Theory of Greedy Algorithms

Andreas Klappenecker

Greedy algorithms aim to solve a combinatorial optimization problem by successively adding elements to a set with the goal to construct a set of highest possible weight, assuming a maximization problem. The greedy strategy is simple: The algorithm always seeks to add the element with highest possible weight available at the time of selection that does not violate the structure of an optimal solution in an obvious way. Kruskals minimum spanning tree algorithm is an example of a greedy algorithm.

In these notes, we briefly discuss the basic principles underlying many greedy algorithms. For the proofs, the reader should refer to the references.

Notation. If E is a set, then the set consisting of all subsets of E is called the power set of E. The power set of E is denoted by 2^E. We denote by $A \setminus B$ the set difference of A and B, *i.e.*, $A \setminus B = \{x \in A \,|\, x \notin B\}$.

Matroids and Beyond. A **set system** is a pair $M = (E, S)$ consisting of a finite set E and a subset S of 2^E. A set system $M = (E, S)$ is called a **matroid** if and only if it satisfies

M1. $\emptyset \in S$;
M2. If Y in S and $X \subseteq Y$, then X in S;
M3. If X, Y are in S and $|X| > |Y|$, then there is an x in $X \setminus Y$ such that $Y \cup \{x\}$ in S.

A set system satisfying **M2** is called **hereditary**. The axiom **M3** is sometimes called the **exchange axiom**.

Example 1. Let E denote the set of columns of a matrix A over the field $\mathbf{R}$ of real numbers. We define S by

$$S = \{F \subseteq E \,|\, \text{the columns in } F \text{ are linearly independent over } \mathbf{R}\}.$$

Then the set system (E, S) is a matroid. Indeed, the properties **M1** and **M2** are obvious, and **M3** is a consequence of the Steiniz exchange theorem. This matrix example is the eponym for the term matroid.

Example 2. Let E denote the set of edges of some undirected graph $G = (V, E)$. We define S by

$$S = \{F \subseteq E \,|\, (V, F) \text{ is a forest}\}.$$

Then (E, S) is a matroid, called the graphical matroid of the graph G.

We can formulate greedy algorithms based on matroids. The graphical matroid underlies Kruskal's minimum spanning tree algorithm, as we will see. However, there are greedy algorithms that cannot be formulated with the help of matroids. For example, Dijkstra's single source shortest path algorithm.

Korte and Lovasz observed that in many cases the second axiom of a matroid is not needed, and introduced the notion of a greedoid. A **greedoid** is a set system (E, S) satisfying **M1** and **M3**.

A directed graph is called **connected** if and only if the underlying undirected graph is connected. A directed graph is called a **branching** if and only if the underlying undirected graph is a forest and each vertex has at most one entering edge. A connected branching is called an **arborescence**. An arborescence with n vertices has $n-1$ edges. Thus, there exists one vertex without incoming edge; this vertex is called the root of the arborescence.

Proposition 3. *Let E be the edge set of a directed graph $G = (V, E)$. Let r be a vertex in V. Define*

$$S = \{F \subseteq E \,|\, (V, F) \text{ is an arborescence rooted at } r\}.$$

Then (E, S) is a greedoid, called the directed branching greedoid of G.

Proof. It is clear that **M1** holds. If (V_1, F_1) and (V_2, F_2) are arborescences in G rooted at r with $|F_1| > |F_2|$, then $|V_1| = |F_1| + 1 > |F_2| + 1 = |V_2|$. Let x be a vertex in $V_1 \setminus V_2$. Then the path from r to x in (V_1, F_1) contains an edge (v, w) with v in V_2 such that w is not in V_2. This edge can be added to (V_2, F_2) so that $F_2 \cup \{(v, w)\}$ is in S. Therefore, **M3** holds as well. □

Sometimes even more general set systems are considered. A set system (E, S) satisfying

A1. $\emptyset \in S$;

A2. For every nonempty set X in S there exists an element x in X such that $X \setminus \{x\}$ is contained in S.

is called an **accessible set system**.

Exercise 4. Show that a greedoid is an accessible set system.

In a greedy algorithm, we want to add one element at a time, so accessible set systems are the most general set systems that can be used for greedy algorithms. The greedy algorithm works for all matroids, but unfortuntately not for all greedoids or accessible set systems. However, one can characterize the greedoids and the accessible set systems for which a greedy algorithm works.

Let $M = (S, E)$ an accessible set system. The elements of S are called the **feasible sets** of M. If M is a matroid, then the feasible sets are also known as **independent sets**. A maximal feasible set is called **basis**; in other words, a basis is a feasible set that is not properly contained in another feasible set.

Exercise 5. Show that in a greedoid all bases have the same cardinality.

