MATROIDS

Definition:

Many problems that have use greedy algorithms can be understood through the concept of matroids, a mathematical structure that generalizes the idea of linear independence in vector spaces.

Matroids

A matroid (S, J) consists of:

- S is a finite set known as ground set.
- J is a collection of subsets of S called independent sets.

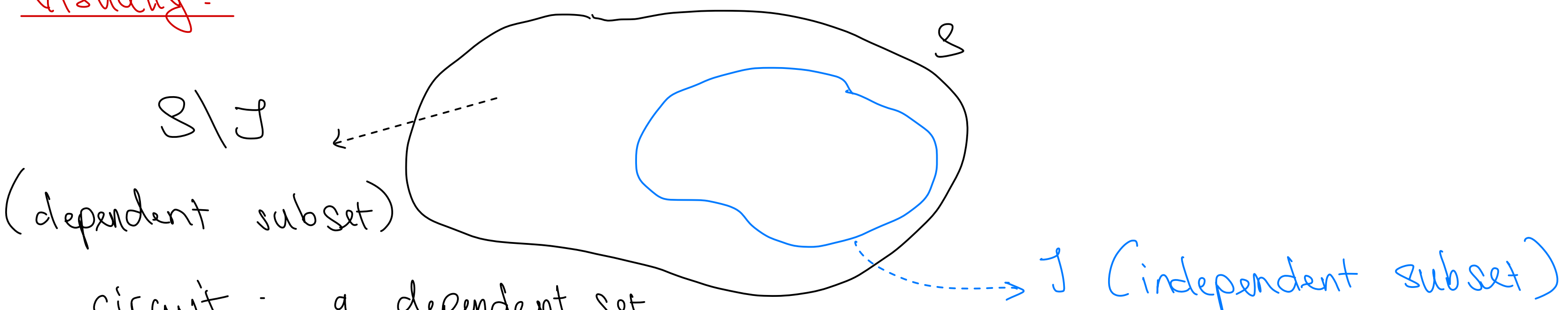
The independent sets must satisfy 3 properties:

- Non-emptiness: The empty set is included in J.
- Heredity: If a set X is in J, then every subset of X is also in J.
- Exchange: If two sets X and Y are in J, and $|X| > |Y|$, then there exists an element in X that can be added to Y while keeping it independent

Intuition:

- In linear algebra, a set of vectors are independent if no vector can be expressed a combination of others. Similarly, in matroid, an independent set cannot be formed by combining elements from dependent sets
- Greedy algorithms: works by selecting the best local choice at each step, adding the local optima to your solution set would still yield an independent set

Visually:



circuit : a dependent set

basis : largest possible independent subset

Examples of Matroid:

- **Linear Matroid:**

  Let A be any $n \times m$ matrix. Any subset $I \subseteq \{1, 2, 3, \dots n\}$ corresponding to columns $\{a_1, a_2, a_3, \dots, a_n\}$ is independent if and only if the colums are linearly independent.

  Proof: (in backward direction)

  Assume the corresponding columns are independent, then any subsets of the independent columns set are also independent. Denotes $I \subseteq \{1, 2, 3, \dots, n\}$ the indices set of the independent columns, it follows that every subsets of I is also an independent set, which satisfy the heredity property.

- **Graphic / cycle matroid $M(G)$:**

  Let $G = (V, E)$ be an arbitrary undirected graph. A subset of E is independent if it defines an acyclic subgraph of G.

  A basis in the graphic matroid is a spanning tree of G, a circuit in this matroid is a cycle in G.

Computing a basis of a weighted matroid

This pseudo code describe a natural greedy algorithm that computes a basis for any weighted matroid $(S, \mathcal{I})$, where the ground set S is represented by an array $S[1 \dots n]$, and the weights of ground elements are represented by another array $w[1 \dots n]$

```
GREEDY_BASIS ( S, w):
    sort S in decreasing order of weight w
    G = ∅
    for i: 1 → n :
        if G ∪ { S[i]} ∈ J :
            G << S[i]
    return G
```

$\Rightarrow$ Time Complexity: $O(n \log n + n \cdot F(n))$, where $F(n)$ denotes the time to check any subset $X \in \mathcal{I}$.

