

Họ và Tên Phan Văn Thiện
MSSV 21522628
Lớp CS106.N21

TRÍ TUỆ NHÂN TẠO CS106

GV hướng dẫn: Lương Ngọc Hoàng

Assignment 2 – Solving Knapsack problems using Google OR tools.

I. Tổng quan

- Dạng bài toán: 0/1 Knapsack problems.
 - Một bài toán tối ưu hóa trong lĩnh vực tối ưu hóa tổ hợp.
 - Mục tiêu bài toán là đóng một cái túi với khả năng chứa giới hạn, sao cho giá trị của các vật phẩm chứa trong túi là lớn nhất có thể.
- Dữ liệu test:
 - Bao gồm 13 nhóm test case:
 - 00 Uncorrelated
 - 01 WeaklyCorrelated
 - 02 StronglyCorrelated
 - 03 InverseStronglyCorrelated
 - 04 AlmostStronglyCorrelated
 - 05 SubsetSum
 - 06 UncorrelatedWithSimilarWeights
 - 07 SpannerUncorrelated
 - 08 SpannerWeaklyCorrelated
 - 09 SpannerStronglyCorrelated
 - 10 MultipleStronglyCorrelated
 - 11 ProfitCeiling
 - 12 Circle
 - Mỗi test case sẽ chọn ra 5 test case với số lượng khác nhau (50, 100, 200, 500, 1000)

II. Phương pháp thực hiện

1. Tiêu chí đánh giá tính tối ưu:
 - Lời giải tìm được được coi là tối ưu nếu thời gian để tìm ra lời giải nhỏ hơn hoặc bằng với `time_limit`.
2. Giải bài toán Knapsack sử dụng Ortools
- Cài đặt thư viện :

```
!pip install ortools
```

- Git Clone dữ liệu testcase từ github về thư mục `./sampledata`:

```
!git clone https://github.com/likr/kplib ./sampledata
```

- Khai báo các thư viện

```
from ortools.algorithms import pywrapknapsack_solver # Thư viện OR-Tools
import os # Thư viện OS để list các thư mục Test case
from time import time # Thư viện time để đo thời gian chạy của CT
```

- Khai báo đường dẫn:

```
current_path = "./sampledata/"
folders = sorted(os.listdir(current_path))[1:-1]
subfolders = ['n00050', 'n00100', 'n00200', 'n00500', 'n01000']
```

- Hàm giải và in lời giải:

```
def Solver(input_url):
    input = open(input_url, "r").read().splitlines()
    capacities = [int(input[2])]
    size = input[1]
    values = []
    weights = [[]]
    for i in input[4:]:
        values.append(int(i.split()[0]))
        weights[0].append(int(i.split()[1]))
    solver = pywrapknapsack_solver.KnapsackSolver(
        pywrapknapsack_solver.KnapsackSolver.
        KNAPSACK_MULTIDIMENSION_BRANCH_AND_BOUND_SOLVER, 'Knapsack')
    solver.set_time_limit(150)
    solver.Init(values, weights, capacities)
    start_time = time()
    computed_value = solver.Solve()
    packed_items = []
    packed_weights = []
    total_weight = 0
    for i in range(len(values)):
        if solver.BestSolutionContains(i):
            packed_items.append(i)
            packed_weights.append(weights[0][i])
            total_weight += weights[0][i]
    end_time = time()
    runtime = end_time - start_time
    print(input_url[-22:-1], computed_value, total_weight, runtime, runtime <= 150)
def run(index):
    for subfolder in subfolders:
        url = current_path + folders[index] + "/" + subfolder + "/R01000/s000.kp"
        print("| " + url)
        Solver(url)
```

- Chạy chương trình

```
for i in range(13):
    run(i)
```

3. Thống kê và đánh giá kết quả

a. Bảng thống kê

	Type	Direction	Total value	Total weight	Runtime	Optimal
1	00Uncorrelated	/n00050/R01000/s000.k	20995	14721	0.000	True
2		/n00100/R01000/s000.k	46537	22519	0.000	True
3		/n00200/R01000/s000.k	84317	50302	0.002	True
4		/n00500/R01000/s000.k	207992	118693	0.000	True
5		/n01000/R01000/s000.k	400811	252480	0.000	True
6	01WeaklyCorrelated	/n00050/R01000/s000.k	15768	14232	0.000	True
7		/n00100/R01000/s000.k	31064	29013	0	True
8		/n00200/R01000/s000.k	56976	51563	0	True
9		/n00500/R01000/s000.k	139258	127276	0	True
10		/n01000/R01000/s000.k	273052	245972	0.005	True
11	02StronglyCorrelated	/n00050/R01000/s000.k	17539	14239	0.035	True
12		/n00100/R01000/s000.k	35617	29017	0.199	True
13		/n00200/R01000/s000.k	65363	51563	150.109	False
14		/n00500/R01000/s000.k	162178	127278	150.202	False
15		/n01000/R01000/s000.k	316372	245972	150.087	False
16	03InverseStronglyCorrelated	/n00050/R01000/s000.k	14914	16714	2.871	True
17		/n00100/R01000/s000.k	30468	33968	0.000	True
18		/n00200/R01000/s000.k	54964	61464	0.010	True
19		/n00500/R01000/s000.k	136031	152031	150.175	False
20		/n01000/R01000/s000.k	263977	295477	150.095	False
21	04AlmostStronglyCorrelated	/n00050/R01000/s000.k	17556	14238	0.017	True
22		/n00100/R01000/s000.k	35611	29016	0.075	True
23		/n00200/R01000/s000.k	65385	51563	150.031	False
24		/n00500/R01000/s000.k	162154	127278	150.014	False
25		/n01000/R01000/s000.k	316415	245972	0.217	True

