

Nguyễn Đức Anh Phúc

20520276

Bài báo cáo CS106

GoogleORTools_Knapsack

01.01 Mục tiêu báo cáo

- Mục tiêu của bài báo cáo này là:
 - + Phân tích thuật toán Knapsack và đề xuất ra 2 hướng giải quyết cho bài toán Knapsack sử dụng kỹ thuật quy hoạch động
 - + Chạy thực nghiệm kỹ thuật quy hoạch động trên bài toán Knapsack để ra được kết quả tối ưu
 - + Chạy thực nghiệm tool GoogleORTools với trình solver Knapsack sử dụng kỹ thuật nhanh cận Branch and Bounds
 - + Phân tích số liệu thu thập được từ 2 cách chạy trên ra được bản kết quả

01.02 Kỹ thuật quy hoạch động bài toán Knapsack

- Quy hoạch động là kỹ thuật để giải các bài toán tối ưu với thời gian nhanh và thường thường để giải quyết các bài toán càng nhiều constraint thì lời giải quy hoạch động sẽ càng khó khăn.

Em sẽ đề xuất hướng giải quyết cho bài toán Knapsack này:

Cho một cái vali chứa được khối lượng tối đa là W . Có n đồ vật được đánh số từ 1 đến n , đồ vật thứ i có khối lượng $weight[i]$ và giá trị $value[i]$, $1 \leq i \leq n$. Tìm cách xếp đồ vật vào vali sao cho tổng giá trị các đồ vật trong vali đạt giá trị lớn nhất.

- + Trọng lượng của vali: $W = 10$ (units).
- + Số lượng đồ vật: $N = 4$.
- + Giá trị của các đồ vật: $v[] = \{10, 40, 30, 50\}$.
- + Khối lượng của các đồ vật: $w[] = \{5, 4, 6, 3\}$.
- + Kết quả: 90.
- + Các đồ vật có thể chọn là: $\{1, 3, 4\}$ hoặc $\{2, 4\}$

01.02.01 Đặc trưng của bài toán

- Hàm mục tiêu: f : Tổng giá trị các đồ vật của vali.
- Tổng giá trị các đồ vật của vali phụ thuộc vào 2 yếu tố: có bao nhiêu vật đang được xét và tổng khối lượng của các vật. Do đó, bảng phương án sẽ là bảng 2 chiều.

01.02.02 Phương pháp đệ qui

- Ta có: W khối lượng chứa tối đa của vali.
- Ta có: n số lượng đồ vật.
- Khối lượng các đồ vật $weight[1], weight[2], weight[3], \dots, weight[n]$.
- Giá trị các đồ vật $value[1], value[2], \dots, value[n]$ $1 \leq i \leq n$.
- Đối với đồ vật thứ $[n]$ ta có hai phương án:
 - + Đồ vật thứ $[n]$ được chọn bỏ vào vali (khối lượng của đồ vật thứ $[n]$ nhỏ hơn bằng khối lượng W).
 - + Đồ vật thứ $[n]$ không được chọn bỏ vào vali.
- Tổng giá trị của các đồ vật trong vali có giá trị lớn nhất là một trong hai giá trị sau:
 - + Giá trị lớn nhất của việc sắp $(n - 1)$ đồ vật vào trong vali có khối lượng W .
 - + Giá trị của đồ vật thứ $[n]$ cộng với giá trị lớn nhất của việc sắp $(n - 1)$ đồ vật vào trong vali có khối lượng $W - weight[n]$.

