

Solving Knapsack Problems Using Google OR-Tools

Le Thi Lien

Student ID : 21522282

(April 20, 2023)

1. Introduction

The **Knapsack Problem** involves selecting a group of items, each with a specified value and size (weight or volume), to be packed in a container with a limited capacity. If the total size of the selected items exceeds the capacity, not all of them can be packed. Thus, the main objective of the problem is to choose a subset of items that maximizes their total value while fitting within the container's capacity.

In this report, we demonstrate how to solve the knapsack problem using OR-Tools in the following sections.

2. Experimental Setup

2.1. Test Case Selection

As it is not feasible to run through all the test cases, I created a function called *generate data* to randomly select 8 test cases from each group of test cases in the given data set.

- Declare a list of the names of the test case groups and the names of the corresponding subdirectories in the original data set.
- Create a complete name string for each test case.
- Randomly generate a number in the range $[0; 1]$. If this number is greater than or equal to 0.5, the test case selected will belong to the R01000 folder, otherwise it will belong to the R10000 folder.
- Finally, a random integer is generated within the range $[10; 99]$, and this number is used to create the name of the selected test case.

2.2. OR-Tools

To solve the Knapsack problem using the Google OR-Tools, I perform the following steps:

- Read input data from a file.
- Initialize a solver from OR-Tools.
- Set a time limit for solving the problem (180 seconds).
- Initialize the input parameters for the knapsack problem.
- Solve the problem using the solver.
- Retrieve the results from the solver and display information such as the total weight, total value, number of selected items, list of selected items, and their weights.
- Write the results to an output file.

2.3. Result storage

The output includes the name of each test case, the maximum capacity, the total value of the solution, the total weight of the items in the solution, the number of selected items, and the list of selected items, which are saved to an output file and then used to generate a statistical table.

3. Experimental Results

With a total of 104 test cases, and a maximum processing time of 3 minutes for each test case, we have the statistics table below. From this table, we can see that OR-Tools found optimal solutions for many test cases, however, there are also cases where this tool cannot find a solution within the given time limit.

For each test case group, I selected all test cases from all input data sizes ranging from 50 to 10000.

I clearly observed that when the input data size is small, from 50 to 500, OR-Tools quickly finds solutions and provides many optimal solutions.

As the size increases to around 1000 to 2000, the time to find solutions becomes significantly slower, and the solutions provided are less optimal.

And when the input data size becomes very large, such as 5000 or 10000, we can see that OR-Tools does not provide any result within the 3-minute time limit. I extended the time for these large-sized test cases to be able to evaluate their solutions, and it was not disappointing. OR-Tools indeed provides some optimal solutions, even for such large input data sizes.

No.	Test cases	N	Capacities	Total weights	Total values
1	00/n00050/R10000/s018.kp	50	123760	123255	190317
2	00/n00100/R01000/s085.kp	100	23356	23336	41455
3	00/n00200/R10000/s023.kp	200	526186	525855	791932
4	00/n00500/R10000/s018.kp	500	1244445	1244410	2042604
5	00/n01000/R01000/s052.kp	1000	248370	248367	407099
6	00/n02000/R10000/s024.kp	2000	5080540	5080527	8093045
7	00/n05000/R01000/s026.kp	5000	1232650	1232650	2026397
8	00/n10000/R01000/s011.kp	10000	2467553	2467552	4034773

No.	Test cases	N	Capacities	Total weights	Total values
9	01/n00050/R10000/s021.kp	50	120575	120425	135221
10	01/n00100/R01000/s077.kp	100	24216	24216	27224
11	01/n00200/R01000/s076.kp	200	49263	49262	54907
12	01/n00500/R01000/s076.kp	500	125283	125283	138553
13	01/n01000/R01000/s093.kp	1000	251958	251958	278763
14	01/n02000/R01000/s040.kp	2000	491858	491858	541948
15	01/n05000/R10000/s039.kp	5000	12263784	12263783	13577369
16	01/n10000/R10000/s083.kp	10000	25018917	25018917	27624291

