I214 System Optimization
Chapter 6: Combinatorial Optimization (Part A)

Brian Kurkoski

Japan Advanced Institute of Science and Technology

2023 January

Outline

Combinatorial Optimization

In *combinatorial optimization*, the feasible set is a **discrete set**, and often a **finite set**.

We find the minimum value of an objective function over all discrete values in the feasible set.

For example, a problem has n variables, and each variable can be either 0 or 1.

6.1 Greedy Method

Most algorithms follow a step-by-step procedure. In a greedy algorithm,

- at each step make a locally optimal decision
- without considering the global solution.
- In special cases, greedy algorithms may give an optimal solution
- In general, greedy algorithms are not optimal, but provide acceptable approximations of the optimal solution.

Example: making change problem

6.1.1 Minimum Spanning Tree

An example of a problem that can be solved by a greedy algorithm is finding the minimum-spanning tree.

Definition

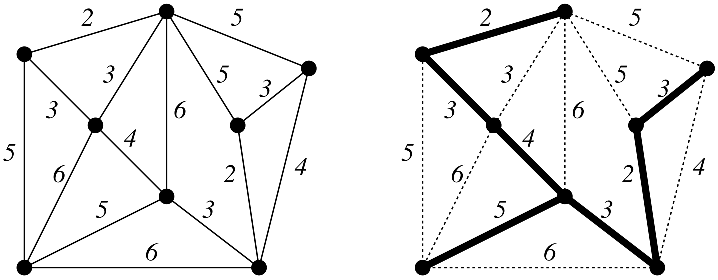
Given an undirected graph $(\mathcal{V}, \mathcal{E})$, a *spanning tree* T is a subgraph that is a tree which include all nodes $\mathcal{V}$.

Definition

Given a **weighted** undirected graph, a *minimum spanning tree* (MST) is a spanning tree with minimum weight.

$\bigstar 1$

Example of Minimum Spanning Tree (MST)



(a) Problem instance

(a) An optimum solution

Figure 1: An instance of the minimum spanning tree problem, and its one of the optimum solutions.

6.1.2 Kruskal's Algorithm

Kruskal Algorithm

1. Sort edges with their weights, and let its result be $e_{I_1}, e_{I_2}, \cdots, e_{I_m}$. That is,

$$\lambda(e_{I_1}) \leq \lambda(e_{I_2}) \leq \cdots \leq \lambda(e_{I_m})$$

2. $E_T = \{e_{I_1}\}$, $k = 2$.
3. If the set of edges $E_T \cup \{e_{I_k}\}$ does not include cycles, then $E_T \leftarrow E_T \cup \{e_{I_k}\}$.
4. If (V, E_T) is a spanning tree of G, then quit ((V, E_T) is a cost-minimum spanning tree).
   Otherwise, $k \leftarrow k + 1$ and go to Step 3.

6.1.3 Knapsack Problem

Problem

Given a set of n items, each with given weight a_i and utility $c_i \geq 0$, select the items to maximize the total utility while having total weight not greater than W.

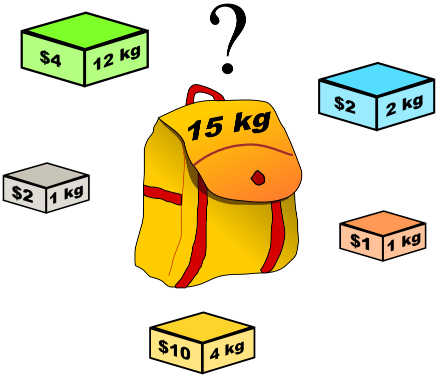


Figure 2: Knapsack Problem

Knapsack Problem Example

Using the following example

$$
\begin{aligned}
n &= 4 \\
(c_1, c_2, c_3, c_4) &= (7, 8, 1, 2) \\
(a_1, a_2, a_3, a_4) &= (4, 5, 1, 3) \\
W &= 6
\end{aligned}
$$

solve the problem using a greedy approach.

6.2 Branch-and-Bound Method

- A combinatorial optimization problem has a finite number of feasible solutions,
- an optimum solution can be found by enumerating and evaluating all feasible solutions.
- However, a naive enumeration has exponential complexity.
- In **branch-and-bound**, the computation time is shortened by skipping hopeless candidate solutions.

Branch-and-Bound Method

- Branch-and-bound can be represented on a tree
- Each level of the tree corresponds to a variable
- The algorithm explores branches of this tree
- The branch is checked against upper and lower estimated bounds on the optimal solution, and the branch is discarded if it cannot produce a better solution than the best one found so far.

