Solving Multidimensional Knapsack Problem

Shih-Fen Cheng
School of Information Systems
Singapore Management University
sfcheng@smu.edu.sg

2017-7-21 22:58

1 Background

In this mini project, we will look at the knapsack problem, one of the most important and well studied "hard optimization problems". Defined very generally, knapsack problem deals with the selection of items from an available pool, and the goal is to achieve highest possible value while observing the capacity limit.

The mathematical formulation of a general multi-dimensional knapsack problem is as follows:

$$\begin{aligned}
\max \quad & v_1x_1 + v_2x_2 + \ldots + v_nx_n \qquad (1)\\
s.t. \quad & \\
& w_{11}x_1 + w_{12}x_2 + \ldots + w_{1n}x_n \leq b_1 \\
& w_{21}x_1 + w_{22}x_2 + \ldots + w_{2n}x_n \leq b_2 \\
& \ldots \\
& w_{m1}x_1 + w_{m2}x_2 + \ldots + w_{mn}x_n \leq b_m \\
& x_i \in \{0, 1\}.
\end{aligned}$$

In the above formulation, n stands for the number of tasks, m stands for the number of resources, v_i stands for the value one could obtain in completing task i, b_j stands for the capacity limit on resource j, w_{ji} stands for the consumption of resource j for task i. x_i is a binary variable, indicating whether to include task i or not.

For simplicity, we assume that all problem constants (v_i, w_{ij}, b_i) are integers.

2 Programming Assignment

The goal of this mini project is to let you be familiar with the various searching techniques one can use to solve difficult problems. Also, you will get some feeling about the trade offs involved in these problems.

2.1 General Information

Besides the working programs, we will always expect:

- A specification on how to run your programs with the test data.

- A description of the strategies and implementation techniques used.
- A description on the experimental results obtained with the algorithms for the various benchmarks.

2.2 Input Data Format

Your program should be designed to take a problem data file as input, with the following format:

```
n
m
v1 w11 w21 ... wm1
v2 w12 w22 ... wm2
...
vn w1n w2n ... wmn
b1 b2 ... bm
```

The output of the program should be the value of x_i's, one line for each x_i. A random instance generator is available at: `http://www.mysmu.edu/faculty/sfcheng/projects/mkp/`. The random instance generator is constructed following Bertsimas and Demir (2002), and the usage is as follows:

```
usage: genPref2 <M> <N> <A> <C> <mode> <tightness> <tag>
```

`M` and `N` are numbers of resources and tasks respectively. `A` and `C` stand for the upper bound for the resource requirement and task value respectively. `mode` is either 0 or 1, indicating whether the resource requirements and capacities are correlated or not (correlated problems will be much harder). `tightness` is a real number less than 1, indicating how tight the knapsack constraints are, i.e., $b_j = \texttt{tightness} \cdot \sum_{i=1}^{N} w_{ji}$. For example, if you would like to generate a single dimensional knapsack problem with 10 tasks, you can use the following command:

```
genPref2 1 10 20 50 1 0.5 test1
```

The problem data generated will be stored in `test1.pref`.

2.3 Dynamic Programming

Task: Formulate a dynamic programming model for the multi-dimensional knapsack problem described in (1). Study its empirical performance on randomly generated benchmarks.

2.4 Local Search

Task: Design a local-search algorithm and implement it for the multi-dimensional knapsack problem described in (1). Study its empirical performance (for both running time and solution quality) on randomly generated benchmarks against DP. For ideas, you can reference the genetic algorithm approach by Chu and Beasley (1998) or the tabu search approach by Hanafi and Freville (1998).

References

Dimitris Bertsimas and Ramazan Demir. An approximate dynamic programming approach to multidimensional knapsack problems. *Management Science*, 48(4):550–565, 2002.

P. C. Chu and J. E. Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4(1):63–86, 1998. doi: 10.1023/A:1009642405419.

Said Hanafi and Arnaud Freville. An efficient tabu search approach for the 0-1 multidimensional knapsack problem. *European Journal of Operational Research*, 106(2-3):659–675, April 1998. doi: 10.1016/S0377-2217(97)00296-8.