Sutra: International Journal of Mathematical Science Education

Vol. 1, No. 1, 30 - 35, 2008

CLASS-ROOM NOTES:

OPTIMIZATION PROBLEM SOLVING - I

R. Akerkar

Technomathematics Research Foundation
Kolhapur, India
kpr_tmrf@bsnl.in

Purpose of these notes is to discuss the optimization problem. An optimization problem consists of finding the best (cheapest, heaviest, etc.) element in a large set P, called the feasible region and usually specified implicitly, where the quality of elements of the set are evaluated using a function *f(x)*, the objective function, usually something fairly simple. The element that minimizes (or maximizes) this function is said to be an optimal solution of the objective function at this element is the optimal value. The greedy approach is a technique in which an optimal solution is obtained by means of the set of choices. A greedy algorithm always makes the choice that looks best at that moment. In this lecture, we will discuss greedy approach. To determine whether the greedy approach will yield the optimal solution the basics of matroid theory are presented.

Keywords: optimization, greedy algorithm, matroid theory.

1. Introduction

Greedy approach is a technique in which an optimal solution is obtained by the means of the set of choices. The idea of greedy approach is to make a locally optimal choice, hoping that this choice will lead to a globally optimal solution. There are many problems for which greedy approach provides an optimal solution much more quickly than a dynamic programming approach would have done.

Greedy algorithms are algorithms, which follow the problem solving meta-heuristic of making the locally optimum choice at each stage with the hope of finding the global optimum.

While most solution algorithms make choices based on a global overview of all current and future possibilities aiming at reaching the single global optimum solution, greedy algorithms make choices that look best at that very moment. In other words, they make locally shortsighted choices believing that the choice will eventually lead to the global optimum solution.

In general, greedy algorithms have five pillars (*Wikipedia*):

- A candidate set, from which a solution is created,

- A selection function, which chooses the best candidate to be added to the solution,
- A feasibility function, that is used to determine if a candidate can be used to contribute to a solution,
- An objective function, which assigns a value to a solution, or a partial solution, and
- A solution function, which will indicate when we have discovered a complete solution.

In order to develop a greedy algorithm the following steps can be used.

- Determine the optimal substructure of the problem.
- Develop a recursive solution.
- Prove that at any stage of the recursion, one of the optimal choices is the greedy choice. Thus it is safe to make the greedy choice.
- Show that all but one of the subproblems induced by having made the greedy choice are empty.
- Develop a recursive algorithm that implements the greedy strategy
- Convert the recursive algorithm to an iterative algorithm.

However in practice we usually streamline the above steps, when designing a greedy algorithm.

These steps can be reduced to more general sequence of steps.

1. Cast the optimization problem as one in which a choice is made and we are left with one subproblem to solve.
2. Prove that there is always an optimal solution to the original problem that makes the greedy choice.
3. Show that, having made the greedy choice, the part that remains is a subproblem with the property that if we combine an optimal solution to the subproblem with the greedy choice made, an optimal solution to the problem will be reached.

There is no way in general that one can specify if a greedy algorithm will solve a particular optimization problem. However if the following properties can be demonstrated, then it is probable to use greedy algorithm.

- *Greedy-choice property*: a globally optimal solution can be arrived at by making a locally optimal greedy choice. That is, when we are considering which choice to make, we make the choice that looks best in the current problem, without considering results from subproblems.
- *Optimal substructure*: A problem exhibits optimal substructure if an optimal solution to the problem contains within it optimal solutions to subproblems.

When compared to algorithms that guarantee to yield a global optimum solution: greedy-choice property, greedy algorithms have several advantages. They are easier to implement, they require much less computing resources, they are much faster to execute. Their only disadvantage is that they don’t reach the globally optimal solution consistently, since they usually don't operate exhaustively on all the data. Nevertheless they are useful because they are quick to think up and often give good approximations to the optimum. If a greedy algorithm can be proven to yield the global optimum for a given problem class, it naturally becomes the method of choice. To determine whether the greedy approach will yield the optimal solution the *matroid theory* can be used. This theory covers many cases of particular interest and is being rapidly developed and extended to cover more applications.

Matroids are abstract structures and they support greedy algorithms. It turns out that the structures, for which the greedy algorithm does lead to an optimal solution, are the matroids. It is worth studying them, not only because it enables us to recognize when the greedy algorithm applies, but also because there exist fast algorithms for `intersections' of two different matroids.

2. Essentials of Matroid Theory

The name "matroid" suggests a structure related to a matrix. Hassler Whitney introduced matroids theory in 1935 to capture abstractly the essence of dependence in linear algebra and graph theory. Since then, it has been recognized that matroids arise naturally in combinational optimization and can be used as a framework for approaching a diverse variety of combinational problems.