Branch-and-Bound Method

0. Find a tentative solution by some appropriate method, and let z^* be the value of the objective function for this tentative solution. (The tentative solution and its corresponding tentative minimum value may be updated afterward.)
1. Generate subproblems P_1, P_2, $\cdots$, P_m from the original problem P_0.
   $\mathsf{A} \leftarrow \{\mathsf{P}_1, \mathsf{P}_2, \cdots, \mathsf{P}_m\}$
2. Choose one subproblem P_i from A and $\mathsf{A} \leftarrow \mathsf{A} \backslash \{\mathsf{P}_i\}$.
   - 2-1. If P_i has no feasible solution, then go to Step 4, otherwise go to Step 2-2.
   - 2-2. If the optimum solution of P_i is obtained, let z_i be the value of the objective function achieved by this solution, otherwise go to Step 2-3.
     When $z_i < z^*$, then update the tentative solution and $z^* \leftarrow z_i$.
     After that, go to Step 4.
   - 2-3. A lower bound $\underline{z_i}$ of the value of the objective function for solutions of P_i is computed.
     If $\underline{z_i} < z^*$, then go to Step 3.
     If $\underline{z_i} \geq z^*$, then go to Step 4.

Branch-and-Bound Method

3. Generate subproblems $\mathsf{P}_j, \cdots, \mathsf{P}_k$ from the subproblem P_i, update A as $\mathsf{A} \leftarrow \mathsf{A} \cup \{\mathsf{P}_j, \cdots, \mathsf{P}_k\}$, and go to Step 2.
4. If $\mathsf{A} = \emptyset$, then quit (the tentative solution at this time is an optimum (minimum) solution of the original problem P_0).
   Otherwise, go to Step 2.

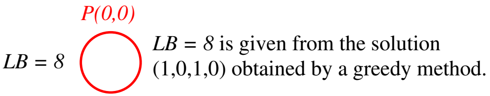
6.2.1 Knapsack Problem

- The branch-and-bound method may be used to solve the knapsack problem.
- Upper bound using relaxation
- Lower bound using best-found solution

Step 1

(1) With respect to the original problem $\mathsf{P}(\emptyset, \emptyset)$, compute an initial tentative solution (this example uses the solution $(1, 0, 1, 0)$ computed from a greedy method) and a lower bound $LB = 8$ for the objective value achieved by the optimum solution of $\mathsf{P}(\emptyset, \emptyset)$.

(1)



Step 2

Generate a subproblem P$(\{x_1\}, \emptyset)$ from P$(\emptyset, \emptyset)$ with fixing $x_1 = 0$.

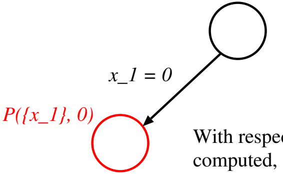
$$
\begin{array}{lll}
\mathrm{P}(\{x_1\}, \emptyset) & \text{Objective:} & 8x_2 + x_3 + 2x_4 \to \max \\
 & \text{Constraints:} & 5x_2 + x_3 + 3x_4 \leq 6 \\
 & & x_i = 0, 1 \quad (i = 2, 3, 4)
\end{array}
$$

In order to find the upper bound on the objective value achieved by the optimum solution for P$(\{x_1\}, \emptyset)$, we will solve the relaxed continuous version of P$(\{x_1\}, \emptyset)$.

$$
\begin{array}{lll}
\mathrm{P}(\{x_1\}, \emptyset)_{relax} & \text{Objective:} & 8x_2 + x_3 + 2x_4 \to \max \\
 & \text{Constraints:} & 5x_2 + x_3 + 3x_4 \leq 6 \\
 & & 0 \leq x_i \leq 1 \quad (i = 2, 3, 4)
\end{array}
$$

Step 2 continued

(2)



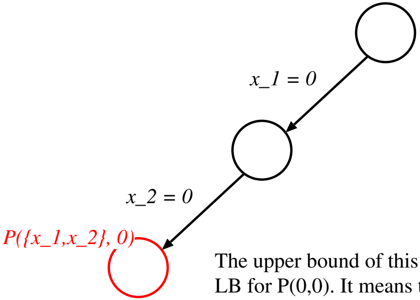
With respect to P({x_1},0), the solution of the relaxed subproblem is computed, and the upper bound UB = 8+1 = 9 for P({x_1},0) is obtained. Since UB (for P({x_1},0)) > LB (for P(0,0)), we will continue the search of subtree below this node

Comparing the upper bound $UB = 9$ on the objective value for P$(\{x_1\}, \emptyset)$ with the lower bound $LB = 8$ for P$(\emptyset, \emptyset)$, $UB > LB$ indicates the possibility of the presence of the optimum solution for P$(\emptyset, \emptyset)$ in the subproblem P$(\{x_1\}, \emptyset)$, and the search of the subproblems below this node will follow.

