Posets and Dilworth's Theorem

Joonhyung Shin

August 4th, 2020

Introduction

- In many cases, we can define a certain *order*.
    - Real numbers: $0 < 1$, $\sqrt{5} < 3$, $\cdots$
    - Set inclusion: $\{1,2\} \subset \{1,2,5\}$, $\phi \subset \mathbb{N}$, $\cdots$
    - In a lab: Professor $>$ doctoral student $>$ master's student
    - Brain power: 만레라부싫사일레라부즐사단 $>$ joon
- An *order* over a set P is simply a subset of $P \times P$.
- What properties should an order satisfy?
- For instance, we don't want to have $a < b < c < a$.

Posets

Definition

A binary relation $\leq$ over a set P is a **partial order** if it satisfies the following:

- $a \leq a$ (reflexivity)
- if $a \leq b$ and $b \leq a$, then $a = b$ (antisymmetry)
- if $a \leq b$ and $b \leq c$, then $a \leq c$ (transitivity)

A pair $(P, \leq)$ with a set P together with a partial order $\leq$ is called a **partially ordered set**, or a **poset**.

- When $a \leq b$ and $a \neq b$, we write $a < b$.
- Note that there may exist two elements $a, b \in P$ that are neither $a \leq b$ nor $b \leq a$. In this case, they are said to be **incomparable**.

Examples of Posets

Example

- $(\mathbb{R}, \leq)$ is a poset.
- For a set S, $(2^S, \subseteq)$ is a poset.
- $(\mathbb{N}, |)$ is a poset, where $|$ is the divisibility relation.
- For a group G, $(\mathcal{S}, \leqslant)$ is a poset, where $\mathcal{S}$ is the set of subgroups of G and $\leqslant$ is the subgroup relation.
- For a directed acyclic graph (DAG) $G = (V, E)$, let $v_1 \preceq v_2$ for vertices v_1 and v_2 if and only if v_2 is reachable from v_1. Then $(V, \preceq)$ is a poset.

- Most of the time, we will only consider *finite* posets!

Chains and Antichains

Definition

Let $(P, \leq)$ be a poset.

- Let $C \subseteq P$. If for any $a, b \in C$ with $a \neq b$ we have either $a \leq b$ or $b \leq a$, then C is called a **chain**.
- Let $A \subseteq P$. If for any $a, b \in A$ with $a \neq b$ we have neither $a \leq b$ nor $b \leq a$, then A is called an **antichain**.

Examples of Chains and Antichains

Example

- $1 \leq 2 \leq 3 \leq \cdots \leq 100$ is a chain in $(\mathbb{R}, \leq)$. Every antichain of this poset is a singleton.
- $\{a\} \subseteq \{a, b\} \subseteq \{a, b, c, d, e\}$ is a chain, while $\{\{a, b\}, \{b, c\}, \{a, c\}\}$ is an antichain.
- For a DAG $G = (V, E)$, a chain $v_1 \preceq \cdots \preceq v_k$ gives a directed path from v_1 to v_k.

- Ok, so far everything is *combinatorial*.
- But what's so *optimization* about it?

Mirsky's Theorem

Theorem (Mirsky, 1971)

Let $(P, \leq)$ be a poset. Then the minimum size of a collection of disjoint antichains covering P is equal to the maximum size of a chain.

- Another min-max theorem!
- Try to imagine an intuitive image of this theorem.

Proof of Mirsky's Theorem

Theorem (Mirsky, 1971)

Let $(P, \leq)$ be a poset. Then the minimum size of a collection of disjoint antichains covering P is equal to the maximum size of a chain.

Proof.

It is clear that any collection of antichains covering P has size at least the maximum size of a chain.
Now for each $a \in P$, define $h(a)$ as the maximum size of a chain with maximum a. Let $A_k = \{a \in P : h(a) = k\}$. Then each A_k is an antichain and the size of the collection of A_k is equal to the size of the maximum chain. □

- This proof is constructive and gives an efficient polynomial time algorithm to find a minimum size antichain covering.
- When applied to the poset derived from a DAG, what would the time complexity be?

