Example 1: Capital Budgeting

A telecommunication company wants to build switching centers at certain given sites. The sites should be selected so that the total profit is maximized, under the constraint that the total cost remains within the available budget.

The following information is available to formulate the problem:

- There are N available sites.
- The cost of building a switching center at site i is c_i.
- If a center is built at site i, then it generates a profit p_i (say, per year).
- The total available budget that can be used for building the centers is C.
- Objective: select which of the N sites will be used for building switching centers, so that the total profit is maximized, while keeping the total cost within the budget. (Any number of sites can be selected out of the N.)

Let us formulate the problem as a mathematical program!

Variables: one variable, x_i, for each site i. The meaning of x_i is:

$$x_i = \begin{cases} 1 & \text{if site } i \text{ is selected as a center} \\ 0 & \text{if site } i \text{ is not selected as a center} \end{cases}$$

In other words, x_i is the *indicator variable* of the fact that site i is selected/not selected as a center.

Such indicator variables are frequently used in network planning formulations when one has to express yes/no choices.

Objective function

Goal: maximize the total profit.

How can we express it? The profit from site i is p_i if the site is selected as a center, otherwise 0. The definition of x_i suggests that this can be conveniently expressed as $p_i x_i$. Then the total profit is:

$$\sum_{i=1}^{N} p_i x_i.$$

Thus, the objective function is:

$$\max Z = \sum_{i=1}^{N} p_i x_i.$$

Constraints:

The total cost cannot be more than the budget C:

$$\sum_{i=1}^{N} c_i x_i \leq C$$

The variables can only take 0-1 values:

$$x_i \in \{0,1\}, \qquad i = 1, \ldots, N.$$

Collecting the parts, the complete formulation is this:

$$\max Z = \sum_{i=1}^{N} p_i x_i$$

Subject to

$$\begin{aligned} \sum_{i=1}^{N} c_i x_i &\leq C \\ x_i &\in \{0,1\}, \qquad i = 1, \ldots, N \end{aligned}$$

Comment: Though we have not mentioned it, one can easily recognize that this problem is exactly identical to the well known *Knapsack Problem.*

Why is this good? Because we can then use all the methodology that has been developed for the Knapsack Problem, so we can stand "on the shoulders of giants."

How can we find a solution?

It is well known from the study of algorithms that the Knapsack Problem in the general case is NP-complete. In the special case when the coefficients are polynomially bounded, it is known to have a Dynamic Programming solution, but in the general case (where the coefficients can be exponentially large) we cannot reasonably hope to find the *exact* optimum for a large problem, via an efficient algorithm. A good approximation, however, is still possible:

Principle of the heuristics: Sort the sites according to their profit/cost ratio. (Clearly, a higher profit/cost ratio is more preferable.) Select the sites one by one in this order, thus, always trying the most preferable first, among the yet unselected ones. (Preference = profit/cost). If the considered site still fits in the remaining budget, then select it, otherwise proceed to the next one in the preference order.

This is a natural variant of the so called *Greedy Algorithm*.

Trying the sites according to the profit/cost preference order is so natural, that one may wonder: why it does not *guarantee* an optimum solution?

Exercise (to test your understanding for yourself): Find a specific example when the above greedy algorithm does not yield the optimum.

A simple example is on the next page, but try to find one yourself, before proceeding to the next page.

Solution to the exercise

Consider the following data:

$$\begin{aligned} &p_1 = 90, \qquad c_1 = 10 \\ &p_2 = 20, \qquad c_2 = \ \ 2 \\ &p_3 = \ \ 8, \qquad c_3 = \ \ 1 \end{aligned}$$

Budget: $C = 11$

Then the profit/cost ratios are:

$$p_1/c_1 = 9, \qquad p_2/c_2 = 10, \qquad p_3/c_3 = 8.$$

The Greedy Algorithm would select $i = 2$ first, since p_2/c_2 is the largest. This choice uses $c_2 = 2$ units from the budget. The next largest ratio is p_1/c_1, but $c_1 = 10$ already does not fit in the remaining budget of $11 - 2 = 9$. So the algorithm proceeds to $i = 3$, which still fits in the budget.

Thus, the Greedy Algorithm selects sites 2 and 3, achieving a total profit of $20 + 8 = 28$. On the other hand, if we select sites 1 and 3, then we can achieve a profit of $90 + 8 = 98$, while the cost still fits in the budget.