01.02.03 Công thức qui hoạch động

- Gọi $L(i, w)$ là tổng giá trị lớn nhất của các đồ vật trong vali khi xét từ vật 1 tới vật i và trọng lượng của vali chưa vượt quá trọng lượng w . Trong đó $0 \leq i \leq n$ và $0 \leq w \leq W$.
- L là ma trận hai chiều kích thước $(n + 1) \times (W + 1)$.
- Dễ dàng ta thấy $L(0, w) = 0$.
- Tính $L(i, w)$: vật đang xét là vật $[i]$ có khối lượng $weight[i]$ và giá trị là $value[i]$. Vật $[i]$ được quyết định bỏ vào vali hay không có điều kiện ràng buộc là “tổng khối lượng các vật trong vali không được vượt quá trọng lượng w ”.
- Có hai khả năng xảy ra:
 - Nếu chọn đồ vật $[i]$ đưa vào vali tổng khối lượng vali trước đó phải không quá $w - weight[i]$.
Do đó $L(i, w) = L(i - 1, w - weight[i]) + value[i]$.
 - Nếu không chọn đồ vật $[i]$ tổng khối lượng vali là như cũ (như lúc trước khi chọn đồ vật $[i]$).
Do đó $L(i, w) = L(i - 1, w)$.
- Tóm lại ta có: $L(i, w) = \max(L(i - 1, w), L(i - 1, w - weight[i]) + value[i])$.

01.02.04 Bảng phương án

- Ví dụ: $W = 13, n = 5$.

	1	2	3	4	5
<i>weight[i]</i>	3	4	5	2	1
<i>value[i]</i>	4	5	6	3	1

- Bảng phương án:

$$L(i, w) = \max(L(i - 1, w), L(i - 1, w - \text{weight}[i]) + \text{value}[i])$$

[illegible]

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

[illegible]

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

[illegible]

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

[illegible]

	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

$L(i, w)$		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
$i=1$ $w_1=3$ $v_1=4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0											
	2														
	3														
	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

$L(i, w)$		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
$i=1$ $w_1=3$ $v_1=4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4										
	2														
	3														
	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

$L(i, w)$		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
$i=1$ $w_1=3$ $v_1=4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4									
	2														
	3														
	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

$L(i, w)$		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
$i=1$ $w_1=3$ $v_1=4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4	4								
	2														
	3														
	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

$L(i, w)$		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13

[illegible]

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

[illegible]

$$L(i, w) = \max(L(i - 1, w), L(i - 1, w - \text{weight}[i]) + \text{value}[i])$$

[illegible]

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

[illegible]

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

[illegible]

	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

$L(i, w)$		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
$i=1$ $w_1=3$ $v_1=4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4	4	4	4	4	4	4	4		
	2														
	3														
	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

$L(i, w)$		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
$i=1$ $w_1=3$ $v_1=4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4	4	4	4	4	4	4	4	4	
	2														
	3														
	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

$L(i, w)$		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
$i=1$ $w_1=3$ $v_1=4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4	4	4	4	4	4	4	4	4	4
	2														
	3														
	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

$L(i, w)$		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
$i=2$ $w_2=4$ $v_2=5$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4	4	4	4	4	4	4	4	4	4
	2	0													
	3														
	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

$L(i, w)$		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13

i = 2 w ₂ =4 v ₂ =5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4	4	4	4	4	4	4	4	4	4
	2	0	0												
	3														
	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

L(i, w)		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
i = 2 w ₂ =4 v ₂ =5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4	4	4	4	4	4	4	4	4	4
	2	0	0	0											
	3														
	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

L(i, w)		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
i = 2 w ₂ =4 v ₂ =5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4	4	4	4	4	4	4	4	4	4
	2	0	0	0	4										
	3														
	4														
	5														

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

L(i, w)		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
i = 2 w ₂ =4 v ₂ =5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4	4	4	4	4	4	4	4	4	4
	2	0	0	0	4	5									
	3														
	4														
	5														

.....

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

L(i, w)		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
i=5 w ₅ =1 v ₅ =1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4	4	4	4	4	4	4	4	4	4
	2	0	0	0	4	5	5	5	9	9	9	9	9	9	9
	3	0	0	0	4	5	6	6	9	10	11	11	11	15	15

	4	0	0	3	4	5	7	8	9	10	12	13	14	15	15
	5	0	1	3	4	5	7	8	9	10	12	13	14	15	

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i])$$

$L(i, w)$		W													
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
i=5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	4	4	4	4	4	4	4	4	4	4	4
	2	0	0	0	4	5	5	5	9	9	9	9	9	9	9
	3	0	0	0	4	5	6	6	9	10	11	11	11	15	15
	4	0	0	3	4	5	7	8	9	10	12	13	14	15	15
	5	0	1	3	4	5	7	8	9	10	12	13	14	15	16

- Tổng giá trị các đồ vật trong vali đạt giá trị lớn nhất là 16.

