Final Project Proposal

Knapsack Problem

Jingyuan Wang(jw3732), Shaohua Tang(st3207)

Fall 2019

Description

Knapsack Problem is one of the most famous NP problems in the mathematics areas. It is a problem in the area of combinatorial optimization. Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.[1]

And here we would focus on the bounded knapsack problem(BKP), which doesn't require the number of items to be only one or zero, but rather gives a non-negative integer value c to represent the maximum copies of each item.

Specifically, Given a set of n kinds of items numbered from 1 up to n, each with a weight w_i, a value v_i and number of copies x_i, along with a maximum weight capacity W and a non-negative integer value c, the objective function is given by [1]

$$\text{maximize } \sum_{i=1}^{n} v_i x_i$$
$$\text{subject to } \sum_{i=1}^{n} w_i x_i \leq W \text{ and } 0 \leq x_i \leq c$$

So far, for this problem, there exists no known polynomial algorithm, while only the decision part of can be computed within polynomial runtime. Currently, the most popular mechanism to solve this problem is to use DP(dynamic programming).

Our goal is to find best possible solution using Haskell with parallel programming. Additionally, we would also like to show the difference of runtime between using parallel functional programming and without.

References

[1] Knapsack problem. https://en.wikipedia.org/wiki/Knapsack_problem. Accessed: 2019-11-21.