Proof: GREEDY_BASIS return maximum-weight basis of $M = (S, \mathcal{I})$, which is any matroid and any weight function w.

We prove by contradiction:

- Negation: Assume set G is not maximum-weight basis of M. And there exist set H s.t H's total weights is larger than G's total weights:

$$\sum_{i=1}^{k} w(g_i) < \sum_{i=1}^{l} w(h_i)$$

- Show contradiction:
  - WLOG, assume $|G| = |H|$ (or $k = l$).
  - Let i be the smallest index s.t. $w(g_i) < w(h_i)$
  - This means that up until element $(i-1)^{th}$: $w(g_j) \geq w(h_j)$
  - The exchange property says that, if $|G_{i-1}| < |H_i|$, then $\exists\ h_i \in H_i$ s.t. $|G_{i-1}| \cup h_i \in \mathcal{I}$.

    But $h_i \notin G_i$, meaning GREEDY_BASIS rejects h_i despite it being the heavier element, which is a contradiction.

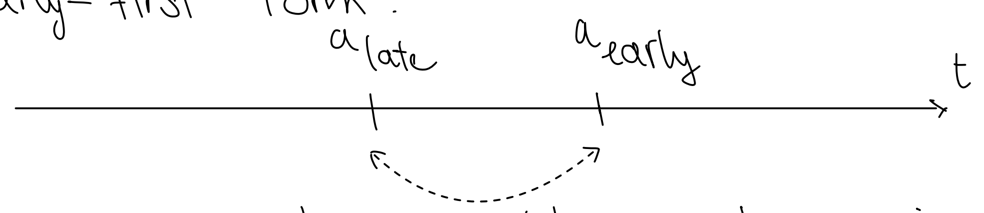
A task-scheduling problem as a matroid

Inputs:
- A finite set $S = \{a_1, a_2, \dots, a_n\}$ of unit-time tasks
- A corresponding deadlines $D = \{d_1, d_2, \dots, d_n\}$ $(1 \leq d_i \leq n)$
- A set of nonnegative penalties $P = \{p_1, p_2, \dots, p_n\}$

Goal: Find a schedule for S s.t. the total penalties for missed deadlines is minimized

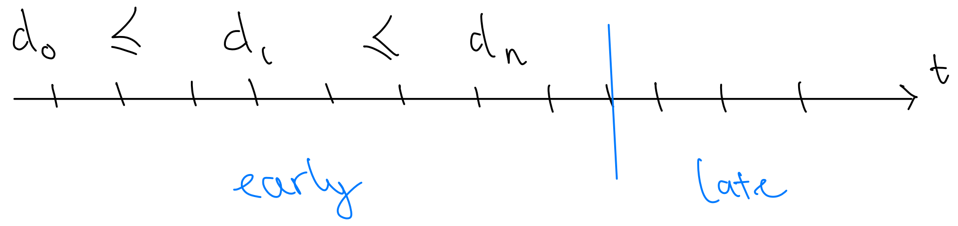
How to frame this as matroids?

- First, we recognize that, given an arbitrary schedule, we can always transform it into early-first form, and then canonical form.
  - early-first form:

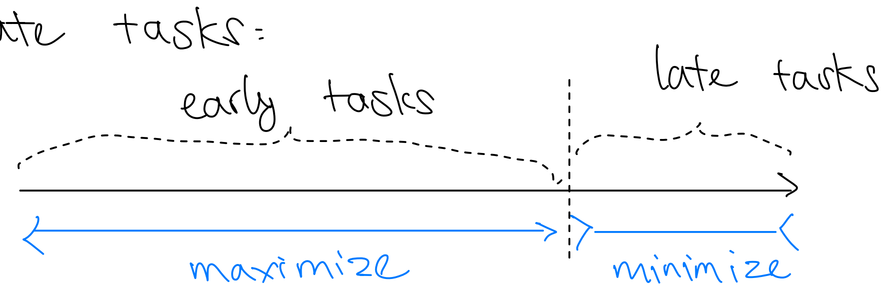


swapping these would not change its state.