01.02.05 Truy vết

Phản truy vết thì ta có thể xử lý như sau:

- Bắt đầu từ ô $L(n, W)$ trên dòng n ta dò ngược về dòng 0 theo nguyên tắc:
 - Nếu $L(i, w) \neq L(i-1, w)$ thì đồ vật thứ i được chọn, ta truy tiếp ô $L(i-1, w - \text{weight}[i])$.
 - Nếu $L(i, w) = L(i-1, w)$ thì đồ vật thứ i không được chọn, ta truy tiếp ô $L(i-1, w)$.

01.02.06 Chuyển đổi công trạng thái

- Gọi $L(i, w)$ là tổng giá trị lớn nhất của các đồ vật trong vali khi xét từ vật 1 tới vật i và trọng lượng của vali **đúng bằng** trọng lượng w . Trong đó $0 \leq i \leq n$ và $0 \leq w \leq W$
- Khi chuyển đổi như vậy thì chúng ta sẽ đảm bảo tìm được lời giải chính xác trọng lượng w và công thức quy hoạch động vẫn là như vậy

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i]).$$

01.03 Kỹ thuật quy hoạch động bài toán Knapsack tối ưu không gian

$$L(i, w) = \max(L(i-1, w), L(i-1, w - \text{weight}[i]) + \text{value}[i]).$$

- Ta thấy được công thức quy hoạch động chúng ta chỉ cần quan tâm đến hai dòng cuối cùng nên không việc gì chúng ta lưu lại trạng thái của toàn bộ bảng phương án mà thay vào đó chỉ cần lưu lần lượt 2 dòng: $L(0, w)$ và $L(1, w)$ rồi chuyển đổi tính toán cho nhau

01.04 Bảng kết quả và đánh giá

- Em quan sát thấy được có những testcase là $\sum weight[i] \leq capacities$. Vì vậy, em có optimize lại bằng cách $capacities = \min(capacities, \sum weight[i])$ và thấy hiệu quả rõ rệt, có những case giảm gần 5 lần thời gian thực thi

```
#Optimize based on mathematical observation
total_weight=0
for weight in weights:
    total_weight=total_weight+weight
capacities=min(capacities,total_weight)
```

- Lấy những testcase này và là test số 5

```
['n00050', 'n00100', 'n00200', 'n00500', 'n01000']
['R01000']
```

- Thuật toán quy hoạch động sẽ giải quyết bài toán với độ phức tạp là $O(n \times \sum weight[i])$
 - Lấy testcase lớn nhất dựa trên cận trên là mỗi item có $weight[i] = 1000$ có 1000 item \rightarrow Độ phức tạp là 10^9
- Cũng tương tự với ORTools nên việc lấy lớn hơn nữa sẽ không mang nhiều ý nghĩa mà còn làm hư máy 😊
- Em quyết định để set_time cho optimal answer bằng qui hoạch động là unlimited và cho ORTools là 120s
- Để tránh hư máy, đối với qui hoạch động nếu complexity của test case nào $n \times \sum weight[i] > 10^9$ thì thuật toán sẽ không thực thi, những case đó sẽ xuất ra giá trị -1 .
- Các phép toán được xử lí bởi CPU, cấu hình CPU:

Device name LAPTOP-R449ACEN
Processor 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz
 2.30 GHz
Installed RAM 16.0 GB (15.8 GB usable)

- Bảng kết quả được thống kê như sau:

	Case	Total Value Optimal	Total Weight Optimal	Runtime Optimal	Total Value gOrt	Total Weight gOrt	Runtime gOrt
0	00Uncorrelated_n00050_R01000_s005.kp	19780	11921	0.290603876	19780	11921	0
1	00Uncorrelated_n00100_R01000_s005.kp	42224	28455	1.47677803	42224	28455	0.000997543
2	00Uncorrelated_n00200_R01000_s005.kp	86024	47601	5.330918312	86024	47601	0.000999928
3	00Uncorrelated_n00500_R01000_s005.kp	202400	120137	33.89817786	202400	120129	0.002002954
4	00Uncorrelated_n01000_R01000_s005.kp	403106	243748	137.4539495	403106	243745	0.006999731
5	01WeaklyCorrelated_n00050_R01000_s005.kp	14125	12715	0.310998678	14125	12715	0.001000404
6	01WeaklyCorrelated_n00100_R01000_s005.kp	27933	24642	1.298731565	27933	24637	0.000999689
7	01WeaklyCorrelated_n00200_R01000_s005.kp	57753	53102	5.656203508	57753	53102	0.001000404
8	01WeaklyCorrelated_n00500_R01000_s005.kp	135595	123199	34.00370932	135595	123199	0.002999306
9	01WeaklyCorrelated_n01000_R01000_s005.kp	268214	243336	138.2448173	268214	243336	0.005000114
10	02StronglyCorrelated_n00050_R01000_s005.kp	16219	12719	0.315999508	16219	12719	0.057000637
11	02StronglyCorrelated_n00100_R01000_s005.kp	31842	24642	1.308218241	31842	24642	0.000999689
12	02StronglyCorrelated_n00200_R01000_s005.kp	66904	53104	5.704015732	66904	53104	0.079999208
13	02StronglyCorrelated_n00500_R01000_s005.kp	158399	123199	33.99786091	158161	122961	124.893677
14	02StronglyCorrelated_n01000_R01000_s005.kp	314136	243336	140.4536872	313948	243148	124.3336353
15	03InverseStronglyCorrelated_n00050_R01000_s005.kp	13594	15194	0.369002581	13594	15194	0.195999861
16	03InverseStronglyCorrelated_n00100_R01000_s005.kp	26592	29592	1.544261932	26592	29592	0.003000259
17	03InverseStronglyCorrelated_n00200_R01000_s005.kp	56477	62977	6.746890545	56477	62977	30.86735225
18	03InverseStronglyCorrelated_n00500_R01000_s005.kp	132251	147951	41.51561785	132251	147951	55.92194915
19	03InverseStronglyCorrelated_n01000_R01000_s005.kp	261648	292748	171.2726612	261648	292748	124.6291571
20	04AlmostStronglyCorrelated_n00050_R01000_s005.kp	16204	12719	0.313998938	16204	12719	0.027999878
21	04AlmostStronglyCorrelated_n00100_R01000_s005.kp	31891	24641	1.306042671	31891	24641	0.001000166
22	04AlmostStronglyCorrelated_n00200_R01000_s005.kp	66890	53104	5.691141367	66890	53104	0.004002333

