13 Finding the best combination within budget

～Combinatorial optimization using binary decision diagram～

Abstract

Knapsack problem (KP) is one of the fundamental combinatorial optimization problems, and covers a wide range of practical problems such as text summarization, bin packing, and network design. The general solution of KP cannot simply handle additional constraints like “we do not want to include some combination of items in the solution”, which is often required in actual use. To tackle this inconvenience, we have developed a general methodology for solving constraint-added variants of KP. The key to our method is **Zero-suppressed Binary Decision Diagram (ZDD)**. ZDD can represent a set of feasible solutions as a compact graph, and it can efficiently guide the solver to find the best solution of the constraint-added variants of KP. Our methodology will impact many real applications.

Knapsack problem with additional constraints

Select a combination of items with maximum value and satisfies weight limit and **additional constraints**.

Items

	value	weight
	¥500	3kg
	¥800	1kg
	¥300	2kg
	¥1000	2kg
	¥800	4kg

Knapsack problem (KP)

Find the combination of items with maximum value and weight is below **6kg**

The optimal solution within 6kg

Easy to solve

KP with additional constraints

There may exists some additional constraints in practical situations.

A bottle cannot be selected with a fragile item (Exclusion)

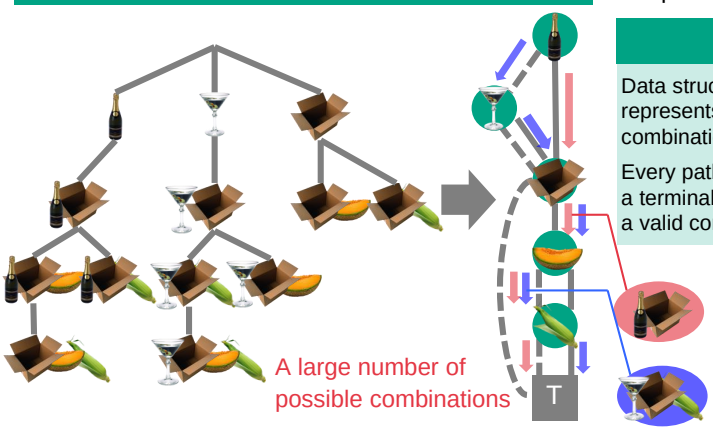
A box is required when we select fruits or vegetables (Ordering)

The optimal solution within 6kg that satisfies additional constraints

Difficult to solve

Combinatorial optimization using ZDD

Using **ZDD** to represent all feasible combinations of items and solve the optimization problem by a **dynamic programming** algorithm.



A large number of possible combinations

Possible combination satisfying additional constraints

The ZDD representing a set of combinations of items

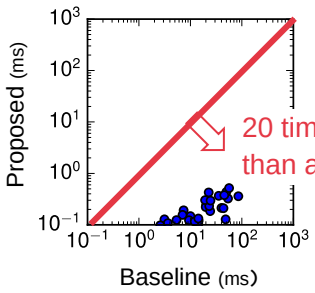
The paths corresponding to combinations of items

ZDD

Data structure that represents a set of combinations as a graph

Every path from the root to a terminal node represents a valid combination of items

Comparison of running time



Proposed method

- Can handle several constraints
- Can estimate the running time
- Efficient

Related works

[1] M. Nishino N. Yasuda, S. Minato, M. Nagata, “BDD-constrained search: a unified approach to constrained shortest path problems” in *Proc. 29th AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[2] M. Nishino N. Yasuda, T. Hirao, S. Minato, M. Nagata, “A dynamic programming algorithm for tree trimming-based text summarization” in *Proc. 2015 Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT),* 2015.

Contact

Masaaki Nishino Linguistic Intelligence Research Group, Innovative Communication Laboratory
E-mail：nishino.masaaki(at)lab.ntt.co.jp