ANNALES UMCS INFORMATICA DOI: 10.17951/AI.2016.16.2.48

Matroids And Greedy Algorithms

A Deeper Justification of Using Greedy Approach To Find A Maximal set of a Matroid

Syed Aqib Haider
Zakir Husain College of Eng. & Tech.
Aligarh Muslim University, Aligarh.
Aligarh, Uttar Pradesh, India
e-mail: aqibsyed443@gmail.com

***Abstract* -- Greedy algorithms are used in solving a diverse set of problems in small computation time. However, for solving problems using greedy approach, it must be proved that the greedy strategy applies. The greedy approach relies on selection of optimal choice at a local level reducing the problem to a single sub problem, which actually leads to a globally optimal solution. Finding a maximal set from the independent set of a matroid M(S, I) also uses greedy approach and justification is also provided in standard literature (e.g. Introduction to Algorithms by Cormen et .al.). However, the justification does not clearly explain the equivalence of using greedy algorithm and contraction of M by the selected element. This paper thus attempts to give a lucid explanation of the fact that the greedy algorithm is equivalent to reducing the Matroid into its contraction by selected element. This approach also provides motivation for research on the selection of the test used in algorithm which might lead to smaller computation time of the algorithm.**

Keywords -- Contraction, Greedy , Independence Matroid, Maximal

I. Introduction

Many problems in various areas of engineering are solved using a greedy approach. A greedy approach to solve a problem refers to making a decision based upon what looks optimal at the moment and reduces the problem to a single subproblem. Although, making a locally optimal choice might lead to suboptimal solution to the original problem, in many cases, it may lead to an optimal one. Finding a maximal set out of the set of independent sets of a matroid is one such problem and use of a greedy algorithm to solve the problem is well known and universally accepted. This paper tries to explain some important underpinnings of the justification for using greedy algorithm for dining matroid set. Based on the justification, it also provides motivation for finding a modified algorithm for smaller computation time.

II. The Matroid Theory And Greedy Approach

This section will given formal definition of matroid theory, the problem of finding a maximal set and the greedy algorithm that is used for solving it. Matroid: A matroid M (S, I) is an ordered pair of two sets S (which must be finite) and I if and only if I ≠ ϕ and I is a nonempty set of some subsets of S such that if $B \in I$ then A ∈ I for all A⊆B. This property is known as hereditary property of I. The elements of I are known as independent subsets of S.

If A,B ∈ I and $|A| < |B|$, there exists at least one element x ∈ B such that A U {x} ∈ I. This is known as exchange property.

Extension of A: Any element x ϵ S is known as extension of A (x ∉ A and A ∈ I) if and only if A U {x} ∈ I. Maximal set of M A set A ∈ I is maximal if it has no extensions. Firstly, we observe that all maximal sets of matroid M(S,I) are of same size.

Proof: Let A ∈ I ,B ∈ I, $|A| < |B|$ A and B are maximal.

A U {x} ∈ I for at least are element x ∈ B ⇒ A is not maximal.

Hence we arrive at a contradiction. Thus the theorem is proved.

Weighted matroid: A matroid M is said to be weighted if every elements x ∈ S is assigned a positive weight which extends as summation i.e. w $(A)= \sum w(x)$ where A ⊆ S.

Many problems in diverse areas of engineering may be reduced to finding an independent set of maximum weight of a matroid. For example, weight the problem of finding a minimum length tree of a graph (minimum spanning time problem) can be easily formulated as the above problem.

Now, we observe that any independent set of maximum weight must be maximal because all the weights are positive and any suboptimal set may be modified by adding its extension to it, thus increasing the weight.

Now we give the greedy algorithm MAXWEIGHT which takes a matroid M(S,I) as input and returns an independent maximum weight subset of S.

MAXWEIGHT (M, w)

1.A = ϕ

2.Sort elements of S in monotonically decreasing order

3.For each element of x ∈ S if A ∪ {x} ∈ I A = A ∪ {x}

4.return A

The above algorithm return an optimal solution.

III. Validity Of Maxweight

The section gives the formal proof that is provided in support of the above algorithm in standard literature. We make the following observations which will be used.

1. If {x} ∉ I, then x ∉ A for all A ∈ I

   To prove these, suppose other wise, i.e. {x} ∉ S and A ∈ I such that x ∈ A

   ⇒{x} ⊆ A

   ⇒ {x} ∈ I by hereditary property.

   ⇒ Contradiction

   Hence the theorem is proved.

2. If M (S, I) is a weighted matroid with S sorted into monotonically decreasing order by weight, then if x is the first element such the {x} ∈ I (if such an element exist), there exists an optimal subst A of S such that x ∈ A (the optimal refers to maximum weight independent subset).

   This is typically known as optimal substructure property.

   Proof: Let B be a nonempty optimal set. If x ∈ B, the theorem is true.