Definition 1: Let S be a finite set and let I be nonempty family of subsets of S. Then the pair $M=(S; I)$ is called a matroid if it satisfies the following conditions:

(i) Æ $\in I$,

(ii) If $B \in I$ and $A \subseteq B$ then $A \in I$,

(iii) If $A \in I$ and $B \in I$ and $|A| < |B|$ then $A \cup \{x\} \in I$ for some $x \in B\text{-}A$. (Exchange property)

Further, we will see some more notions.

Given $M=(S; I)$ matroid, element $x \notin A$ is called an *extension of* $A \in I$ id x can be added to A while preserving independence.

If A is an independent subset in a matroid M, A is *maximal* if it is not contained in any larger independent subset of M.

Let $Y \subseteq X$. A subset B of Y is called a *basis* of Y if B is an inclusion wise maximal independent subset of B. That is, for any set $Z \in I$ with $B \subseteq Z \subseteq Y$ one has $Z = B$.

A base of M is a maximal independent subset of S.

The common cardinality of all bases is called the rank of the matroid. If I is the collection of forests in a connected graph $G = (V, E)$, then the bases of the matroid (E, I) are exactly the spanning trees in G.

Definition 2: $M=(S; I)$ matroid is *weighted* if there is an associate weight function ω that assigns a strictly positive weight ω (x) to each element $x \in S$. the weight function ω extends to subsets of S by summation. $\sum_{x \in A} w(x)$ for any $A \subseteq S$.

In given $M=(S; I)$ matroid, a subset A where $A \in I$ and A has maximum possible weight is called an *optimal subset of matroid*.

There are different examples of matroids. Most useful ones are matric matroids, graphic matroids, uniform matroids and vectorial matroids. We will now discuss graphic matroids with certain details.

Graphic matroid can be defined as *MG*=(*SG*, *IG*) in terms of given undirected graph *G*=(*V*,*E*) in the following manner.

- The set *SG* is defined to be *E*, the set of edges of *G*.
- If $A \subseteq E$, then $A \in IG$ if and only if *A* is acyclic. That means, a set of edges *A* is independent if and only if the sub graph *GA*=(*V*, *A*) forms a forest.

Theorem 1: *If G=(V, E) is an undirected graph, then MG=(SG, IG) is a matroid.*
Proof: Conditions (i) and (ii) are trivial. To see that condition (iii) holds, let's assume that *GA*=(*V*, *A*) and *GB*=(*V*, *B*) are forests of *G* and that $|B|>|A|$. From graph theory it follows that a forest having k edges contains exactly $|V|$-*k* trees. Thus, forest *GA* contains $|V|-|A|$ trees and forest *GB* contains $|V|-|B|$ trees.

Since forest *GB* has fewer trees than forest *GA*, forest *GB* must contain some tree *T* whose vertices are in two different trees in forest *GA*. Since *T* is connected, it must contain an edge (*u*, *v*) such that vertices *u* and *v* are in two different trees in forest *GA* $\Rightarrow$ the edge (*u*, *v*) can be added to forest *GA* without creating cycle. So *MG* satisfies the exchange property and is matroid.

Theorem 2: *All maximal independent subsets in a matroid have the same size.*
Proof: Proof follows from exchange property of matroid definition. (*Self-study*!)

Theorem 3*: (Closure axioms) A function s : 2^S ® -2^S is the closure operator of a matroid on S if and only if for any A,B Í S and x, y Î S the following conditions hold.*
(S1) A Í sA.
(S2) If A Í B then sA Í sB.
(S3) sA = s sA.
(S4) If y Ï A, y Î s(A È {x}), then x Î s (A È {y}).

Theorem 4*: (Independency axioms) a collection I of subsets S is the set of independent sets of a matroid on S if and only if the following conditions hold.*
(I1) Æ Î I
(I2) If A Í B Î I, then A Î I.
(I3) If A, B Î I and |A| < |B|, then A È {y} Î I for some y Î B - A.

Theorem 5 *(Base axioms) a non-empty collection B of subsets S is the set of bases of a matroid on S if and only if the following conditions hold.*
(B1) If B1, B2 Î B and x Î B1 - B2 then B1 – {x} È {y} Î B for some y Î B2 - B1.

Theorem 6 *(Base axioms) a non-empty collection B of subsets of S is the set of bases of a matroid on S if and only if the following conditions hold.*

(B2) If B1, B2 Î B and x Î B1 – B2, then B2 È {x – y} Î B for some y Î B2 - B1.

3 Greedy Algorithm for Weighted Matroid

There are many problems that can be formulated as problems of finding a maximum-weight independent subset in a weighted matroid for which greedy approach can be used to provide optimal solutions. Greedy algorithm takes as input a weighted matroid *M*=(*S*, *I*) with an associated positive weight function ω, and it returns an optimal subset *A*.

