The Knapsack Problem

A Survey of Solution Approaches

Sayak Biswas
UNIVERSITY *of* **FLORIDA**
UFID: 54584911

April 12, 2016

Abstract

This paper surveys existing literature for different approaches to solve the knapsack problem. The Knapsack Problem is a combinatorial optimization problem in which one has to maximize the profits gained by packing a set of objects in a knapsack without exceeding its capacity. The problem is ***NP***-complete, thus there is no known polynomial time algorithm for a large input.

Specifically, we take a look at the *0-1 Knapsack Problem* and provide a qualitative comparison between the three well-known approaches towards solving the problem: brute-force, dynamic programming and branch & bound algorithms.

1 Introduction

The *0-1 Knapsack Problem* is an optimization problem, which at a high level is to choose the most profitable subset from a collection of available items without overloading the knapsack. The problem is formally defined as follows:

Given a knapsack of maximum capacity C and n items each weighing w_i and with an associated profit of p_i, the *Knapsack Problem* is to choose a subset of the items such that the below holds true:

maximize $\sum\limits_{i=1}^{n} p_i x_i$

on the condition $\sum\limits_{i=1}^{n} w_i x_i \leq C, x_i \in \{0, 1\}, i = 1, ..., n$

The two most common variations are the *Unbounded* and *Bounded* Knapsack Problems.