目　录

第一章 问题定义

1.1 问题描述

用下列几种方法解决 0-1 背包问题，编程实现并对比各算法的求解质量和复杂度：

1. 贪婪算法
2. 动态规划
3. 分支限界或回溯
4. 模拟退火或遗传算法（随机算法运行 30run）

1.2 问题生成

实验中，设置问题规模 N = 500，然后随机生成重量范围在 1-100 和价值范围在 1-100 的 N 个物体，重量和价值是独立无关的，背包容量是设置成物体总重量的二分之一。

- 问题生成的代码文件是：source/gen_problem.py
- 生成的问题保存在：source/problem.txt 格式是
  - 第一行为背包容量和物体个数，V 和 N
  - 第二行到第 N+1 行是每个物体的重量和价值，w_i 和 v_i

第二章 算法过程

2.1 贪婪算法

实验中，采用了三种贪心策略进行对比：

- 第一种策略是每次先取具有最高价值的物体，直到背包不能再装下任何物体
- 第二种策略是每次先取具有最小重量的物体，直到背包不能再装下任何物体
- 第三种策略是每次先取单位重量价值（v_i / w_i）最大的物体，直到背包不能再装下任何物体

  源代码文件是：source/greedy.py

2.1.1 贪婪策略一

运行结果如图 1：

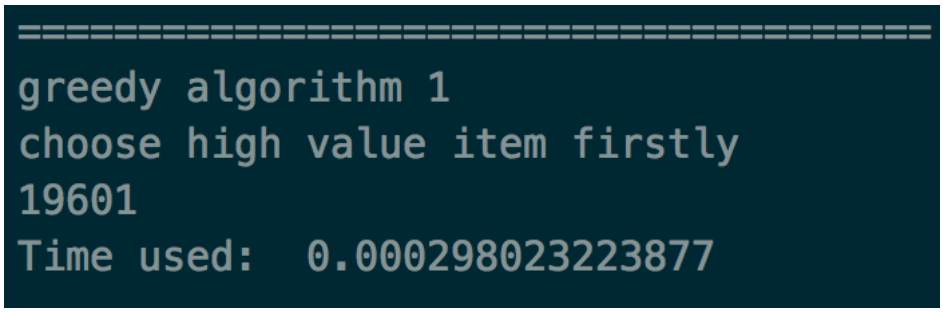

图 1 贪婪算法 1

2.1.2 贪婪策略二

运行结果如图 2：

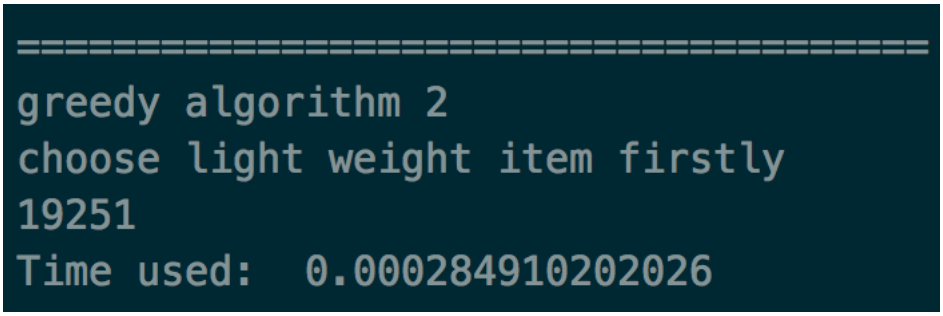

图 2 贪婪算法 2

2.1.3 贪婪策略三

运行结果如图 3：

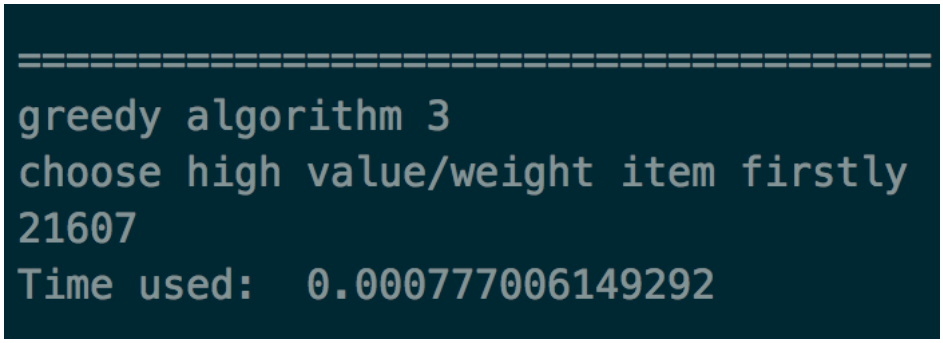

图 3 贪婪算法 3

从三种策略的运行结果可知，第三种贪婪策略（先取单位重量价值最高的物体）得到的结果最好。

2.2 动态规划

问题的状态转移方程如下：

f[i][v] = max{f[i-1][v], f[i-1][v-w[i]] + v[i]}

源代码文件是：source/dp.py

运行结果如图 4：

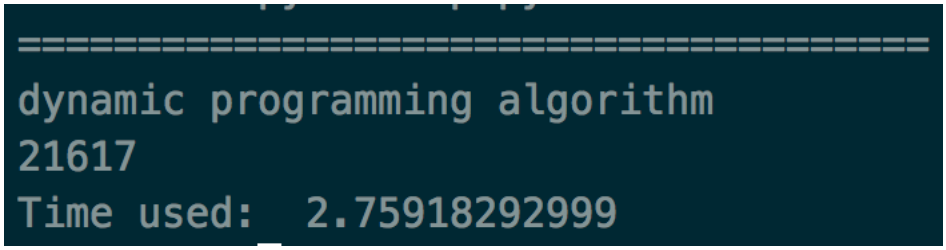

图 4 动态规划算法

2.3 回溯法

按照单位重量价值从大到小对物品进行排列，用深度优先递归地遍历子集树，设置限界函数减去得不到最优解的子树，限界函数主要考虑如果子树的最大上界都没有当前最好解大的话，就进行剪枝。

