Kapitel 1: Greedy Algorithms

Inhalt:

- Intervall Scheduling
  - Greedy stays ahead
- Scheduling to Minimize Lateness
  - Exchange Argument
- Theoretical Foundations of the Greedy Method
  - Matroid Theory

Matroid

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

We call $M=(S,I)$ a matroid if and only if

1) If $B \in I$ and $A \subseteq B$, then $A \in I$.

   [The family I is called hereditary]

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

   [This is called the exchange property]

Example 1 (Matrix Matroid)

Let M be a matrix.
Let S be the set of columns of M and
$I = \{ A \mid A \subseteq S, A \text{ is linearly independent} \}$

Claim: (S,I) is a matroid.

Clearly, I is not empty.

1) If B is a set of linearly independent columns of M, then any subset A of B is linearly independent. Thus, I is hereditary.
2) If A, B are sets of linearly independent columns of M, and $|A|<|B|$, then dim span A < dim span B. Choose a column x in B that is not contained in span A. Then $A \cup \{x\}$ is a linearly independent subset of columns of M. Therefore, I satisfied the exchange property.

Example 2 (Graphic Matroid)

Let $G=(V,E)$ be a connected, undirected graph.
Choose $S = E$ and
$I = \{ A \mid H = (V,A)$ is an induced subgraph of G such that H is a forest (i.e. H is acyclic) $\}$.

Claim: (S,I) is a matroid.

1) I is a nonempty hereditary set system.
2) Let A and B in I 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 e connecting the two different trees in A and obtain another forest $(V,A\cup\{e\})$. Therefore, I satisfied the exchange property.

Counter example (Interval scheduling)

Let S={1, … , n} be a set of jobs. Job j starts at s_j and finishes at f_j. Two jobs are compatible if they don't overlap.

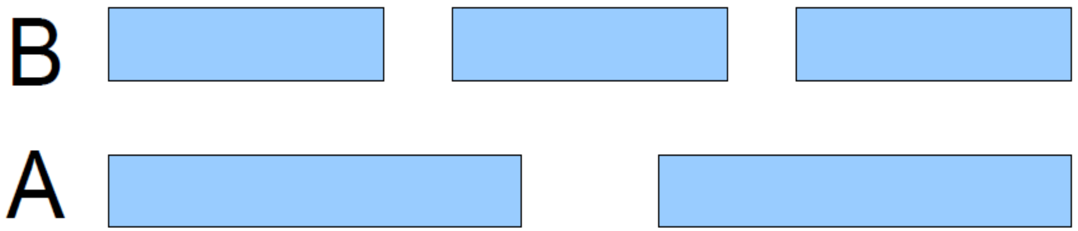
U $\subseteq$ S belongs to I if its requests are mutually compatible

(i.e. $i,j \in U \Rightarrow [s_i, f_i) \cap [s_j, f_j) = \emptyset$)

Claim: (S,I) is not a matroid.

1) I is a nonempty hereditary set system.

2)



Therefore, I satisfied not the exchange property.

Maximal Independent Subset

Let $M=(S,I)$ be a matroid. $A \in I$ is called maximal, if there exists no $x \in S \setminus A$ such that $A \cup \{x\} \in I$.

Lemma: All maximal independent subsets in a matroid have the same size.

Example 1) Matrix Matroid

A is maximal $\Leftrightarrow$ $|A|$ = rank(S)

Example 2) Graphic Matroid

A is maximal $\Leftrightarrow$ $|A| = |V| - 1$

$\Leftrightarrow$ A is a spanning tree of G

Weighted Matroid

A matroid $M=(S,I)$ is called weighted if there is a weight function $w:S \rightarrow R^+$, i.e. w assigns a weight $w(x) > 0$ to each element $x \in S$.

For $A \subseteq S$ we set $w(A) := \sum_{x \in A} w(x)$.

The maximum weight independent subset problem:
Find a maximum weight independent subset in a weighted matroid.

(Remark: Since $w(x) > 0$ for all $x \in S$, a maximum weight independent subset is also a maximal independent subset.)

Greedy Algorithm for Matroids

Greedy($M=(S,I),w$)

$A := \emptyset$;

Sort S into non increasing order by weight w.

for each $x \in S$ taken in non increasing order **do**

    if $A \cup \{x\} \in I$ **then** $A := A \cup \{x\}$; **fi**;

od;

return A;

Matrix Matroid

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

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

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

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

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

Minimizing or Maximizing?