Erdös-Szekeres Theorem

Theorem (Erdös-Szekeres)

For positive integers n and m, let $X = (a_1, \cdots, a_{nm+1})$ be a sequence of real numbers of size $nm + 1$. Then there exists either a monotonically increasing subsequence of size $n + 1$ or a monotonically decreasing subsequence of size $m + 1$.

Proof.

Let $P = \{1, \cdots, nm + 1\}$ and define an order $\preceq$ on P as follows: $i \preceq j$ if and only if $i \leq j$ and $a_i \leq a_j$. Then a chain in $(P, \preceq)$ corresponds to a monotonically increasing subsequence, and an antichain in $(P, \preceq)$ corresponds to a strictly decreasing subsequence. By Mirsky's theorem, there is a chain of size at least $n + 1$ or an antichain cover of size at most n. In the latter case, by the pigeonhole principle, there should be an antichain of size at least $m + 1$. □

Longest Increasing Subsequence

- The well known $O(n \log n)$ algorithm for computing a longest increasing subsequence is related to Mirsky's theorem.
- Given a sequence $S = (a_1, \cdots, a_n)$, define an order $\preceq$ on $P = \{1, \cdots, n\}$ similar to the proof of Erdös-Szekeres theorem: $i \preceq j$ if and only if $i \leq j$ and $a_i < a_j$.
- Then we have the following:
    - A chain corresponds to a (strictly) increasing subsequence.
    - An antichain corresponds to a monotonically decreasing subsequence.
- By Mirsky's theorem, instead of finding a longest increasing subsequence, we can find a smallest partition consisting of monotonically decreasing subsequences.

Longest Increasing Subsequence

Example

Let $S = (1, 7, 2, 5, 8, 4, 3, 9, 10, 6)$. Starting from the beginning, try to place each element in a proper deck, while keeping the size of the decks as small as possible.

Weighted Mirsky's Theorem

- There is also a weighted variant of Mirsky's theorem.

Theorem

Let $(P, \leq)$ be a poset, and let $w : P \to \mathbb{Z}^+$ be a weight function. Then the minimum size of a collection of antichains covering each $a \in P$ exactly $w(a)$ times is equal to the maximum weight of a chain.

Proof.

Exercise! $\square$

Dilworth's Theorem

Theorem (Dilworth, 1950)

Let $(P, \leq)$ be a poset. Then the minimum size of a collection of disjoint chains covering P is equal to the maximum size of an antichain.

- There is a disjoint condition for the sake of consistency. It does not matter.
- In some sense, this is *dual* to Mirsky's theorem.
- This looks very similar to Mirsky's theorem, and there are indeed some relationships, but it is not so obvious as one may think. For example, finding a maximum anitchain is more difficult than finding a maximum chain.
- There are lots of interesting applications of Dilworth's theorem!
- Before the proof, let's take a look at some applications.

Applications of Dilworth's Theorem

Theorem (Gallai-Milgram)

Let $G = (V, E)$ be a DAG. Then the minimum size of a collection of paths covering V is equal to the maximum size of a path-independent set of vertices.

Proof.

Let $(V, \preceq)$ be the poset derived from G. Then the result follows from Dilworth's theorem. $\square$

- Note that unlike Dilworth's theorem, there is no *disjoint* condition here!
- The minimum path cover problem can be applied to software testing.

Applications of Dilworth's Theorem

- Recall the definition of directed cut.

Definition

Let $G = (V, E)$ be a DAG. For a nonempty vertex set $X \neq V$, $\delta^+(X)$ is called a **directed cut** if $\delta^-(X) = \phi$. If G has a unique source vertex s and a unique sink vertex t, then a directed cut $\delta^+(X)$ such that $s \in X$ and $t \notin X$ is called an s-t directed cut.

