Mathematical Challenge July 2018

Parametric knapsack problems

References

- [1] H. Kellerer, U. Pferschy, D. Pisinger, Knapsack problems (2004).
- [2] P. Carstensen, Complexity of some parametric integer and network programming problems, Mathematical Programming 26 (1) (1983) 64–75.

Description

One of the most fundamental problems in combinatorial optimization is the so-called *knapsack problem*, which can be described as follows: Suppose we want to go to a hiking trip and want to fill our knapsack. We can choose between n different items, denoted by 1 to n. Each item has a positive profit $p_i \in \mathbb{N}$ that describes the utility the respective item has if we take it with us, but also a positive weight $w_i \in \mathbb{N}$. We cannot lift more than a given value W, so we must wisely decide which items to pack and which ones to leave at home. In more mathematical terms, we can formulate the standard knapsack problem as an optimization problem in the following way:

$$\max \sum_{i=1}^{n} p_i \cdot x_i \quad \text{s.t.} \quad \sum_{i=1}^{n} w_i \cdot x_i \leq W \quad \text{and} \quad x_i \in \{0,1\} \text{ for all } i \in \{1, \dots, n\}.$$

In other words: We want to find a solution vector x with binary entries (= our choice of items) that maximizes the profit function while fulfilling the capacity constraint of the knapsack [1].

The knapsack problem is known to be (weakly) *NP-hard to solve*, which means that it is highly unlikely to ever come up with an algorithm with a polynomial running time. Based on this fact, the problem was (and still is) extensively studied in the field of *approximation algorithms*. In fact, it turns out that approximate solutions with provably good quality can be computed very easily. One of the most famous such algorithms can be described as follows:

1. Sort the items $i \in \{1, \dots, n\}$ by the ratio $\frac{p_i}{w_i}$ in decreasing order.
2. Pick as many items as possible in this order until all items are in the knapsack or the next item j' would exceed the knapsack's capacity. Call the resulting solution x.
3. Either return this solution x or the solution x' consisting of item j' only, depending on what yields the higher profit.

The proof of the following question can be found in many textbooks (e.g. [1]). We still encourage you to come up with a solution on your own:

- Q1: Show that the above algorithm always computes a *½-approximation* for the problem, i.e., a solution whose profit is at least half as high as the maximum achievable profit (even if we don't know the optimal solution).

Recently, research on knapsack problems was extended to a more powerful generalization, called the *parametric knapsack problem*: We introduce a parameter t and generalize the profit of each item from a scalar value to an affine function $p_i(t) = a_i + t \cdot b_i$. The aim is to find an optimal solution for the knapsack problem for *all values of* $t \in \mathbb{R}$. We can think of t as an unknown parameter such as time or the value of an option's underlying. The problem is then to get a full picture of the maximum achievable profit as a function of t, no matter what t eventually evaluates to.

Early research has shown that the parametric knapsack problem is extremely hard to solve since we might already need an exponential number of knapsack solution only to describe the optimal solution in the parametric setting, yielding an exponential worst-case running time for every exact algorithm [2].

Like in the case of the standard knapsack problem, research recently focused on approximation algorithms for the parametric knapsack problem. An obvious idea is to generalize the ½-approximation solution described above to the parametric setting.

- Q2: Suppose we execute the ½-approximation algorithm for the standard knapsack problem for all possible values of $t \in \mathbb{R}$. Show that there will be less than n^2 different solutions. Can you come up with an exact number? Can you turn this idea into an approximation algorithm?
- Q3: Is the function mapping the parameter t to the profit of its approximate solution a continuous function? If yes, explain why. If no, can we make it continuous without degrading the solution quality?

In the following, we consider another extension of the knapsack problem, in which we add a *cardinality constraint* of the form $\sum_{i=1}^{n} x_i \leq k$. In addition to the capacity constraint, we restrict the number of items in the knapsack to *k*. This problem is known as the *cardinality constrained knapsack problem* [1].

- Q4: If we stick to constant profits p_i (setting *t=0*) and relax the constraints $x_i \in \{0,1\}$ to $x_i \in [0,1]$, show that there is always an optimal solution x* which has two or less *fractional values* $x_i \in (0,1)$. Can we use this fact to get a ½-approximation algorithm for the cardinality constrained knapsack problem?
- Q5 (Bonus): Is the bound given in Q2 still valid if we enter the parametric setting? Can we get an approximation algorithm for the parametric cardinality constrained knapsack problem?

We look forward to your opinions and insights.

Best Regards,

swissQuant Group Leadership Team