The greedy algorithm consists of selecting *y1,…, yr* successively as follows. If *y1,….,* *yk* have been selected, choose $y \in S$ so that:

(1) $y \notin$ {*y1,….,yk*} and { *y1,….,yk, y*}$\in$ *I*,

(2) ω (*y*) is as large as possible among all y satisfying (1).

We stop if no y satisfying (1) exist, that is, if {*y1,….,yk* } is a basis.

In pseudocode Greedy algorithm can be presented like this.

```
Algorithm GREEDY (M, w)
        1. sort the elements according to weights: ω (y1) ≥ ω
           (y2) ≥ . . . ≥ ω (ym) ;
        2. let A= ∅ ;
        3. for i = 1 to m do
        4. if A ∪ {yi}  ∈ I then A = A È {yi}  .
```

The time cost is $O(n \log n + nf(n))$.

This is greedy algorithm is exactly Kruskal's algorithm (only for Kruskal algorithm the elements should be sorted in increasing order according to weights.).

Theorem 7: *If M = (S, I) is a weighted matroid with weight function w, then GREEDY(M,w) returns an optimal subset.*

The proof of the theorem is based on three lemmas discussed below.

Lemma 1: Let *x* be the first element of *S* such that *{x}* is independent, if any such *x* exists. If *x* exists, then there exists an optimal subset *A* of *S* that contains *x*.

Proof: If no such *x* exists, then the only independent subset is ∅ and we are done. Otherwise, let B be any nonempty optimal subset.

Assume that *x* Ï *B*, otherwise *A* = *B* is well done.

1. ∀*y* Î *B*, we have *w*(*y*) ≤*w*(*x*) because of the choice of *x*.

2. Construct *A* as follows:

(a) Let *A* = *{x}*, then *A* is independent.

(b) Repeatedly find new element of *B* than can be added to *A* until |*A*| = |*B*|, preserving the independence of *A*.

(c) Then, *A* = *B* - *{y}* È *{x}* for some *y* Î *B*. And *w*(*A*) = *w*(*B*) - *w*(*y*)+*w*(*x*)³ *w*(*B*)

So, *A* must be also optimal and *x* Î *A*.

Lemma 2 **:** For any matroid *M* = (*S*, *I*), if *x* Î *S* is an extension of some independent subset *A* of *S*, then *x* is also an extension of ∅.

Proof: Since x is an extension of A, we have that $A \cup \{x\}$ is independent. Since I is hereditary $\{x\}$ must be independent. Thus, x is an extension of $\emptyset$.

Corollary 1 : Let $M = (S, I)$ be any matroid. If $x \in S$ is not an extension of $\emptyset$, then x is not an extension of any independent subset A.

Remark: This corollary says that any element that cannot be used immediately can never be used. So, GREEDY never makes error when it skips over any initial element in S that is not an extension of $\emptyset$

Lemma 3 *:* Let $x \in S$ be the first element chosen by GREEDY for the weighted matroid $M = (S, I)$. The remaining problem of finding a maximum-weight independent subset containing x reduces to finding a maximum-weight independent subset of the weight matroid $M' = (s', I')$, where

1. $S' = \{y \in S \mid \{x, y\} \in I\}$
2. $I' = \{B \subseteq S - \{x\} \mid B \cup \{x\} \in I\}$, and the weight function for M' is the weight function for M, restricted to S'.

Proof:

1. If A is any maximum-weight independent subset of M containing x, then $A' = A - \{x\}$ is an independent subset of M'.
2. Conversely, any independent subset A' of M' yields an independent subset $A = A' \cup \{x\}$ of M.
3. Since we have in both cases that $w(A) = w(A') + w(x)$, a maximum-weight solution in M containing x yields a maximum weight solution in M', and vice versa.

Proof of Theorem 7: *using Lemmas proven above we can conclude.*

(1) Any elements that are passed over initially can be forgotten forever. Once an element x is selected, there must be an optimal containing it.

(2) $B \in I'$ is independent if $B \cup \{x\} \in I$ is independent. The subsequent operation of GREEDY will find a maximum-weight independent subset for M', and the overall operation of GREEDY will find a maximum-weight independent subset for M.

We will present various applications in the next issue of *Sutra*.

References

Akerkar, R. A. and Akerkar, R. R. *Discrete Mathematics*, Pearson Education (India), 2004.

Baase, S., Gelder, A. *Computer Algorithms.* Addison Wesley, 2000.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. *Introduction to algorithms, 2nd edition*. The MIT Press, 2001.

Oxley, J. G.: "Matroid Theory", Oxford University Press, New York, 1992.