Greedy Algorithms and Matroids

Andreas Klappenecker

Greedy Algorithms

A greedy algorithm solves an optimization problem by working in several phases.

In each phase, a decision is made that is locally optimal given the information that has been obtained so far. This decision is made without regard for future consequences.

This greedy "take what you can get now" strategy is explains the name for this class of algorithms.

Correct Greedy Algorithms

When a greedy algorithm terminates, then the hope is that the greedy choices in each phase lead to a global optimum of the optimization problem. If a global optimum is always reached, then the algorithm is correct.

Properties

A greedy algorithm successively solves subproblems of the optimization problem.

An optimization problem has optimal substructure if and only if an optimal solution to the problem contains within it optimal solutions to subproblems.

Correct greedy algorithms require that the problem has the optimal substructure property.

Properties

The greedy choice property is that a globally optimal solution can be arrived at by making a series of locally optimal choices.

A correctness of a greedy algorithm will in general rely on the optimal substructure and greedy choice property.

Greedy Algorithms

Greedy algorithms are easily designed, but correctness of the algorithm is harder to show.

We will look at some general principles that allow one to prove that the greedy algorithm is correct.

Matroids

Matroid

Let S be a finite set, and F a nonempty family of subsets of S, that is, $F \subseteq P(S)$.

We call (S,F) a matroid if and only if

M1) If $B \in F$ and $A \subseteq B$, then $A \in F$.

[The family F is called hereditary]

M2) If $A,B \in F$ and $|A|<|B|$, then there exists x in $B \setminus A$ such that $A \cup \{x\}$ in F

[This is called the exchange property]

Matric Matroids

Let M be a matrix.

Let S be the set of rows of M and

F = { A | A⊆S, A is linearly independent }

Claim: (S,F) is a matroid.

Clearly, F is not empty (it contains every row of M).

M1) If B is a set of linearly independent rows of M, then any subset A of B is linearly independent. Thus, F is hereditary.

M2) If A, B are sets of linearly independent rows of M, and |A|<|B|, then dim span A < dim span B. Choose a row x in B that is not contained in span A. Then A∪ {x} is a linearly independent subset of rows of M.Therefore, F satisfies the exchange property.

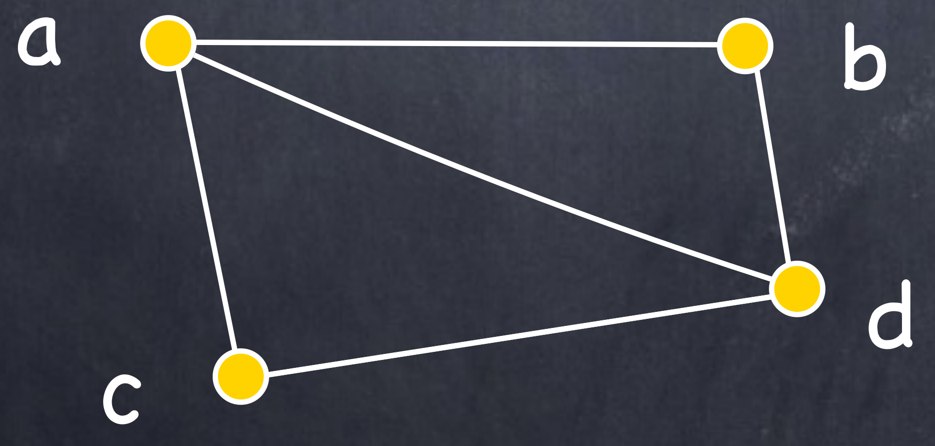
Undirected Graphs

Let V be a finite set,

E a subset of $\{ e \mid e \subseteq V, |e|=2 \}$

Then (V,E) is called an undirected graph.

We call V the set of vertices and E the set of edges of the graph.