No.	Test cases	N	Capacities	Total weights	Total values
17	02/n00050/R10000/s092.kp	50	138210	138210	171210
18	02/n00100/R01000/s012.kp	100	22786	22786	29986
19	02/n00200/R01000/s083.kp	200	52030	52030	65830
20	02/n00500/R10000/s028.kp	500	1209952	1208196	1565196
21	02/n01000/R10000/s056.kp	1000	2481346	2481346	3183346
22	02/n02000/R10000/s094.kp	2000	5035904	5030779	6430779
23	02/n05000/R01000/s083.kp	5000	1255530	1254997	1605597
24	02/n10000/R10000/s085.kp	10000	24806058	24805192	31851192

4. Conclusion

In conclusion, our analysis shows that Google OR-Tools is a powerful tool for solving Knapsack problems. It has proven to be effective in solving most of the test cases with high accuracy and efficiency. However, there are also limitations to this tool, as it struggles with certain types of data sets.

Overall, Google OR-Tools is a valuable tool for solving Knapsack problems, but it may not always be the best choice depending on the specific context of the problem. It is important to carefully consider the characteristics of the data set and the problem requirements before choosing a particular method for Knapsack problem-solving.

No.	Test cases	N	Capacities	Total weights	Total values
25	03/n00050/R10000/s099.kp	50	152541	152541	136541
26	03/n00100/R10000/s081.kp	100	306185	306185	274185
27	03/n00200/R10000/s072.kp	200	597035	597035	532035
28	03/n00500/R10000/s039.kp	500	1430936	1430007	1278007
29	03/n01000/R01000/s066.kp	1000	297854	297339	265639
30	03/n02000/R01000/s019.kp	2000	588598	588492	525592
31	03/n05000/R01000/s060.kp	5000	1477951	1477455	1320255
32	03/n10000/R10000/s074.kp	10000	29878825	29876465	26701465

No.	Test cases	N	Capacities	Total weights	Total values
33	04/n00050/R01000/s084.kp	50	11705	11705	15212
34	04/n00100/R01000/s095.kp	100	25040	25040	31845
35	04/n00200/R01000/s087.kp	200	46331	46331	60646
36	04/n00500/R01000/s042.kp	500	124243	124009	159269
37	04/n01000/R10000/s023.kp	1000	2484308	2482557	3183506
38	04/n02000/R01000/s042.kp	2000	501242	500820	640533
39	04/n05000/R10000/s044.kp	5000	12285041	12285041	15811974
40	04/n10000/R01000/s031.kp	10000	2503037	2502820	3205028

No.	Test cases	N	Capacities	Total weights	Total values
41	05/n00050/R01000/s051.kp	50	14361	14361	14361
42	05/n00100/R10000/s052.kp	100	245103	245103	245103
43	05/n00200/R01000/s015.kp	200	51129	51129	51129
44	05/n00500/R01000/s089.kp	500	125430	125430	125430
45	05/n01000/R01000/s093.kp	1000	251958	251958	251958
46	05/n02000/R01000/s061.kp	2000	488478	488478	488478
47	05/n05000/R10000/s027.kp	5000	12590050	12590050	12590050
48	05/n10000/R10000/s046.kp	10000	24770168	24770168	24770168

No.	Test cases	N	Capacities	Total weights	Total values
49	06/n00050/R10000/s018.kp	50	2476484	2401332	17242
50	06/n00100/R10000/s026.kp	100	4952818	4902130	37730
51	06/n00200/R01000/s036.kp	200	9906099	9905423	74087
52	06/n00500/R01000/s050.kp	500	24765068	24712081	184670
53	06/n01000/R10000/s019.kp	1000	49529989	49525065	358910
54	06/n02000/R10000/s060.kp	2000	99058668	99049404	744536
55	06/n05000/R01000/s014.kp	5000	247650106	247625375	1866210
56	06/n10000/R01000/s083.kp	10000	495297280	495246785	3755778