Comment

As we have seen, the Greedy Algorithm does not necessarily find the optimum for this problem. A natural question is:

How far is it from the optimum? In other words, can we prove some *performance guarantee* for the greedy solution?

The answer is yes. We present the preformance guarantee below.

Theorem:

Let T_{opt} be the optimum solution to the considered problem, that is, the maximum achievable total profit under the budget constraints. Let T_{greedy} be the profit achieved by the Greedy Algorithm. Further, denote by p_{max} the largest profit coefficient, that is, $p_{max} = \max_i p_i$. Then

$$T_{greedy} \geq T_{opt} - p_{max}$$

always holds.

Proof: See the Appendix at the end of this note. The proof is included only for showing how can one possibly analyze the result of the greedy heuristic in this situation, you do not have to learn the proof details.

Exercises

1. Assume that we execute the Greedy Algorithm on a specific Capital Budgeting problem and we find that $T_{greedy} = 1000$ (in some appropriate units). Further, we also know that the largest profit coefficient is $p_{max} = 5$. Give an estimate on the optimum profit T_{opt}.

Solution

From the preceding Theorem we know that

$$T_{greedy} \geq T_{opt} - p_{max}$$

always holds. Rearranging, we get

$$\frac{T_{greedy}}{T_{opt}} \geq 1 - \frac{p_{max}}{T_{opt}}.$$

Since $T_{opt} \geq T_{greedy}$ by definition, therefore, if we replace T_{opt} by T_{greedy} on the righthand-side, then p_{max}/T_{opt} can only grow. This means, we can only subtract more from 1, so

$$1 - \frac{p_{max}}{T_{opt}} \geq 1 - \frac{p_{max}}{T_{greedy}}$$

must hold. As a result, we get

$$\frac{T_{greedy}}{T_{opt}} \geq 1 - \frac{p_{max}}{T_{greedy}}.$$

Substituting the given numerical values on the righthand-side yields

$$\frac{T_{greedy}}{T_{opt}} \geq 1 - \frac{5}{1000} = 0.995 = 99.5\%.$$

This means, in this case we can argue that the Greedy Algorithm will give a result that is within 0.5% of the optimum, even though we do not know what the optimum is!

The value of $T_{greedy} = 1000$ gives a lower bound on T_{opt}. If we also want an upper bound, then from the Theorem we have $T_{greedy} \geq T_{opt} - p_{max}$, which, after rearranging, yields

$$T_{opt} \leq T_{greedy} + p_{max}.$$

Using the numerical values, we get $T_{opt} \leq 1005$.

2. In the example which showed that the Greedy Algorithm can produce a non-optimal solution, the greedy solution was less than half of the optimum. Can we modify the Greedy Algorithm such that it always guarantees *at least half* of the optimum profit?

Solution

To exlude trivial cases, let us assume that $c_i \leq C$ for every i. (If $c_i > C$ for some i, then it can be excluded, since it never fits in the budget).

If $T_{opt} \geq 2p_{max}$, then we have

$$T_{greedy} \geq T_{opt} - p_{max} \geq \frac{T_{opt}}{2},$$

since the first inequality follows from the Theorem and the second inequality, after rearranging, is equivalent to $T_{opt} \geq 2p_{max}$.

Thus, if $T_{opt} \geq 2p_{max}$, then the greedy solution satisfies the requirement. What if $T_{opt} < 2p_{max}$? This means $p_{max} > T_{opt}/2$, so then simply selecting the site with maximum profit already achieves more than half of T_{opt}. Let us call this latter choice 1-site heuristic. Clearly, the profit achieved by the 1-site heuristics is p_{max}.

It follows that at least one of the greedy and 1-site heuristics gives half or more of T_{opt}. But we do not know in advance which one. There is, however, an easy way to know it: run both and take the result that gives the larger profit. Thus, the required modification is:

Run the Greedy Algorithm to obtain T_{greedy}. If $T_{greedy} \geq p_{max}$, then keep the greedy solution, otherwise replace it with the 1-site heuristics.

Appendix

Proof of the Theorem.