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

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

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

$$w'(x) = m-w(x) \qquad \text{for } x \in S,$$

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

Graphic Matroids

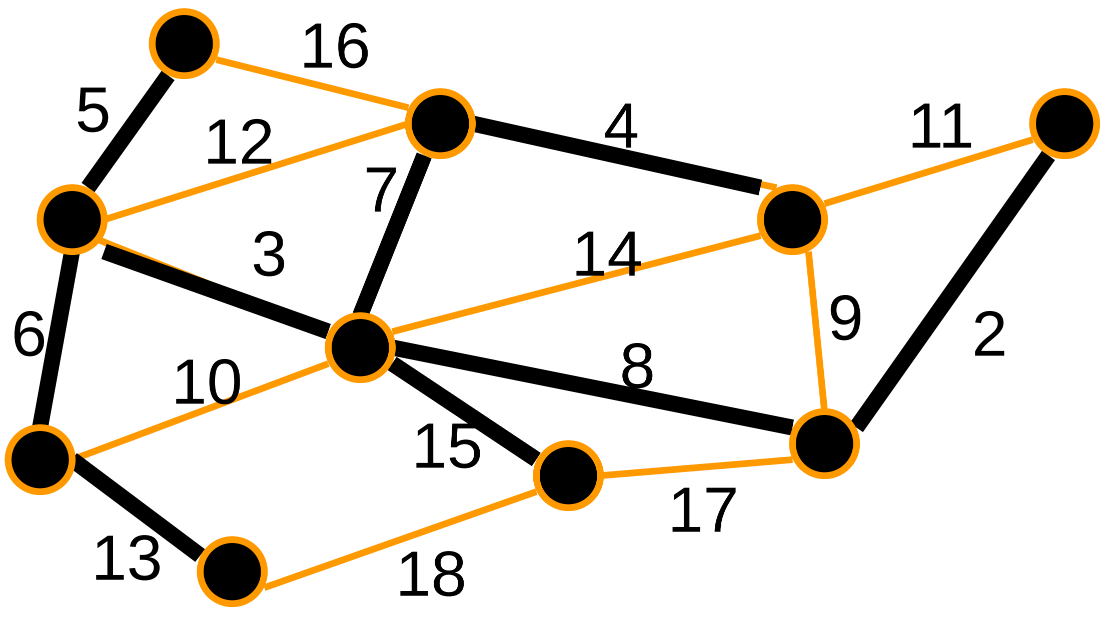
Let G=(V,E) be an undirected connected graph.
Let S = E and I = { A | H = (V,A) is an induced subgraph of G such that H is a forest }.

Let w be a weight function on E.
Define w'(x)=m-w(x), where m>w(x), for all $x \in S$.
It holds w'(x) > 0 for all $x \in S$.

Greedy((S,I), 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

Greedy Algorithm for Matroids

Greedy(M=(S,I),w)

A := ∅;

Sort S into non increasing order by weight w.

for each x ∈ S taken in non increasing order **do**

    if A∪{x} ∈ I **then** A := A∪{x}; **fi**;

od;

return A;

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 independent or not. 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.

Correctness

- **Greedy as Algorithmic technique** can optimally solve an optimization problem,
  - if the **greedy algorithm stays ahead**. Show that after each step of the greedy algorithm, its solution is at least as good as any other algorithm's or
  - if we can find an **exchange argument**. Gradually transform any solution to the one found by the greedy algorithm without hurting its quality or
  - if the problem has the **greedy-choice property**. Prove that a globally-optimal solution can always be found by a series of local improvements from a starting configuration or
  - if we can define an adequate **Matroid** for the problem.

Correctness

Lemma: ("Greedy choice property")

Let M= (S,I) be a weighted matroid with weight function w and S is sorted into non increasing order by weight. Let x be the first element of S such that {x} is independent. Then there exists maximal subset A of S that contains x.

Theorem: ("Correctness")

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

Conclusion

Matroids characterize a group of problems for which the greedy algorithm yields an optimal solution. (“Sufficient Condition”)

Thus, even though Greedy algorithms in general do not produce optimal results, the greedy algorithm for matroids does!

Fragen?