The Application of Greedy Algorithm in Real Life

Jun Liu, Chuan-Cheng Zhao and Zhi-Guo Ren

ABSTRACT

Greedy algorithm, also known as voracity algorithm, and is simple and easy to adapt to the local area of the optimization strategy. In other words, every time it makes the choice is the best choice in the current. It is not considered from the point of view of the overall optimization to consider, but every time the choice is only in a sense of the local optimal choice. So, for a problem, we should consider whether it can use the greedy algorithm strategy, and whether the solution we find is the optimal solution. It is not an overall optimal solution to any problem, but most of the problems with a wide range of problems can be solved by the global optimal solution or the similar solution of the global optimal solution.

With the arrival of the era of big data, a large number of data information needs to be processed in a timely manner, In order to meet the needs of people to deal with a large number of data information, and to solve various practical problems in a timely manner, computer algorithm got the rapid development, A large number of operations research models have been applied to the computer algorithm, which has produced many effective algorithms to solve practical problems.¹

Greedy algorithm in solving the problem, it is from the initial stage, in each stage is to make a local optimal greedy choice. Each time the greedy choice transforms the original problem into a sub problem of the same form as the original problem. Greedy algorithm has no fixed algorithm framework, the key of the design algorithm is the choice and determination of greedy strategy.

¹ Jun Liu. Institute of Information Technology Application, Lanzhou City University.No.11 Jiefang Road, Anning District, Lanzhou City, Gansu Province, 730070, P.R.China. Supported by the Doc’s Foundation (LZCU-BS2013-08; LZCU-BS2013-09).
1. THE BASIC IDEA AND IMPLEMENTATION PROCESS OF THE GREEDY ALGORITHM

1.1 The basic idea of the greedy algorithm

Greedy algorithm is a step by step, according to a certain optimization measure, each step should be able to ensure that the local optimal solution can be obtained. If the next data and partial optimal solution is no longer feasible solution, then the data can not be added to the partial optimal solution until all the data are enumerated or can not be added so far. The hierarchical processing method of the optimal solution which can be obtained by some kind of measure is called the greedy strategy.

If you want to use the greedy strategy to solve the problem, it is necessary to solve the following two problems:

1. Whether the problem can be solved by greedy strategy;
2. How to determine the greedy choice strategy, to get the best or better solution.

1.2 The realization process of greedy algorithm

1. Select the appropriate greedy choice as the standard.
2. It is proved that the problem has the greedy choice property under this standard.
3. It is proved that this problem has the optimal substructure property.
4. According to the greedy choice criterion, the greedy algorithm is designed, and the optimal solution is obtained.

2. THE CORE OF THE GREEDY ALGORITHM

The core problem of the greedy algorithm is to choose the best measure of the optimal solution of the problem, that is, the specific greedy strategy.

Greedy strategy is to point out the initial state of the problem, through each of the greedy choice and get the optimal value of a problem solving method. The choice by the greedy strategy made, all is the optimal selection at the present times, and not from the whole into consideration, the solution is just a sense of local optimal solution, so characteristic of the problem itself, decided to whether we can use the greedy strategy to obtain the optimal solution.

2.1 The character of greedy choice

The so-called greedy nature, refers to the application of the same rules, the original problem into a similar, but the size of the smaller sub problems, each
step in the choice is the best choice for the current. This choice is dependent on
the choice made, and does not depend on the choices made. Greedy algorithm
usually adopts top-down manner, in iterative manner make subsequent greedy
selection, through each step of the greedy choice, we can get an optimal solution,
although each step can be guaranteed to obtain a local optimal solution, but the
resulting overall solution sometimes is not necessarily the optimal. The use of
greedy strategy to solve the problem in the process of running without
backtracking process, which is the first basic elements of the greedy algorithm.³

How to determine the nature of the greedy choice, we must prove that greedy
choice will lead to the overall optimal solution. First of all, it is proved that the
existence of a global optimal solution of the problem must contain the first
greedy choice. Then it is proved that the original problem is simplified to a
similar sub problem with smaller size after being greedy. The mathematical
induction method can be used to prove that the overall optimal solution can be
obtained through a series of greedy choices. Among them, it is proved that the
choice of the problem is the key to simplify the similar sub problem of smaller
size.

2.2 Optimal substructure property

When the optimal solution of a problem contains the optimal solution of
the sub problem, it is called the optimal sub structure. This is the second basic
elements of the greedy algorithm, also to determine a problem can be used to
solve dynamic programming or greedy algorithm to solve another key feature.
Greedy algorithm and dynamic programming are two algorithms which
require the optimal sub structure. Greedy algorithm is relatively simple, the
efficiency is higher, but the solution is not necessarily the overall optimal
solution. While the dynamic programming solution to the problem by it, and
then through the deconstruction of sub problems make the solution of the
original problem, relatively complex.

2.3 The fit problem about greedy algorithm

The greedy algorithm is usually used to solve optimization problems with
the maximum or minimum value. It is from a certain initial state and
according to the local rather than the global optimal decision, to satisfy the
equation of constraint conditions and to make the objective function value of
the fastest or slow as a criterion, choose a most quickly reach the requirements
of input elements, so as soon as possible to pose a problem for solution.

Optimization problem: the inputs numbers is N, and its solution by the N
input to meet some of the prior given constraints of a subset of the set of
constraints, and to meet the constraints of a subset of the conditions known as
the solution of the problem. Obviously, the feasible solutions in general not
only, in order to measure the quality of the feasible solution, the problem also
give a value function, called the target function is take the objective function extremum value (maximum or minimum) of feasible solution, known as the optimal solution.