  - canonical form:



- Second, given that we have transformed the schedule into canonical form, and we are trying to minimize the penalties in the late tasks:



$\Rightarrow$ We can reframe the problem as maximizing set A of early tasks

- Lastly, we define a set A task is independent if exist a schedule such that no tasks are late.

  Then we can frame this problem as finding basis of maximum-weight of any matroid $M(S, \mathcal{J})$ and weights w where:

  - S : ground set (finite set of tasks)
  - $\mathcal{J}$: set of independent subsets (set of possible schedules such that no tasks are late)
  - $w \Leftrightarrow p$: weights array (maximizing the early tasks will effectively minimizing the late tasks)

How to decide if a subset of tasks is "independent"?

Denote π as the schedule such that all tasks in subset A completed on time, we can say that:

- $\pi(a_t) \geq t$: time to complete task a_t must be at least t since the task is unit-time.
- $d_t \geq \pi(a_t)$: because the scheduled task a_t is on time

$\Rightarrow d_t \geq t$: the deadline of task at position t must be at least t

$\Rightarrow$ (Generalize for all tasks)

$|A(t)| \leq t$: for $A(t)$ tasks to be on time, its size must not exceed t. Otherwise, it would contradict the previously derived inequality

Construct a solution with greedy algorithm

Now that we have all the ingredients we need to construct a solution, let reuse the previous greedy pseudo code

```
GREEDY SCHEDULE ( D, P )
    Sort P in increasing order, permute D to match
    t = 1
    for i = 1 → n:
        S[t] = a_i
        if S[1 ... t] is "independent":
            t = t + 1
    return the canonical schedule for S[1...t]
```

As stated before, assume checking $S[1..t]$ is independent took $F(n)$, this algorithm will costs $O(n\log n + n.F(n))$

Proof that Schedule Deadlines is a matroids problem

We have "framed" the problem as matroids and provide solved it with greedy, but never actually proves it is a matroids. Lets do it now:

- Non-emptiness: an empty set $\emptyset$ can be considered independent because if there is no tasks to schedule, then no tasks are lated.
  $\Rightarrow \emptyset \in$
- Heredity: Let A be subset of independent tasks, it's easy to see that subset $C \subseteq A$ is also independent as:
  $$|C(t)| \leq |A(t)| \leq t \quad \text{where } 0 \leq t \leq n$$
- Exchange:
  . Let A and B be independent subsets of tasks where $|B| > |A|$.
  k be the largest timestamp s.t. $|B(k)| \leq |A(k)|$ ①
  . Since: $|B| > |A|$
  $$\Rightarrow |B(k)| + |B(j)| > |A(k)| + |A(j)| \quad ②$$
  where: $k+1 \leq j < n$
  ① & ② $\Rightarrow |B(j)| > |A(j)|$ where $k+1 \leq j < n$
  Therefore: exists at least 1 task in $B(j)$ where $k+1 \leq j < n$

- Let $\begin{cases} b_i \in B_{k+1 \leqslant j < n} & \text{with deadline } k+1 \\ A' = A \cup \{b_i\} \end{cases}$

We know show that A' is also independent:

- For $0 \leqslant t \leqslant k$, we know that:

  3 $\begin{cases} |A(t)| \leqslant t & \text{since } A \text{ is independent} \\ |A'(t)| = |A(t)| & \text{since the number of tasks with deadline} \end{cases}$

  $t < k$ in $A(t)$ won't change when we add b_i task with deadline $t = k+1$ to form $A'(t)$

- For $k+1 \leqslant t < n$, we know that:

  4 $\begin{cases} |B(t)| \leqslant t & \text{since } B \text{ is independent} \\ |A'(t)| \leqslant |B(t)| & \text{since } A' = A \cup \{b_i\} \text{ is at most} \end{cases}$

  the size of B if $B_{k+1 \leqslant j < n}$ only contains task b_i

3 & 4 $\Rightarrow$ A' is independent on $[0, n)$

Conclusion: If set S is a set of unit-time tasks with deadlines, and J is the set of all independent sets of tasks, then the corresponding system (S, J) is a matroid.