26	05 SubsetSum	/n00050/R01000/s000.k	14239	14239	0.000	True
27		/n00100/R01000/s000.k	29017	29017	0.000	True
28		/n00200/R01000/s000.k	51563	51563	0.000	True
29		/n00500/R01000/s000.k	127278	127278	0.000	True
30		/n01000/R01000/s000.k	245972	245972	0.000	True
31	06UncorrelatedWithSimilarWeights	/n00050/R01000/s000.k	19676	2401482	0.034	True
32		/n00100/R01000/s000.k	39791	4902253	7.541	True
33		/n00200/R01000/s000.k	75678	9904900	0.008	True
34		/n00500/R01000/s000.k	189769	24712055	150.051	False
35		/n01000/R01000/s000.k	371246	49525319	0.070	True
36	07SpannerUncorrelated	/n00050/R01000/s000.k	13472	4569	0.000	True
37		/n00100/R01000/s000.k	24228	8748	141.97	True
38		/n00200/R01000/s000.k	47836	17274	150.115	False
39		/n00500/R01000/s000.k	114616	42898	150.634	False
40		/n01000/R01000/s000.k	228624	84656	150.068	False
41	08SpannerWeaklyCorrelated	/n00050/R01000/s000.k	10354	11452	1.698	True
42		/n00100/R01000/s000.k	20550	20824	0.000	True
43		/n00200/R01000/s000.k	40575	41116	19.814	True
44		/n00500/R01000/s000.k	98713	100076	145.697	True
45		/n01000/R01000/s000.k	196050	198664	150.059	False
46	09SpannerStronglyCorrelated	/n00050/R01000/s000.k	28440	11540	150.042	False
47		/n00100/R01000/s000.k	51656	20956	10.511	True
48		/n00200/R01000/s000.k	101888	41288	150.041	False
49		/n00500/R01000/s000.k	245128	99928	150.050	False
50		/n01000/R01000/s000.k	488672	198772	150.040	False
51	10MultipleStronglyCorrelated	/n00050/R01000/s000.k	21338	14238	0.004	True
52		/n00100/R01000/s000.k	43316	29016	1.471	True
53		/n00200/R01000/s000.k	81658	51558	23.422	True
54		/n00500/R01000/s000.k	203778	127278	150.205	False

55		/n01000/R01000/s000.k	399170	245970	150.055	False
56	11ProfitCeiling	/n00050/R01000/s000.k	14229	14238	1.179	True
57		/n00100/R01000/s000.k	29001	29015	150.035	False
58		/n00200/R01000/s000.k	51540	51562	146.413	True
59		/n00500/R01000/s000.k	127239	127277	150.048	False
60		/n01000/R01000/s000.k	245877	245972	150.073	False
61	12Circle	/n00050/R01000/s000.k	300031	14239	1.09	True
62		/n00100/R01000/s000.k	611418	29017	0.151	True
63		/n00200/R01000/s000.k	1086483	51563	150.046	False
64		/n00500/R01000/s000.k	2681868	127278	150.057	False
65		/n01000/R01000/s000.k	5182856	245972	0.006	True

b. Đánh giá

- Các nhóm test case dễ thực hiện nhất là các nhóm test case có độ tương quan yếu hoặc không có tương quan dữ liệu. Các nhóm này dễ dàng tìm ra lời giải tối ưu với khoảng thời gian ngắn.

Bao gồm các test cases :

- o 00Uncorrelated
 - o 01WeaklyCorrelated
- Các nhóm test case khó thực hiện là các nhóm test case có độ tương quan mạnh như là:
 - o 09SpannerStronglyCorrelated: false 4/5 test case
 - o 02StronglyCorrelated : false 3/5 test case
 - o 11ProfitCeiling : false 3/5 test case

- Giải thích

Nhóm test case 02 StronglyCorrelated: Các trọng số của các món đồ trong bài toán này có mối quan hệ mạnh với nhau. Điều này có thể dẫn đến việc các bài toán con tối ưu hóa không thể được giải quyết bằng cách sử dụng giá trị giới hạn cận trên (UB) tốt nhất hiện tại và do đó, thuật toán Branch and Bound sẽ không thể tối ưu hóa tốt hơn.

Nhóm test case 11 ProfitCeiling: Bài toán này yêu cầu tối ưu hóa giá trị của các món đồ, nhưng với giá trị giới hạn cận trên (UB) được thiết lập tốt, các món đồ có giá trị lớn hơn giá trị giới hạn này sẽ bị loại bỏ sớm trong quá trình tối ưu hóa, dẫn đến thuật toán Branch and Bound cho kết quả không tốt.

Nhóm test case 09 SpannerStronglyCorrelated: Bài toán này kết hợp các đặc tính của hai nhóm test case trên. Các trọng số của các món đồ có mối quan hệ mạnh với nhau và bài toán yêu cầu tối ưu hóa giá trị. Tương tự như nhóm test case 02 StronglyCorrelated, việc tối ưu hóa bằng thuật toán Branch and Bound trên nhóm test case này cũng gặp khó khăn.