）；

栈式分支限界法与回溯法的区别：

栈式分支限界类似于深度优先，各结点只有一次机会成为扩展结点。

回溯法是深度优先，各结点的遍历次序是： A->B->D->B->E->A->C->F->C->G而且结点 A,B,C 都两次成为扩展结点。