源代码文件是：source/backtrack.py

运行结果如图 5：

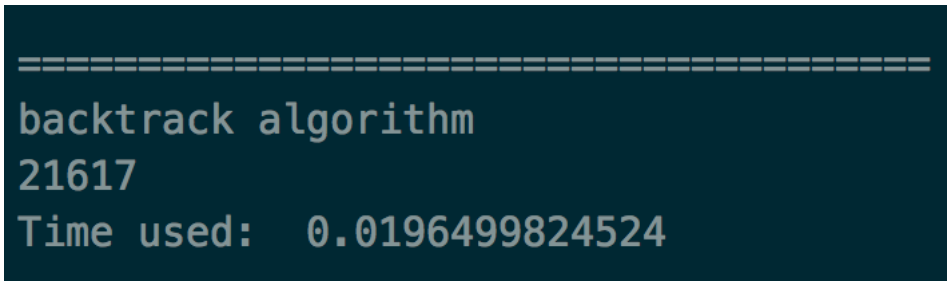

图 5 回溯算法

2.4 模拟退火算法

2.4.1 算法流程

(1) 随机生成初始解 state0，停止准则设置为温度小于参数 beta，目标函数为整个背包中的物品的价值总和。

(2) 接受准则：当新解的总重量不超过背包容量时，如新解优于原解，则接受它，否则以一定的概率接受它

(3) 构造领域：随机选择一个物品 i，如果其在背包中，则直接将其取出，或同时随机放入一个物品 j；如果物品 i 不在背包中，则将直接其放入背包，或同时随机取出一个物品 j[1]。

(4) 冷却机制：T = alpha * T

(5) 为了提高解的质量，将算法过程中出现的最优解保存下来，并作为最终解返回。

2.4.2 算法参数实验

模拟退火算法中，初始温度、停止系数、冷却系数和迭代次数的选择都会影响最终的结果，因此，对这些参数分别进行实验，便于选择最优参数组合。

实验代码文件是: source/SA_experiment.py

(1) 不同初始温度 t0 的对照结果，如表 1 所示：

表 1 不同初始温度结果

t0	5	20	50	70	100	200	500
最优解	21588	21593	21587	21594	21587	21591	21599

(2) 不同停止系数 beta 值的对照结果，如表 2 所示：

表 2 不同停止系数结果

beta	0.0001	0.001	0.01	0.1	1.0	10.0
最优解	21579	21577	21583	21559	21561	21015

(3) 不同冷却系数 alpha 值的对照结果，如表 3 所示：

表 3 不同冷却系数结果

alpha	0.1	0.3	0.5	0.7	0.9
最优解	21435	21513	21546	21569	21588

(4) 不同最大迭代书 max_iters 值的对照结果，如表 4 所示：

表 4 不同最大迭代数结果

max_iters	N	5 * N	10 * N	20 * N	50 * N	100 * N
最优解	21462	21540	21587	21602	21595	21605