Applications of Dilworth's Theorem

Theorem

Let $G = (V, E)$ be a DAG. Then the minimum size of a collection of paths covering E is equal to the maximum size of a directed cut.

Proof.

Exercise! □

Theorem

Let $G = (V, E)$ be a DAG with a unique source s and a unique sink t. Then the minimum size of a collection of s-t paths covering E is equal to the maximum size of an s-t directed cut.

Applications of Dilworth's Theorem

- Recall Hall's theorem.
- It is possible to prove Hall's theorem using Dilworth's theorem!

Theorem (Hall)

Let $G = (X \cup Y, E)$ be a bipartite graph. Then G has a matching covering X if and only if $|N(A)| \geq |A|$ for every subset $A \subseteq X$.

Proof.

We only prove the converse. For $x, y \in X \cup Y$, let $x \preceq y$ if and only if $x \in X$, $y \in Y$, and $\{x, y\} \in E$. Then $(X \cup Y, \preceq)$ is a poset. By the marriage condition, the size of a maximum antichain is $|Y|$. By Dilworth's theorem, there is a chain cover of size $|Y|$, which gives a matching covering X. □

Weighted Dilworth's Theorem

- Similar to Mirsky's theorem, there is also a weighted version of Dilworth's theorem.
- The idea of the proof is similar to the weighted Mirsky's theorem and is left as an exercise!

Theorem

Let $(P, \leq)$ be a poset and let $w : P \to \mathbb{Z}^+$ be a weight function. Then the minimum size of a collection of chains covering each $a \in P$ exactly $w(a)$ times is equal to the maximum weight of an antichain.

Proof of Dilworth's Theorem

- There are many proofs of Dilworth's theorem.
- Today, we will (at least briefly) discuss four of them!
- Some of the proofs automatically give a polynomial algorithm for finding a minimum chain cover or a maximum antichain.

Direct Proof

We proceed by induction on $|P|$.
Let A be a maximum antichain of $(P, \leq)$. We can divide P into two sets:

- $A^{\uparrow} = \{b \in P : a \leq b \text{ for some } a \in A\}$
- $A^{\downarrow} = \{b \in P : b \leq a \text{ for some } a \in A\}$

Then we have $A^{\uparrow} \cap A^{\downarrow} = A$ and $A^{\uparrow} \cup A^{\downarrow} = P$.
Now we consider two cases:

- $A^{\uparrow} \neq A$ and $A^{\downarrow} \neq A$
- $A^{\uparrow} = A$ or $A^{\downarrow} = A$

Direct Proof

Case 1: $A^{\uparrow} \neq A$ and $A^{\downarrow} \neq A$
In this case, we apply the induction hypothesis to $A^{\uparrow}$ and $A^{\downarrow}$, whereby we obtain two collections of chain $\mathcal{C}^{\uparrow}$ and $\mathcal{C}^{\downarrow}$, respectively. Note that:

- $|\mathcal{C}^{\uparrow}| = |\mathcal{C}^{\downarrow}| = |A|$
- For each $C \in \mathcal{C}^{\uparrow}$ we have $|C \cap A| = 1$.
- For each $C \in \mathcal{C}^{\downarrow}$ we have $|C \cap A| = 1$.

Thus, we can merge the two collections $\mathcal{C}^{\uparrow}$ and $\mathcal{C}^{\downarrow}$ to obtain a chain cover of P of size $|A|$.

Direct Proof

Case 2: $A^{\uparrow} = A$ or $A^{\downarrow} = A$
In this case, A is either a set of minimal elements or a set of maximal elements.
Choose $x, y \in P$ so that:

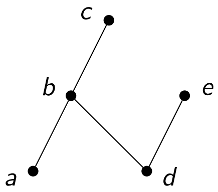
- x is a minimal element
- y is a maximal element
- $x \leq y$.