No.	Test cases	N	Capacities	Total weights	Total values
57	07/n00050/R10000/s037.kp	50	92961	92565	102300
58	07/n00100/R10000/s063.kp	100	179674	179570	98757
59	07/n00200/R10000/s068.kp	200	386105	386100	400140
60	07/n00500/R10000/s086.kp	500	650135	649894	614026
61	07/n01000/R01000/s085.kp	1000	128604	128582	118276
62	07/n02000/R10000/s073.kp	2000	2455169	2454738	2553252
63	07/n05000/R01000/s030.kp	5000	398684	398656	852612
64	07/n10000/R10000/s090.kp	10000	15487765	15487744	4783360

No.	Test cases	N	Capacities	Total weights	Total values
65	08/n00050/R01000/s082.kp	50	3940	3929	14714
66	08/n00100/R10000/s036.kp	100	161110	160834	500926
67	08/n00200/R10000/s041.kp	200	147594	147284	688804
68	08/n00500/R10000/s084.kp	500	1053945	1053655	89361
69	08/n01000/R10000/s042.kp	1000	794780	794381	1213643
70	08/n02000/R10000/s082.kp	2000	1583448	1583399	5849388
71	08/n05000/R10000/s099.kp	5000	3731612	3731334	4839289
72	08/n10000/R10000/s067.kp	10000	10296732	10296064	37439303

No.	Test cases	N	Capacities	Total weights	Total values
73	09/n00050/R10000/s090.kp	50	21383	21364	208364
74	09/n00100/R10000/s096.kp	100	151142	151032	519032
75	09/n00200/R01000/s031.kp	200	2884	2877	76377
76	09/n00500/R01000/s079.kp	500	29713	29704	217904
77	09/n01000/R01000/s014.kp	1000	94179	94167	490067
78	09/n02000/R01000/s054.kp	2000	257202	257186	1045486
79	09/n05000/R10000/s081.kp	5000	5974374	5974252	20375252
80	09/n10000/R10000/s038.kp	10000	13172872	13172607	44819607

No.	Test cases	N	Capacities	Total weights	Total values
81	10/n00050/R10000/s063.kp	50	127710	127684	202684
82	10/n00100/R01000/s028.kp	100	22374	22340	38140
83	10/n00200/R10000/s077.kp	200	503670	503670	816670
84	10/n00500/R01000/s033.kp	500	123558	123558	200658
85	10/n01000/R10000/s038.kp	1000	2528952	2528952	4078952
86	10/n02000/R10000/s049.kp	2000	4822890	4822890	7954890
87	10/n05000/R01000/s096.kp	5000	1244484	1244484	2012184
88	10/n10000/R10000/s078.kp	10000	25046664	25046664	40460664

No.	Test cases	N	Capacities	Total weights	Total values
89	11/n00050/R10000/s065.kp	50	127589	127589	127581
90	11/n00100/R10000/s075.kp	100	257206	257205	257193
91	11/n00200/R10000/s029.kp	200	497233	497233	497211
92	11/n00500/R10000/s089.kp	500	1253205	1253204	1253175
93	11/n01000/R01000/s031.kp	1000	249776	249775	249669
94	11/n02000/R10000/s048.kp	2000	5098116	5098116	5097909
95	11/n05000/R01000/s034.kp	5000	1232334	1232332	1231875
96	11/n10000/R01000/s071.kp	10000	2492769	2492768	2491812

No.	Test cases	N	Capacities	Total weights	Total values
97	12/n00050/R01000/s053.kp	50	11979	11979	252409
98	12/n00100/R01000/s027.kp	100	24793	24793	522410
99	12/n00200/R01000/s041.kp	200	51205	51205	1078934
100	12/n00500/R01000/s068.kp	500	124312	124312	2619369
101	12/n01000/R01000/s066.kp	1000	248350	248350	5232968
102	12/n02000/R10000/s025.kp	2000	4995636	4995636	333025566
103	12/n05000/R01000/s083.kp	5000	1255530	1255530	26455196
104	12/n10000/R01000/s065.kp	10000	2503746	2503746	52756286
