On the Complexity of the Knapsack Problem with Some Graph Theoretic Constraints

Thesis submitted to
Indian Institute of Technology Kharagpur
for the award of the degree

of

Doctor of Philosophy

by

SIPRA SINGH
Roll No.: 19CS92R02

under the supervision of

Prof. Palash Dey
(Department of Computer Science and Engineering)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
KHARAGPUR-721302, WEST BENGAL, INDIA
April 2026

Abstract

The Knapsack Problem is a well-known problem in Computer Science. In this problem, n items are given as inputs, each with a specific weight and profit value. The goal of this problem is to select some items for a bag with a given capacity to maximize the total profit, ensuring the total weight of the items does not exceed the bag's capacity. In this thesis, we study the Knapsack Problem with graph-theoretic constraints. That is, there exists a graph structure on the input set of items of the Knapsack Problem that maps the knapsack items to the vertices of the graph, and the solution also needs to satisfy specific graph theoretic constraints on top of the knapsack constraint. Here, the considered graph-theoretic constraints are Connectedness, Path, Shortest Path, Vertex Cover, and Dominating Set.

We denote the input graph structure as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with n vertices and m edges. Each vertex u has weight $(w(u))_{u \in \mathcal{V}}$ and profit $(\alpha(u))_{u \in \mathcal{V}}$. The size of the knapsack is s, and the target profit value is d. We aim to compute if there exists a set $\mathcal{U} \subseteq \mathcal{V}$ with total weight at most s, and total profit amount at least d, such that any one of the above-mentioned graph constraints holds simultaneously. We obtained the following interesting complexity results.

- **The Knapsack problem with the Connectedness constraint** (Connected Knapsack):
  Here, we aim to find a connected subset of items that gives maximum profit and satisfies the size constraint of the knapsack. We showed that this variant is strongly NP-complete even for graphs of maximum degree four and weakly NP-complete for stars. Thus, it is unlikely to get a polynomial-time

algorithm for this problem. We developed a pseudo-polynomial time algorithm parameterized by treewidth (tw). The complexity of this algorithm is $\mathcal{O}\left(2^{tw \log tw} \cdot \mathrm{poly}(n) \cdot \min\{s^2, d^2\}\right)$. However, by considering vcs (size of the minimum Vertex Cover of the subgraph induced by the solution) as the parameter instead of tw, we proved that there is no *pseudo-Fixed Parameter Tractable (pseudo-*FPT*)* algorithm with running time $\mathcal{O}(f(vcs) \cdot \mathrm{poly}(n, s, d))$ unless *Exponential Time Hypothesis* (ETH) fails. Then we exhibit a $(1-\varepsilon)$ factor FPT-approximation algorithm with running in time $\mathcal{O}\left(2^{tw \log tw} \cdot \mathrm{poly}(n, 1/\varepsilon)\right)$ for every $\varepsilon > 0$.

☐ **The Knapsack problem with Path as the graph-constraint** (Path Knapsack):
For this variant, we showed that it is strongly NP-complete even for graphs of maximum degree three and NP-complete even if the graph has pathwidth at most two, but polynomial time solvable for trees. Like Connected Knapsack there exist pseudo-FPT and $(1-\varepsilon)$ factor FPT-approximation algorithms parameterized by tw with complexities $\mathcal{O}\left(2^{tw \log tw} \cdot \mathrm{poly}(n) \cdot \min\{s^2, d^2\}\right)$ and $\mathcal{O}\left(2^{tw \log tw} \cdot \mathrm{poly}(n, 1/\varepsilon)\right)$, respectively. We developed another FPT algorithm for Path Knapsack parameterized by solution size k runs in time $(2e)^k \cdot k^{\mathcal{O}(\log k)} \cdot n^{\mathcal{O}(1)}$. As for Path Knapsack, $vcs = \frac{k}{2}$, we conclude that the complexity of the FPT algorithm is $\mathcal{O}((2e)^{2vcs} \cdot vcs^{\mathcal{O}(\log vcs)} \cdot n^{\mathcal{O}(1)})$ parameterized by vsc.

☐ **The Knapsack problem with Shortest-path as the graph-constraint** (Shortest Path Knapsack):
It is the variant where the solution must induce a shortest path. We obtained similar results to Path Knapsack; it is NP-complete even if the graph has pathwidth at most two, but it is polynomial-time solvable for trees. We developed two pseudo-polynomial time algorithms for Shortest Path Knapsack with complexities $\mathcal{O}((m + n \log n) \cdot \min\{s^2, \alpha(\mathcal{V})^2\})$ for non-negative edge weights, and $\mathcal{O}(mn \cdot \min\{s^2, \alpha(\mathcal{V})^2\})$ for negative edge weights using Dynamic-programming approach. This variant admits *Fully Polynomial Time Approximation Scheme* (FPTAS) with approximation factor $(1-\varepsilon)$, $\forall \varepsilon > 0$.

Here, $\alpha(\mathcal{V})$ is denoted as $\sum_{v \in \mathcal{V}} \alpha(v)$.

- ❑ **The Knapsack problem with Vertex Cover as the constraint** (Vertex Cover Knapsack):
  We proved strongly NP-completeness results for Vertex Cover Knapsack, Vertex Cover Knapsack with Budget (Vertex Cover Knapsack with fixed size), and Minimal Vertex Cover Knapsack (Knapsack with minimal size Vertex Cover). However, Vertex Cover Knapsack and Vertex Cover Knapsack with Budget admit pseudo-polynomial time algorithms with complexities $\mathcal{O}(n \cdot min\{s^2, (\alpha(\mathcal{V}))^2\})$ and $\mathcal{O}(n^2 \cdot min\{s^2, (\alpha(\mathcal{V}))^2\})$ for trees, respectively, whereas Minimum Vertex Cover Knapsack (Knapsack with minimum size Vertex Cover) is not known to be in class NP (*Non-deterministic Polynomial*), even the decision version of this variant of Vertex Cover Knapsack for the general graph.

- ❑ **The Knapsack problem with Dominating as the constraint** (Dominating Set Knapsack):
  We defined our problem Dominating Set Knapsack by attaching Knapsack Problem with Dominating Set on graphs. We showed that the Dominating Set Knapsack problem is strongly NP-complete even when restricted to Bipartite graphs, but weakly NP-complete for Star graphs. We presented a pseudo-polynomial time algorithm for Trees in time $\mathcal{O}(n \cdot min\{s^2, (\alpha(\mathcal{V}))^2\})$. We showed that Dominating Set Knapsack is unlikely to be FPT by proving that it is $W[2]$-hard parameterized by the solution size. We developed pseudo-FPT algorithms with running times $\mathcal{O}(4^{tw} \cdot n^{\mathcal{O}(1)} \cdot \min\{s^2, (\alpha(\mathcal{V}))^2\})$ and $\mathcal{O}(2^{vck-1} \cdot n^{\mathcal{O}(1)} \cdot \min\{s^2, (\alpha(\mathcal{V}))^2\})$, where tw represents the Treewidth of the given graph, vck is the solution size of the Vertex Cover Knapsack. We obtained similar results for other variants, such as k-Dominating Set Knapsack and Minimal Dominating Set Knapsack.

Keywords: Knapsack Problem, Vertex Cover, Dominating Set, Graph Algorithms, Parameterized Complexity, Approximation Algorithms.