$V = \{a,b,c,d\}$

$E = \{ \{a,b\}, \{a,c\}, \{a,d\}, \{b,d\}, \{c,d\} \}$

Induced Subgraphs

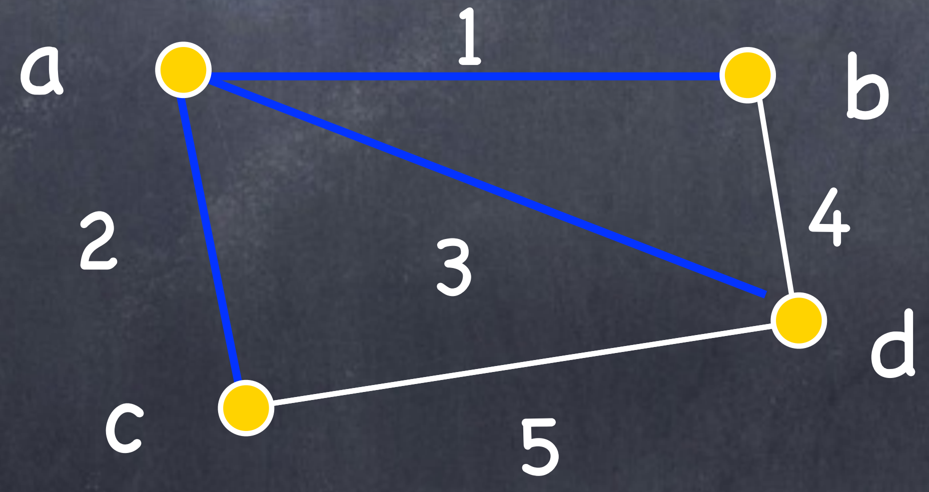
Let G=(V,E) be a graph.

We call a graph (V,E') an induced subgraph of G if and only if its edge set E' is a subset of E.

Thus, an induced subgraph of G=(V,E) has the same vertex set V, and no edges that are not already contained in E.

Spanning Trees

Given a connected graph G, a spanning tree of G is an induced subgraph of G that happens to be a tree and connects all vertices of G. If the edges are weighted, then a spanning tree of G with minimum weight is called a minimum spanning tree.



Graphic Matroids

Let G=(V,E) be an undirected graph.

Choose S = E and F = { A | H = (V,A) is an induced subgraph of G such that H is a forest }.

Claim: (S,F) is a matroid.

M1) F is a nonempty hereditary set system.

M2) Let A and B in F with |A| < |B|. Then (V,B) has fewer trees than (V,A). Therefore, (V,B) must contain a tree T whose vertices are in different trees in the forest (V,A). One can add the edge x connecting the two different trees to A and obtain another forest (V,A∪{x}).

Weight Functions

A matroid (S,F) is called weighted if it equipped with a weight function w: S->$\mathbf{R}^+$, i.e., all weights are nonnegative real numbers.

If A is a subset of S, then

$$w(A) := \Sigma_{a \text{ in } A}\, w(a).$$

Weight functions of this form are sometimes called "linear" weight functions.

Greedy Algorithm for Matroids

```
Greedy(M=(S,F),w)    // maximizing version
A := ∅;
Sort S into monotonically decreasing order by weight w.
for each x in S taken in monotonically decreasing order do
   if A∪{x} in F then A := A∪{x}; fi;
od;
return A;
```

Correctness

Theorem: Let M= (S,F) be a weighted matroid with weight function w. Then Greedy(M,w) returns a set in F of maximal weight.

[Thus, even though Greedy algorithms in general do not produce optimal results, the greedy algorithm for matroids does! This algorithm is applicable for a wide class of problems. Yet, the correctness proof for Greedy is not more difficult than the correctness for special instance such as Huffman coding. This is economy of thought!]

Correctness

- Seeking a contradiction, suppose that Greedy returns C in F, but there exists B in F such that $w(B)>w(C)$.
- Since w is nonnegative, we can assume that B,C are bases; say $B=\{b_1,\dots,b_n\}$ and $C=\{c_1,\dots,c_n\}$.
- Let i be the smallest index such that $w(b_1)<=w(c_1),\dots,w(b_{i-1})<=w(c_{i-1})$, and $w(b_i)>w(c_i)$.
- Greedy would have picked b_i in the ith step . Indeed, $\{b_1,\dots,b_i\}$ and $\{c_1,\dots,c_{i-1}\}$ are in F, hence $\{c_1,\dots,c_{i-1,}b_i\}$ in F by the exchange axiom. Thus $w(B)<=W(C)$, contradiction!!!

Complexity

Let $n = |S|$ = # elements in the set S.

Sorting of S: $O(n \log n)$

The for-loop iterates n times. In the body of the loop one needs to check whether $A \cup \{x\}$ is in F. If each check takes $f(n)$ time, then the loop takes $O(n\, f(n))$ time.

Thus, Greedy takes $O(n \log n + n\, f(n))$ time.

Minimizing or Maximizing?

Let $M=(S,F)$ be a matroid.

The algorithm Greedy(M,w) returns a set A in F maximizing the weight $w(A)$.