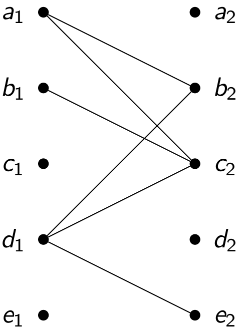
Then by applying the induction hypothesis to $(P \setminus \{x, y\}, \leq)$ we have a chain collection of size $|A| - 1$. Adding $\{x, y\}$, we obtain a chain cover of size $|A|$.

Direct Proof

- This proof is neat, but it does not give an efficient algorithm since the proof assumes that we already have a maximum antichain.
- Is there any proof that gives us a polynomial time algorithm?

Proof Using Bipartite Matching

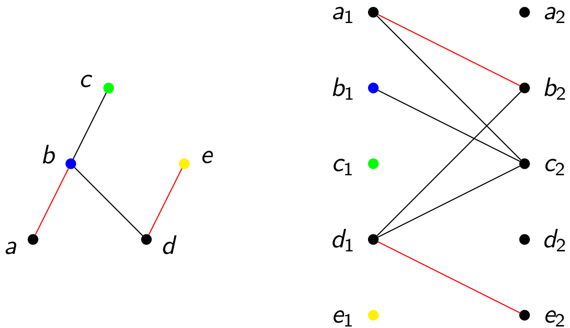




From $(P, \leq)$, we create a bipartite graph according to the following:

- For each $a \in P$, we create two vertices $a_1 \in V_1$ and $a_2 \in V_2$.
- For each $a \leq b$, we create an edge from a_1 to b_2.

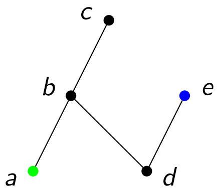
Proof Using Bipartite Matching

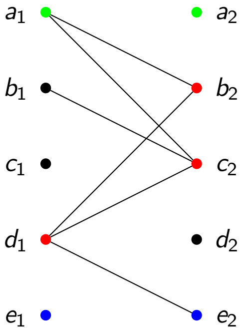


A matching M of size $|M|$ corresponds to a chain cover of size $|P| - |M|$:

- Each vertex in V_1 not covered by M corresponds to the maximal element of the corresponding chain.
- Each edge in M corresponds to a *child-parent* relation in a chain.

Proof Using Bipartite Matching

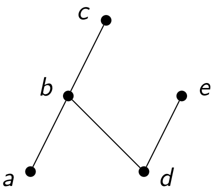


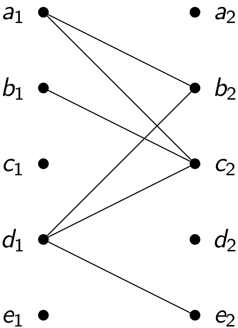


A vertex cover X of size $|X|$ corresponds to an antichain of size $|P| - |X|$.

- The set of vertex a such that a_1 and a_2 are both not in X is an antichain.
- An antichain A of size $|A|$ can be used to create a vertex cover of size $|P| - |A|$.

Proof Using Bipartite Matching





- By König's theorem, these two quantities are equal!
- Therefore, König's theorem implies Dilworth's theorem.
- Moreover, it gives an efficient algorithm to compute both a maximum antichain and a minimum chain cover!

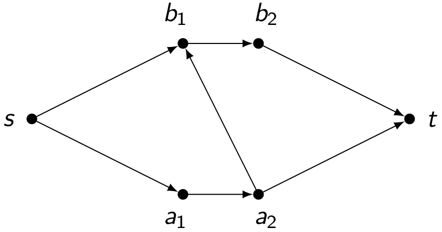
Proof Using Minimum Flow

- Covering the set with chains looks like sending a flow.
- In fact, it is not difficult to model the chain cover problem using the language of flow.
- However, we are familiar with the *maximum* flow problem, whereas in Dilworth's theorem its about *minimizing* the cover.
- So, why don't we first talk about a *minimum* flow problem?