The solution of the optimization problem can be expressed as a n tuple X= (x₁, x₂,..., xₙ), each of which is taken from a set of values S, all of which allow the n-tuple to form a candidate solution set.

Greedy algorithm is to solve the problem by the method of step by step. Greedy algorithm in solving the problem of every step to make some decisions, resulting in a component of n-tuples, greedy algorithm requires the selected a best measure of the standard, as the basis of the current component, the greedy algorithm at each step in the selection criteria for decision-making basis is called optimal measure (greedy criterion, also known as greedy choice property).

3.THEORETICAL BASIS OF GREEDY ALGORITHM

The theoretical basis of greedy algorithm is matroid.

The matroid, also known as the matrix embryo, is an important content of combination optimization and graph theory.

Definition: matroid is a sequence of M=[S, I], where S is an ordered non empty set, I is a nonempty set of S, and become a independent subsets for S. If M is a matrix words of 

\[
M = \begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1m} \\
a_{21} & a_{22} & \ldots & a_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \ldots & a_{nm}
\end{bmatrix}
\]

S is every lines of M, S= (a₁, a₂, ..., aₙ), I is several line of liner independence: aᵢ, aⱼ, aᵣ, ....

If M is an undirected graph matroid of G, Then S is the edge set of the graph, I is a subset of the set of edges that make up the forest.

If the S towards each element X () to give a positive weight W (X), called the M= (S, I) as a weighted matrix.

Suitable to use greedy algorithm to solve many of the problems can be attributed to the weighted matroid array to find a problem with the maximum weight independent subset of a, that is, given a weighted intends to matrix M = (s, I), if we can find an independent and has a maximum weight subset of a, and a is not m greater than its independent subset contains, then an optimal subset is a maximal independent subset.

The theory of matroid array is a kind of theory that can determine when the greedy algorithm can produce the optimal solution, and it is playing an increasingly important role in solving the problem of the optimal solution.
4. RAISE THE PROBLEM OF ACTIVITY SELECTION

We assumption the activity members is n, set E={1, 2, ..., n}, these activities use the same resources, such as the lecture hall, and so on, and this resource at the same time there is only one activity can be used. Each activity has a start time $s_i$ and an end time $f_i$, and $0 \leq s_i < f_i < \infty$. If the activity I is selected, it occupies the resource in the half open time interval $[s_i, f_i)$, if the two activities $a_i$ and $a_j$ satisfy $[s_i, f_i)$ and $[s_j, f_j)$ do not intersect, then said the activity I and activity J is compatible. In other words, when $s_i \geq f_j$ or $s_j \geq f_i$, activity i and j are compatible. An algorithm is designed to obtain the maximal consistent set of activities in all the given activities.

4.1 Greedy algorithm strategy

Apply the greedy algorithm to solve this problem, use the A set to store the selected activities. It is assumed that the activity we choose is 1, and the j is initialized 1. Then we check whether the activity i is compatible with the activity that has been selected in turn. If compatible, then the activity i is added to the collection A, or do not choose activity i, continue to check whether the next activity with the collection A compatibility. Because the activity time in the set A is always the most active time, so after the check is finished, the activity in the set A is the largest consistent set of activities.

4.2 Analysis of algorithm

4.2.1 The nature of greedy choice

For the problem of activity selection, firstly we should choose an activity intuitively, and then choose the remaining resources should be used as far as possible by other tasks. We choose the activity which we can consider, and must have one activity end in the first. So, we should choose the activity which ended first, after we select it the remaining resources can be used to make the rest of the more active. The choice of the first end of the activity, is a kind of greedy choice method. Therefore, the activity selection problem has the greedy choice property.

4.2.2 The optimal substructure

We get rid of the end time which the starting time greater than 1, given the first activity in set A is K, take out K in A, so there is a recurrence relation, in the total activity and the activity of the 1 compatibility of the solution is bound to be compatible with the optimal solution A-K. So it can be the beginning of a greedy choice, so, each step of the greedy choice can make the original problem
into a smaller problem with the original problem. Therefore, the active selection problem has the optimal substructure.

4.2.3 Problem analysis

Given: the starting time and the ending time of 11 activity to be arranged. According to the end time of the ascending order as shown in the table:

<table>
<thead>
<tr>
<th>Activities schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
</tr>
<tr>
<td>s[i]</td>
</tr>
<tr>
<td>f[i]</td>
</tr>
</tbody>
</table>

Designed the reasonable and effective algorithm, to find the most able to arrange the number of activities.

4.2.4 Example analysis

Activities in accordance with the end time has been ordered from small to large, i represents the first i activities, s[i] which represents the beginning time of the first i activities, f[i] representing the end time of the first i activities.

In these activities, pick out the end time earlier activities, and to satisfy the next activity start time to end time later than the previous activity, find the set of all of these activities is the largest compatible activities set. The system checks i every activity, to determine whether it is compatible with the current selection of all activities. If it compatible, the active i to join the set A which has selected activities, otherwise, do not choose the activity i, and continue to determine the next activity and set A compatibility of all activities.

5 CONCLUSION

This paper mainly introduces the definition of greedy algorithm, the greedy algorithm the basic idea and implementation process, core of the greedy algorithm, the basic elements and theoretical basis of greedy algorithm, and combined with specific examples, to show the superiority of greedy algorithm in solving the problem, the important significance in real life of the presence time. Greedy algorithm as a kind of operations research model has unique advantages in many optimization problems, is an effective method to solve the problems fast in the real word, but it is also exist the deficiency, how to make the greed algorithm utilize the real life better. This is also a major issue that needs to be explored.

REFERENCES