Decide if a subset is independent

- To decide if a subset is independent, we need to provide a bound on the subset's cardinality w.r.t its rank.
- Important properties of rank function:
  - For any subset X of ground set S, $0 \leqslant r(X) \leqslant |X|$
    $\Rightarrow$ The rank cannot exceeds the cardinality of the subset.
  - Independence condition: Subset X is independent i.o.i $r(X) = |X|$
    In other words, if $r(X) < |X|$, then X contains dependent elements
  - Monotonicity: If $X \subseteq Y$, then $r(X) \leqslant r(Y)$.
    In other words, if you consider a larger subset, its size can only stays the same or increase.

Basis of Matroid:

- The maximal independent sets, meaning they cannot be expanded further without losing independence.
- All bases share the same cardinality, called rank of the matroid
- Every independent set in matroid is contained within at least one basis.

Examples of Matroid Bases

- Linear Matroid:
  - Definition:
    Maximal linearly independent sets of vectors
  - Example:
    Vectors forming a basis in $\mathbb{R}^n$

- Graphic Matroid:
  - Definition:
    Maximal edge sets without cycles (Spanning forest)
  - Example:
    Edges forming a spanning tree in a graph

- Uniform Matroid:
  - Definition:
    All subsets of size k from the ground set
  - Example:
    Every k-element subset of $\{1, \dots, n\}$

- Partition Matroid:
  - Definition:
    Subsets containing exactly k_i elements from each partitioned category
  - Example:
    Choosing 2 edges from each graph component

Dual matroid definition

The dual matroid M^* of M has bases that are complements of the bases of M.
Formally, a set $B^* \subseteq S$ is a basis of M^* if and only if $S \setminus B^*$ is a basis of M.

Independent sets in M^*

A subset $T \subseteq S$ is independent in M^* if and only if T is contained in some basis of M^*. Since bases of M^* complements the bases of M, $S \setminus T$ contains at least one basis of M.

Basis exchange property ensure M^* is a valid matroid

- Uniform complement size: $|E| - r(M)$
- Exchange property in M directly translates to M^*

Basis Exchange Property

For any 2 distinct bases B_1 and B_2 in matroid M, and any element $e \in B_1 \setminus B_2$, there exists an element $f \in B_2 \setminus B_1$ such that:

$$(B_1 \setminus \{e\}) \cup \{f\} \quad \text{and} \quad (B_2 \setminus \{f\}) \cup \{e\}$$