Proof Using Minimum Flow

- ...In fact, there is not much to talk about.
- For a flow bound $\ell(e) \leq x_e \leq c(e)$, all we have to do is change this to $-c(e) \leq x_e \leq -\ell(e)$ and compute a maximum flow.
- A minimum cut corresponds to a maximum cut. (Min-Flow/Max-Cut theorem?)
- So its essentially just the same. This is why no one cares about minimum flow.
- Now, let's model Dilworth's theorem using flow!

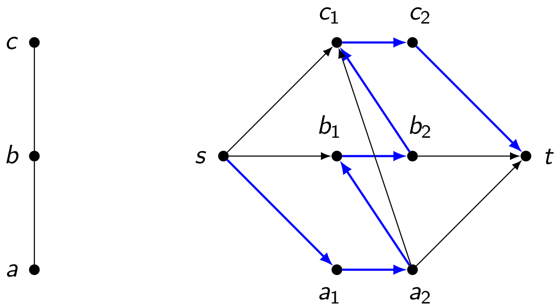
Proof Using Minimum Flow



We construct a new flow graph as follows:

- For each $a \in P$, create two vertices a_1 and a_2. Add an edge $a_1 \to a_2$ with flow lower bound 1 and edges $s \to a_1$ and $a_2 \to t$ with flow lower bound 0.
- For each $a \leq b$, add an edge $a_2 \to b_1$ with flow lower bound 0.

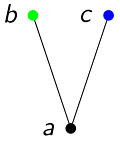
Proof Using Minimum Flow

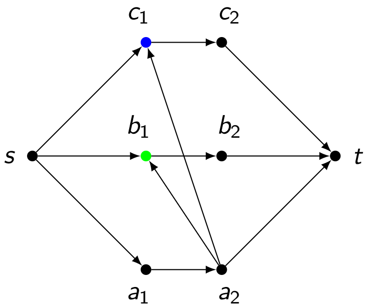


A chain in a chain cover corresponds to an augmenting flow of size 1. Therefore, we have the following:

- The size of a minimum chain cover is equal to the minimum flow.
- We can find a minimum chain cover by finding a minimum flow.

Proof Using Minimum Flow



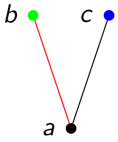


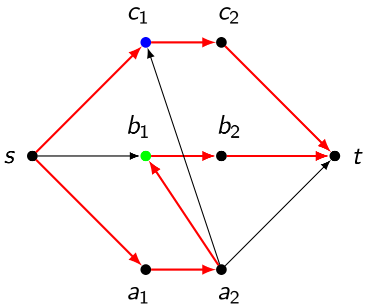
A cutset S with $\delta^-(S) = \phi$ corresponds to an antichain consisting of each $a \in P$ that satisfies the following:

- Its first copy a_1 is in S.
- Its second copy a_2 is not in S.

Note that the condition $\delta^-(S) = \phi$ makes this an antichain.

Proof Using Minimum Flow





Since the edges have no upper bound, a cutset S with a maximum cut produced by a minimum flow will always have $\delta^-(S) = \phi$! Here's the conclusion:

- The Min-Flow/Max-Cut theorem implies Dilworth's theorem.
- We have another polynomial time algorithm for finding both a minimum chain cover and a maximum antichain.

Proof Using Mirsky's Theorem

- As mentioned before, Mirsky's theorem and Dilworth's theorem are related, and it is possible to prove Dilworth's theorem using Mirsky's theorem.
- However, the proof appeals to the perfect graph theorem, which is out of scope of today's material.
- For those who are interested, the proof is outlined in one of the exercises!

Exercises

- There are total 7 problems waiting for you!
- Please solve at least 3 problems.
- I think the problems are not very easy, but not very difficult either. You can do it. :)
- Feel free to discuss on Discord!

References

- W. Pijls, R. Potharst, *Dilworth's theorem revisited, an algorithmic proof*
- `https://www.ic.unicamp.br/~lee/mo824/`
- `https://www.geeksforgeeks.org/dilworths-theorem/`
- `https://brilliant.org/wiki/dilworths-theorem/#applications`