Weight Functions. Let $M = (E, S)$ be an accessible set system. A **linear weight function** on M is a function $w \colon E \to \mathbf{R}$ that is, by abuse of notation, extended to S by defining the weight $w(X)$ of a feasible set X in S by

$$w(X) = \sum_{x \in X} w(x).$$

We say that a weight function w is positive if and only if $w(x) > 0$ for all x in E.

For an accessible set system $M = (E, S)$ with weight function $w\colon E \to \mathbf{R}$, we can consider the optimization problem

$$\text{Maximize } w(B) \text{ for all bases } B \text{ of } M. \tag{1}$$

Greedy Algorithm for Matroids. Let $M = (E, S)$ be ~~an accessible set system~~ a matroid and w a positive linear weight function on M. The greedy algorithm for (M, w) is given by

```
Greedy(E, S, w)
  T := ∅;
  sort E in monotonically decreasing order by weight w;
for x in E taken in monotonically decreasing order do
  if T ∪ {x} in S  then  T := T ∪ {x}; fi;
od;
return T;
```

We call an element of S **optimal** if it has maximal weight. Since we use positive weight functions, an optimal set is a basis.

Proposition 6 (Matroids exhibit the Greedy Choice Property)**.** *Suppose that $M = (E, S)$ is a matroid with positive linear weight function w and that E is sorted into monotonically decreasing order by weight. Let x be the first element of E such that $\{x\}$ is an independent set, if any such x exists. If x exists, then there exists an optimal set in S that contains x.*

Proof. See [Cormen, Leiserson, Rivest, Stein, *Introduction to Algorithms*, 2nd edition, Lemma 16.7]. □

Proposition 7. *Let $M = (E, S)$ be a matroid. If x is an element of E such that $\{x\} \notin S$, then $A \cup \{x\} \notin S$ for all A in S.*

Proof. Seeking a contradiction, we assume that $A \cup \{x\} \in S$. However, this would imply that $\{x\}$ in S by **M2**, contradicting our assumption $\{x\} \notin S$. □

Let $M = (E, S)$ be a matroid. Let x be an element in E such that $\{x\}$ is contained in S. We define

$$\begin{aligned} E_x &= \{y \in E \mid \{x, y\} \in S\}, \\ S_x &= \{B \subset E \setminus \{x\} \mid B \cup \{x\} \in S\}. \end{aligned}$$

Then (E_x, S_x) is a matroid called the **contraction** of M by x.

Proposition 8 (Matroids exhibit the Optimal Substructure Property)**.** *Let x be the first element of E chosen for the matroid $M = (E, S)$. The remaining problem of finding a maximum-weight independent subset containing x reduces to finding a maximum-weight independent subset of the contraction M_x of M by x.*

Proof. See [Cormen, Leiserson, Rivest, Stein, *Introduction to Algorithms*, 2nd edition, Lemma 16.10]. □

Theorem 9. *If $M = (E, S)$ is a matroid with positive linear weight function, then Greedy(M,w) returns an optimal subset.*

Beyond Matroids. Let us now consider the greedy algorithm for more general accessible set systems. Unfortunately, it does not make sense to use the previous version of the greedy algorithm for accessible set systems that are not hereditary. Indeed, it might happen that an element x cannot be added to the current feasible set T, but later it might be allowed to add x to a newly created feasible set, say, $T \cup A$. If $w(x) > w(y)$ for some y in A, then the previous algorithm would be incorrect.

Let $M = (E, S)$ be an accessible set system and w a linear weight function on M. The greedy algorithm for (M, w) is given by

Greedy2(E, S, w)
  $T := \emptyset$;
  $X := E$;
while there exists x in X such that $T \cup \{x\}$ in S **do**
  choose x in X satisfying $T \cup \{x\}$ in S such that
  $w(x) \geq w(y)$ for all y in X with $T \cup \{y\}$ in S;
  $T := T \cup \{x\}$;
  $X := X \setminus \{x\}$;
od;
return T;

As I have mentioned earlier, this algorithm does not work correctly for all accessible set systems, not even for all greedoids. The next theorem characterizes the greeoids for which the above greedy algorithm is correct for any weight function.

Theorem 10. *Let $M = (E, S)$ be a greedoid. The algorithm Greedy2 produces for each weight function an optimal solution if and only if M satisfies the following axiom:*

SE1. *For J, K in S with $|J| = |K| + 1$, there exists an element a in $J \setminus K$ such that $K \cup \{a\}$ and $J \setminus \{a\}$ are contained in S.*

More generally, we can ask for which accessible set systems does the Greedy2 algorithm always solve the optimization problem for any positive linear weight function? This problem was solved in [P. Helman, B.M. Mont, H.D. Shapiro, *An Exact Characterization of Greedy Structures*, SIAM J. Discr. Math, 1993]. In this case, the accessible set system must be embeddable into a matroid and satisfy some additional constraints. After you have gained some familiarity with matroids and greedoids, you can easily read the above paper.