23	04AlmostStronglyCorrelated_n00500_R01000_s005.kp	158403	123199	33.9225564	158273	123083	125.7115219
24	04AlmostStronglyCorrelated_n01000_R01000_s005.kp	314165	243336	139.6344206	314115	243294	125.8311307
25	05SubsetSum_n00050_R01000_s005.kp	12719	12719	0.3089993	12719	12719	0.001001835
26	05SubsetSum_n00100_R01000_s005.kp	24642	24642	1.279164314	24642	24642	0.001000881
27	05SubsetSum_n00200_R01000_s005.kp	53104	53104	5.581935167	53104	53104	0
28	05SubsetSum_n00500_R01000_s005.kp	123199	123199	33.84814215	123199	123199	0.000999451
29	05SubsetSum_n01000_R01000_s005.kp	243336	243336	137.7452257	243336	243336	0.00599885
30	06UncorrelatedWithSimilarWeights_n00050_R01000_s005.kp	19503	2401442	36.0966835	19503	2401442	0.011000872
31	06UncorrelatedWithSimilarWeights_n00100_R01000_s005.kp	39025	4902512	145.6688795	39025	4902512	0.006999016
32	06UncorrelatedWithSimilarWeights_n00200_R01000_s005.kp	-1	-1	0	79002	9904796	0
33	06UncorrelatedWithSimilarWeights_n00500_R01000_s005.kp	-1	-1	0.001000404	186275	24711907	124.4225686
34	06UncorrelatedWithSimilarWeights_n01000_R01000_s005.kp	-1	-1	0.000999451	369485	49523855	0.770149469
35	07SpannerUncorrelated_n00050_R01000_s005.kp	7728	9808	0.194999218	7728	9808	5.639832973
36	07SpannerUncorrelated_n00100_R01000_s005.kp	16595	21061	0.953906536	16562	21022	155.4099302
37	07SpannerUncorrelated_n00200_R01000_s005.kp	32977	41847	4.139671803	32977	41847	154.3141446
38	07SpannerUncorrelated_n00500_R01000_s005.kp	82077	104139	27.3832469	82075	104125	144.4971855
39	07SpannerUncorrelated_n01000_R01000_s005.kp	166206	210866	118.7411809	166173	210827	135.2134476
40	08SpannerWeaklyCorrelated_n00050_R01000_s005.kp	18104	7683	0.160999537	18104	7683	4.935055971
41	08SpannerWeaklyCorrelated_n00100_R01000_s005.kp	38960	16527	0.763990879	38960	16527	148.3652027
42	08SpannerWeaklyCorrelated_n00200_R01000_s005.kp	77288	32786	3.292628527	77262	32763	142.9110072
43	08SpannerWeaklyCorrelated_n00500_R01000_s005.kp	192378	81584	21.51426339	192286	81539	140.0566721
44	08SpannerWeaklyCorrelated_n01000_R01000_s005.kp	389529	165189	91.12802958	389470	165155	134.452503
45	09SpannerStronglyCorrelated_n00050_R01000_s005.kp	21313	7813	0.165829897	21313	7813	0.165060282
46	09SpannerStronglyCorrelated_n00100_R01000_s005.kp	44492	16592	0.770759821	44492	16592	145.4200485
47	09SpannerStronglyCorrelated_n00200_R01000_s005.kp	90294	32794	3.274526119	90261	32761	139.5853996
48	09SpannerStronglyCorrelated_n00500_R01000_s005.kp	225483	81683	21.80767322	225472	81672	137.7749684
49	09SpannerStronglyCorrelated_n01000_R01000_s005.kp	457030	165430	91.85206866	457030	165430	129.6650472
50	10MultipleStronglyCorrelated_n00050_R01000_s005.kp	20813	12713	0.315000057	20813	12713	0.008997202
51	10MultipleStronglyCorrelated_n00100_R01000_s005.kp	40642	24642	1.331157207	40642	24642	0.035003901
52	10MultipleStronglyCorrelated_n00200_R01000_s005.kp	84200	53100	5.693000793	84200	53100	126.4830883
53	10MultipleStronglyCorrelated_n00500_R01000_s005.kp	201698	123198	33.9687829	201698	123198	130.8612614
54	10MultipleStronglyCorrelated_n01000_R01000_s005.kp	399336	243336	141.2285001	399336	243336	0.883996487
55	11ProfitCeiling_n00050_R01000_s005.kp	12708	12719	0.316455126	12708	12718	0.114911795
56	11ProfitCeiling_n00100_R01000_s005.kp	24627	24642	1.320744991	24627	24640	5.478987455
57	11ProfitCeiling_n00200_R01000_s005.kp	53082	53104	5.741071939	53082	53103	139.2547722
58	11ProfitCeiling_n00500_R01000_s005.kp	123153	123199	34.2398839	123153	123199	135.4454768
59	11ProfitCeiling_n01000_R01000_s005.kp	243240	243336	142.1887715	243240	243334	130.0816047
60	12Circle_n00050_R01000_s005.kp	268001	12719	0.318999052	268001	12719	0.169995308
61	12Circle_n00100_R01000_s005.kp	519231	24642	1.318853855	519231	24642	0.165315866
62	12Circle_n00200_R01000_s005.kp	1118954	53104	5.709453821	1118954	53104	144.6535809
63	12Circle_n00500_R01000_s005.kp	2595920	123199	34.51422286	2595920	123199	6.320014
64	12Circle_n01000_R01000_s005.kp	5127320	243336	145.993504	5127320	243336	145.993504

- Những row được tô màu xanh là trình GoogleORTools chạy cho kết quả chưa được tối ưu trong khoảng thời gian là 120s
- Đề ý thấy những testcases có $n \geq 200$ sẽ giải lâu hơn cả kĩ thuật quy hoạch động lẫn GoogleORTools
- Những testcase mà kĩ thuật quy hoạch động không giải được do cần độ phức tạp lớn thì ta sẽ không kiểm chứng được optimality certificate