If we would like to find a set A in F with minimal weight, then we can use Greedy with weight function

$w'(a) = m-w(a)$ for a in A,

where m is a real number such that $m > \max_{s \text{ in } S} w(s)$.

Matric Matroids

Let M be a matrix. Let S be the set of rows of the matrix M and

$F = \{ A \mid A \subseteq S, A \text{ is linearly independent } \}$.

Weight function $w(A)=|A|$.

What does Greedy((S,F),w) compute?

The algorithm yields a basis of the vector space spanned by the rows of the matrix M.

Graphic Matroids

Let G=(V,E) be an undirected connected graph.

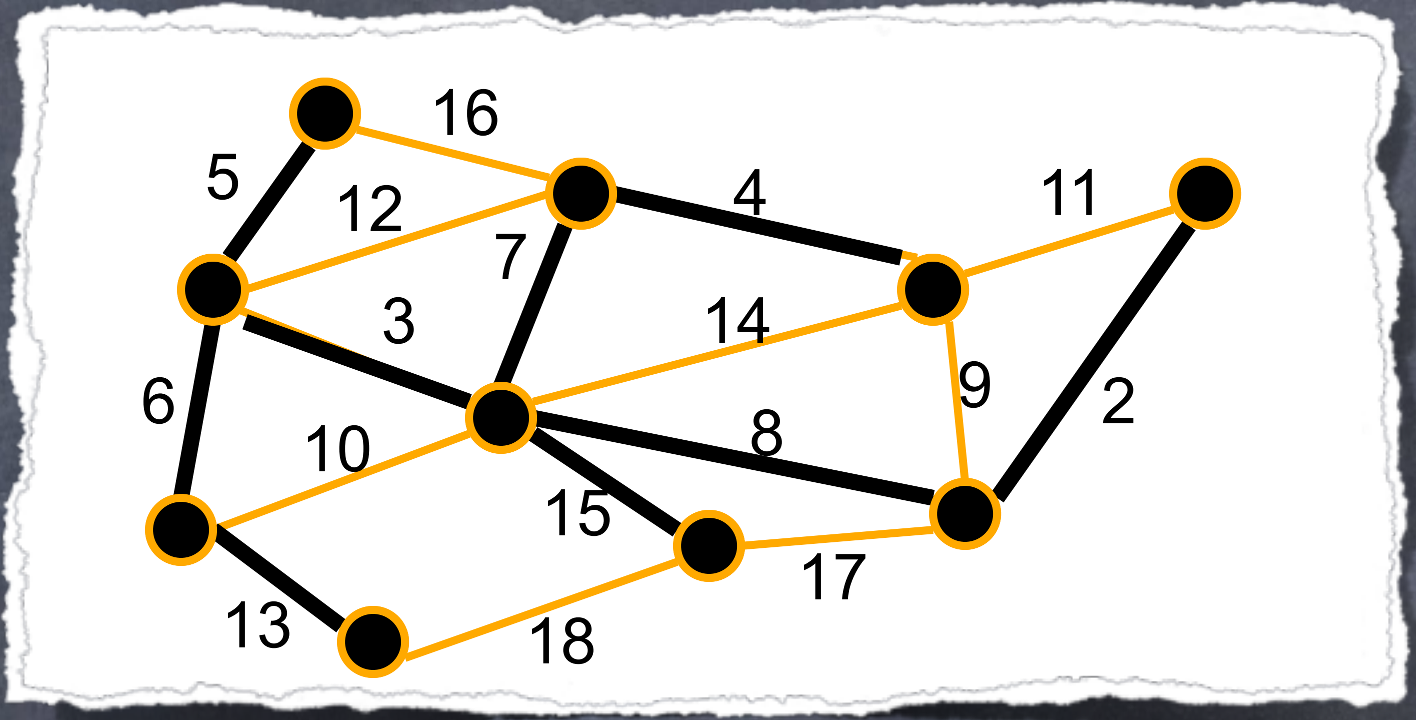
Let S = E and F = { A | H = (S,A) is an induced subgraph of G such that H is a forest }.

Let w be a weight function on E.

Define w'(a)=m-w(a), where m>w(a), for all a in A.

Greedy((S,F), w') returns a minimum spanning tree of G. This algorithm in known as Kruskal's algorithm.

Kruskal's MST algorithm



Consider the edges in increasing order of weight, add an edge iff it does not cause a cycle

[Animation taken from Prof. Welch's lecture notes]

Conclusion

Matroids characterize a group of problems for which the greedy algorithm yields an optimal solution.

Kruskals minimum spanning tree algorithm fits nicely into this framework.