2.4.3 算法迭代过程

选择参数 t0 = 100，beta = 0.01，alpha = 0.9，max_iters = 10 * N，画出算法最优解的迭代过程如图 6 所示：

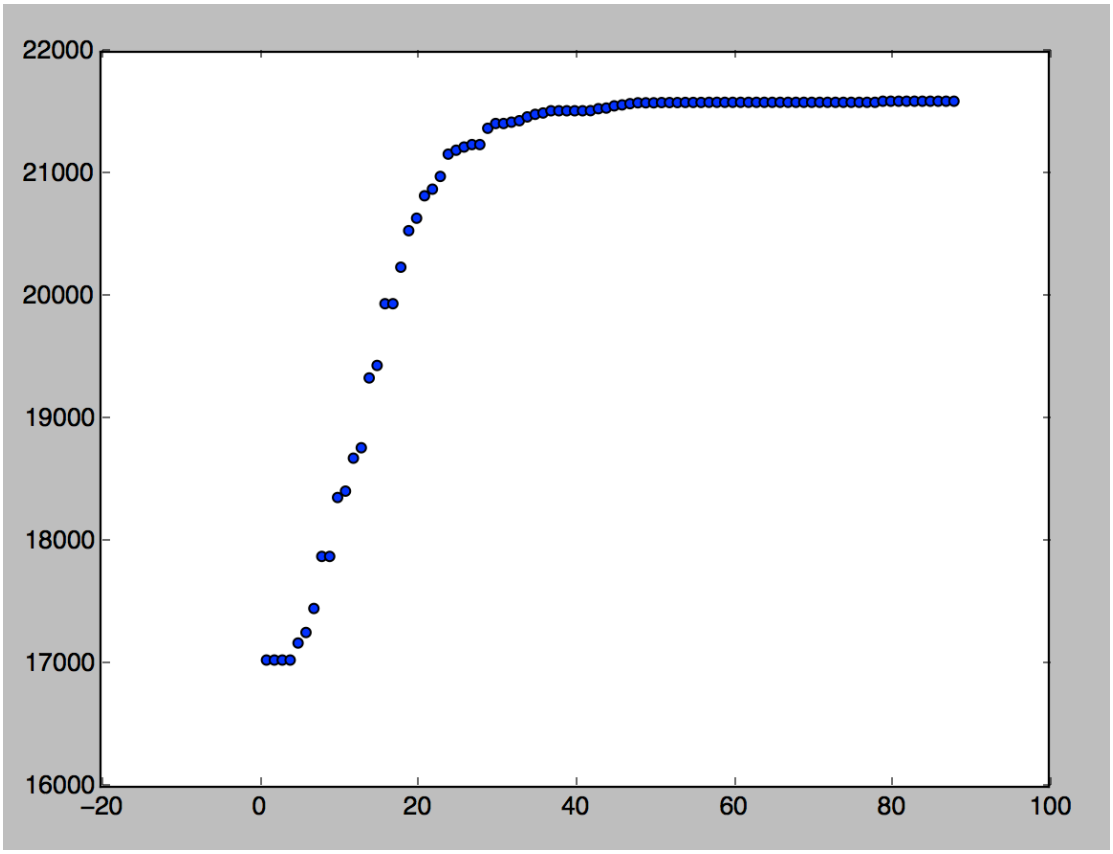

图 6 算法迭代过程

第三章 算法对比

不同算法的运行结果和时间对比如表 5 所示：

表 5

算法	贪婪算法	动态规划	回溯法	模拟退火法
运行时间	0.000777s	2.759s	0.196s	73.88s
最优解	21607	21617	21617	21587

参考文献

[1]. https://github.com/xmuliushuo/algorithms/tree/master/Knapsack