Step 3

(3) Subproblem P$(\{x_1, x_2\}, \emptyset)$ is generated from P$(\{x_1\}, \emptyset)$ with fixing $x_2 = 0$. With respect to this new subproblem, we find the optimum solution $(0, 0, 1, 1)$ for the relaxed continuous problem, which means that the upper bound for P$(\{x_1, x_2\}, \emptyset)$ is $UB = 1 + 2 = 3$. Since $UB = 3 < LB = 8$, we will skip the search of subproblems below this node, and go back to P$(\{x_1\}, \emptyset)$ (and generate another subproblem).

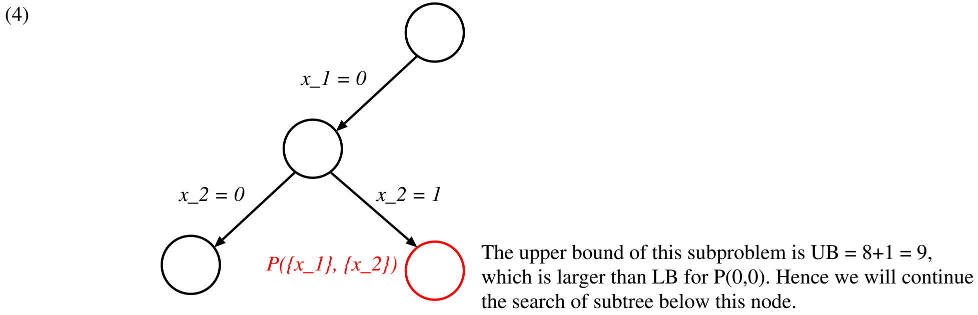
(3)



The upper bound of this subproblem is UB = 1+2 = 3, which is smaller than LB for P(0,0). It means that there is no better solution in this subproblem than the current tentative solution. Hence we stop the search of subtree below this node, and move to another subproblem.

Step 4

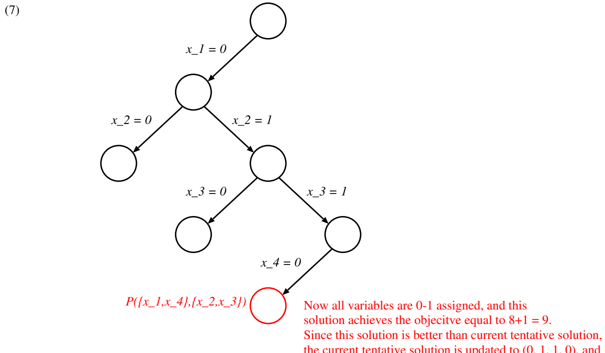
(4) Subproblem P$(\{x_1\}, \{x_2\})$ is generated from P$(\{x_1\}, \emptyset)$ with fixing $x_2 = 1$. The upper bound $UB = 9$ on the objective value for this new subproblem is then computed by solving the relaxed continuous problem (the solution is $(0, 1, 1, 0)$). Since $UB = 9 > LB = 8$, the search of subproblems below this node will follow.



Step 7

(5) and (6) are omitted

(7) Generate a new subproblem from P$(\{x_1\}, \{x_2, x_3\})$ fixing $x_4 = 0$, the result P$(\{x_1, x_4\}, \{x_2, x_3\})$ is a compute 0-1 assignment, i.e., a complete solution, for the original problem P$(\emptyset, \emptyset)$. This solution provides the objective value 9, which is larger than the current tentative solution for P$(\emptyset, \emptyset)$. The tentative solution is updated from $(1, 0, 1, 0)$ to $(0, 1, 1, 0)$ and the lower bound is updated from 8 to 9.



Class Info

- Tutorial Hours: Today at 13:30.
- Homework 6. Deadline: Friday, January 27 at 18:00.
- Homework 7 will be on LMS soon. Deadline: February 3 at 18:00
- Next lecture: Monday, January 30. Continue Combinatorial Optimization