Let $x_1, x_2, \ldots, x_n$ be the solution found by the greedy algorithm. Similarly, let $y_1, y_2, \ldots, y_n$ be an optimal solution. If the greedy solution is not optimal, then there must be an index j for which $x_j = 0$ and $y_j = 1$. The reason is that the 1s in the optimal solution cannot form a proper subset of the 1s in the greedy solution, since then the greedy would have higher value than the optimal. Thus, if the greedy is not optimal, then there must be a 1 in the optimal solution that is not in the greedy solution.

Let us assume that the variables are indexed such that

$$\frac{p_1}{c_1} \geq \frac{p_2}{c_2} \geq \ldots \geq \frac{p_n}{c_n}$$

holds and let j be the smallest index with $x_j = 0,\ y_j = 1$. Then we can write for the total profit obtained by the greedy algorithm:

$$T_{greedy} = \sum_{i=1}^{n} p_i x_i \geq \sum_{i=1}^{j} p_i x_i = \sum_{i=1}^{j} p_i y_i + \sum_{i=1}^{j} p_i (x_i - y_i)$$

Let us now use the fact that

$$\frac{p_i}{c_i} \geq \frac{p_j}{c_j}$$

holds if $i \leq j$, due to the chosen indexing. Therefore, we have

$$p_i \geq \frac{p_j}{c_j} c_i$$

and then substituting p_i by $\frac{p_j}{c_j}c_i$ in the last summation, we obtain

$$T_{greedy} \geq \sum_{i=1}^{j} p_i y_i + \sum_{i=1}^{j} \frac{p_j}{c_j} c_i (x_i - y_i).$$

Note that $x_i - y_i \geq 0$ holds for $i < j$, since, by definition, j is the first index with $x_j = 0,\ y_j = 1$. Therefore, the inequality remains correct after the substitution, as we have no negative number in the summation for $i < j$. When $i = j$, then $x_i - y_i < 0$, but then the coefficient does not change, being $p_i/c_i = p_j/c_j$ for $i = j$.

Further rearranging, we get

$$T_{greedy} \geq \sum_{i=1}^{j} p_i y_i + \frac{p_j}{c_j} \left(\underbrace{\sum_{i=1}^{j} c_i x_i}_{A} - \underbrace{\sum_{i=1}^{j} c_i y_i}_{B} \right).$$

Let us bound now the sums denoted by A and B above.

A is the cost incurred by the first j variables in the greedy solution. This does not include c_j, since $x_j = 0$. Moreover, c_j cannot be added without violating the budget constraint, since otherwise the greedy algorithm would have set $x_j = 1$. Therefore, we must have $A + c_j > C$, which yields

$$A = \sum_{i=1}^{j} c_i x_i > C - c_j.$$

Considering the other sum (B), we can use the fact that the optimal solution must also fit in the budget. Therefore,

$$\sum_{i=1}^{n} c_i y_i \leq C$$

holds, which implies

$$B = \sum_{i=1}^{j} c_i y_i \leq C - \sum_{i=j+1}^{n} c_i y_i.$$

Substituting the sums A and B by the bounds, we can observe that the expression can only decrease, as A is substituted by a smaller quantity, while the subtracted B is substituted by a larger quantity. Thus, we obtain

$$T_{greedy} \geq \sum_{i=1}^{j} p_i y_i + \frac{p_j}{c_j}\left(\underbrace{C - c_j}_{<A} - \underbrace{\left(C - \sum_{i=j+1}^{n} c_i y_i\right)}_{\geq B}\right).$$

After rearranging, we get

$$T_{greedy} \geq \sum_{i=1}^{j} p_i y_i + \sum_{i=j+1}^{n} \frac{p_j}{c_j} c_i y_i - p_j$$

Using now that $\frac{p_j}{c_j} \geq \frac{p_i}{c_i}$ when $i > j$, the second sum can only decrease if the coefficient of y_i is replaced by p_i. This yields

$$T_{greedy} \geq \sum_{i=1}^{j} p_i y_i + \sum_{i=j+1}^{n} p_i y_i - p_j = \sum_{i=1}^{n} p_i y_i - p_j$$

We can now observe that the last summation is precisely T_{opt}, while $p_j \leq p_{max}$, so if we subtract p_{max} instead of p_j, then the expression can only decrease. Thus, we obtain

$$T_{greedy} \geq T_{opt} - p_{max}$$

which proves the Theorem.