   If x ∉ B, let A = {x}.

   Till $|A| < |B|$ we can add some y ∈ B such that A = A U {y}

   So at a point $|A| = |B|$, such that A and B have $|A|$-1 same elements such that x ∈ A, x ∉ B, z ∈ B, z ∉ A for some w (z) ≤ w (x).

   Because z ∈ B ⇒ w (z) ≤ w (x) as x is heaviest independent element of S.

   ⇒A = B –{z} U {x} w(A) =w(B) +w(x)-w(z)

   ⇒w(A) = w (B) + w(x) – w (z) ≥ w(s)

   ⇒ w (A) is optimal.

   Hence the theorem is proved.

3. Matroids exhibited optimal subtracture property. If we select maximum weight element x ε S such the {x} ε I, their remaining problem is to find an optimal subset of matroid M' (S', I') such that 1. S' = { y : x ∈ S and {x,y} ∈ I} I' = { B ⊆ S – {x} : B U {x} ∈ I }. M' is known as contraction of M by x.

   The basis for verification of MAXWEIGHT is that the element passed over by MAXWEIGHT can never be included in any independent subset (by 1). Thus, after selecting x, the problem is reduced to applying the same algorithm on contraction of M by x because B is independent is M' if and only if B U {x} is independent is M.

This is an overview of the basic understanding of applying a greedy algorithm for solving this problem. However, this explanation does not clearly indicate the equivalence of MAXWEIGHT and contraction of M Specifically, the objective of section IV of the paper is to prove that at each iteration of MAXWEIGHT, further iteration is equivalent to applying MAXWEIGHT on the contraction of M by x where x is the element selected in the latest loop iteration. For example, it disproves the assumption that there may be some y ε S such the {x,y} ∈ I for which A U {y} ∉ I and thus, y should have been selected but the algorithm would not select it. This is not usually explained in standard literature and is the core motivation of this producing this paper.

IV. Equivalence Of Maxweight And Contraction Of Matroid

We will now try to prove that ,at every iteration, the element selected is an element of contraction of M by previous element and that every element of S of current reduced matroid is considered.

Proof : At every step A is selected if A U {x} ∈ I. Let the loop run N time and let xk denote element selected at the kth iteration such that optimal set formed finally is A = {x1 x2.. xN} Now xn is selected if AU {xn} ε I, where A = {x1, x2..xn-1} Assuming that till (k-1)th iteration, all elements selected were partt of corresponding contraction, i.e. {xi-1,xi} ε Ii-1 for all i = 2. k-1 for i =2 A = {x1} ⇒A U {x1} ε I if and only if {x1,x2} ∈ I

 assumption is true for i=2.
At kth iteration,
A = {x1, x2… xk-1} ε I
⇔ A2 = A – {x1} ε I (by definition of contraction)
⇔ A3 = A2 – {x2} ε I3 and so on upto
⇔ Ak-1 = Ak-2 – {xk-2} ε Ik-1
⇔{xk-1 , xk} e Ik-1
⇔{xk} εIk
This proves that for any element
x ε S , A U {xn} ε I is equivalent to
{xn} ε In or (xx-1, xx} ε Ik-1.
Therefore we can rewrite the algorithm as
MAXWEIGHT (M, w)
1.A =
2.Sort S in monotonically decreasing order
3.For every element x ε S
if {x} ε I
A = AUI {x}
prev = x
M(S, I) = M' (S', I') where M'(S',I') is contraction of M (S, I) by prev.
4.Return A
The results of both the algorithms will be same

V. Conclusion

Thus both forms of the algorithm are equivalent. The running time of both the algorithm is O(nlgn+nf(n)) where (f(n)) is the asymptotic time taken for test, be it A U {x} ε I or {x} ε I.

Thus, if in any problem, the computation of test {x} ε I, takes lesser time, the algorithm claimed in the paper might give better result in terms of the running time. Also the paper gives a clear explanation of the validity of using greedy approach in finding a maximum weight maximal independent set of a matroid. Thus, further scope of research may lie towards finding the test which takes lesser time to check independence of the set containing element being considered at evey loop iteration in this greedy approach.

VI. References

[1] "Introduction to Algorithms. Third Edition." Cormen,Leiserson,Rivest and Stein.

[2] B. Korte and L. Lovasz, "Mathematical structures underlying greedy algorithms". in F. Gecseg, editor, Fundamentals of Computation Theory,volume 117 of Lecture Notes in Computer Science, pages 205–209. Springer, 1981.

[3] B. Korte and L. Lovasz, "Structural properties of greedoids.", Combinatorica, 3(3–4):359–374, 1983

[4] B. Korte and L. Lovasz, "Greedoids - A structural framework for the greedy algorithm.". In W. Pulleybank, editor, Progress in Combinatorial Optimization, pages 221– 243. Academic Press, 1984.