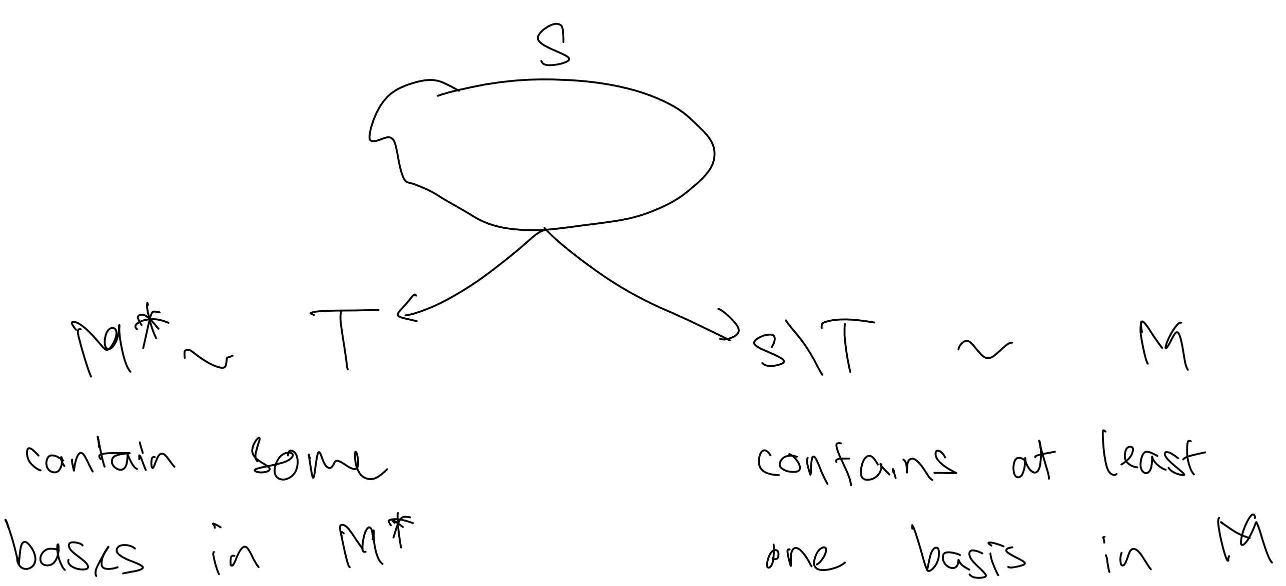
ar also a basis. This ensures that bases can be transformed into one another through element swaps while preserving independence.

Multiple Exchange:

For subsets $X \subseteq B_1 \setminus B_2$, there exists subsets $Y \subseteq B_2 \setminus B_1$ s.t.

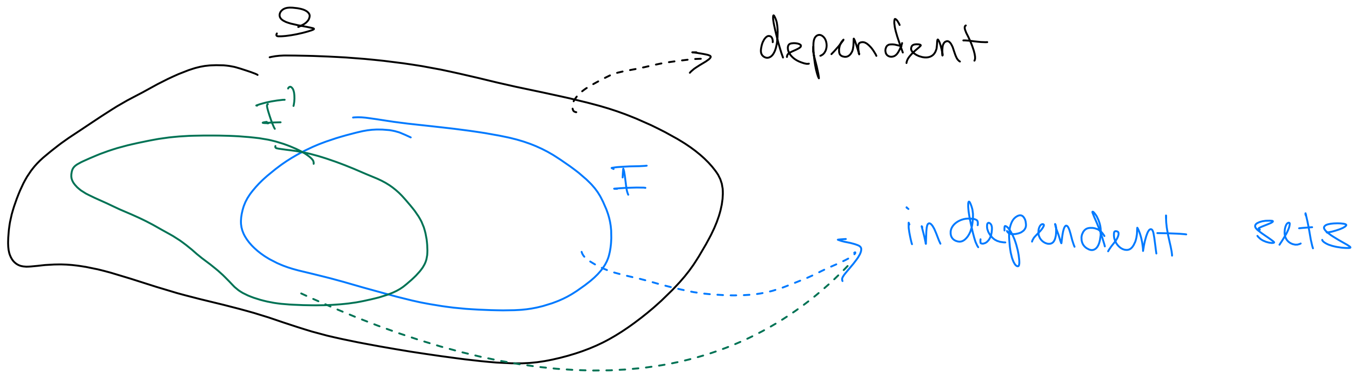
$$(B_1 \setminus X) \cup Y \quad \text{and} \quad (B_2 \setminus Y) \cup X$$

are also bases.



set of sets ~ bases

a set ~ a basis



- $M' = (S, I')$ and $M = (S, I)$ are dual matroids if the bases are complements
  - ↳ If B is a basis of M, then $S \setminus B$ is a basis of M'
  - $\Rightarrow$ any set $T \subseteq S \setminus B$ implies T is independent in M'