Learning Goals

- Definition of disjoint-path covers and the kind of problems they model
- Reduction of the disjoint-path covers problem to bipartite matching

Motivating Question

- There are n construction sites in different locations; each of them needs a heavy machine for a project on a particular day. The machine needed by each site is identical.

Motivating Question

- There are n construction sites in different locations; each of them needs a heavy machine for a project on a particular day. The machine needed by each site is identical.
- Site i needs the machine from time a_i to b_i;

Motivating Question

- There are n construction sites in different locations; each of them needs a heavy machine for a project on a particular day. The machine needed by each site is identical.
- Site i needs the machine from time a_i to b_i;
- It takes time c_{ij} to move the machine from site i to site j;

Motivating Question

- There are n construction sites in different locations; each of them needs a heavy machine for a project on a particular day. The machine needed by each site is identical.
- Site i needs the machine from time a_i to b_i;
- It takes time c_{ij} to move the machine from site i to site j;
- How many machines do we need so that all sites can get their jobs done?

Model

- Construct a directed graph G, each node v_i for a site i;

Model

- Construct a directed graph G, each node v_i for a site i;
- There is an edge from v_i to v_j if it is feasible to move a machine to site j after it is used by site i:

Model

- Construct a directed graph G, each node v_i for a site i;
- There is an edge from v_i to v_j if it is feasible to move a machine to site j after it is used by site i:

$$b_i + c_{ij} \leq a_j.$$

Model

- Construct a directed graph G, each node v_i for a site i;
- There is an edge from v_i to v_j if it is feasible to move a machine to site j after it is used by site i:

$$b_i + c_{ij} \leq a_j.$$

Note that the resulting graph is acyclic;

Model

- Construct a directed graph G, each node v_i for a site i;
- There is an edge from v_i to v_j if it is feasible to move a machine to site j after it is used by site i:

$$b_i + c_{ij} \leq a_j.$$

  Note that the resulting graph is acyclic;
- A path in G contains a set of sites whose jobs can be done sequentially by one machine.

Model

- Construct a directed graph G, each node v_i for a site i;
- There is an edge from v_i to v_j if it is feasible to move a machine to site j after it is used by site i:

$$b_i + c_{ij} \leq a_j.$$

  Note that the resulting graph is acyclic;
- A path in G contains a set of sites whose jobs can be done sequentially by one machine.
- We'd like to find a minimum set of paths so that each vertex is on one path.

Problem Formulation

Definition

Given a directed graph, a set $\mathcal{P}$ of simple paths is a *path cover* if each node lies on at least one path in $\mathcal{P}$.

Problem Formulation

Definition

Given a directed graph, a set $\mathcal{P}$ of simple paths is a *path cover* if each node lies on at least one path in $\mathcal{P}$.
A path cover $\mathcal{P}$ is a *disjoint-path cover* if each vertex lies on *exactly* one path in $\mathcal{P}$.

Problem Formulation

Definition

Given a directed graph, a set $\mathcal{P}$ of simple paths is a *path cover* if each node lies on at least one path in $\mathcal{P}$.
A path cover $\mathcal{P}$ is a *disjoint-path cover* if each vertex lies on *exactly* one path in $\mathcal{P}$.

- Input: Directed acyclic graph (DAG) $G = (V, E)$
- Output: A smallest disjoint-path cover $\mathcal{P}$

Problem Formulation

Definition

Given a directed graph, a set $\mathcal{P}$ of simple paths is a *path cover* if each node lies on at least one path in $\mathcal{P}$.
A path cover $\mathcal{P}$ is a *disjoint-path cover* if each vertex lies on *exactly* one path in $\mathcal{P}$.

- Input: Directed acyclic graph (DAG) $G = (V, E)$
- Output: A smallest disjoint-path cover $\mathcal{P}$

Remark: The problem without the acyclic condition is NP-hard.

Reduction to Bipartite Matching

The algorithm:

- Construct a bipartite graph $G' = (L, R, F)$; for each node v of G, create $\ell_v \in L$ and $r_v \in R$;

Reduction to Bipartite Matching

The algorithm:

- Construct a bipartite graph $G' = (L, R, F)$; for each node v of G, create $\ell_v \in L$ and $r_v \in R$;
- For each edge (u, v) in G, create edge (ℓ_u, r_v) in G';

Reduction to Bipartite Matching

The algorithm:

- Construct a bipartite graph $G' = (L, R, F)$; for each node v of G, create $\ell_v \in L$ and $r_v \in R$;
- For each edge (u, v) in G, create edge (ℓ_u, r_v) in G';
- Find the maximum matching M^* in G'.

Reduction to Bipartite Matching

The algorithm:

- Construct a bipartite graph $G' = (L, R, F)$; for each node v of G, create $\ell_v \in L$ and $r_v \in R$;
- For each edge (u, v) in G, create edge (ℓ_u, r_v) in G';
- Find the maximum matching M^* in G'.

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M^*|$.

Reduction to Bipartite Matching

The algorithm:

- Construct a bipartite graph $G' = (L, R, F)$; for each node v of G, create $\ell_v \in L$ and $r_v \in R$;
- For each edge (u, v) in G, create edge (ℓ_u, r_v) in G';
- Find the maximum matching M^* in G'.

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M^*|$.

The proof will contain an algorithmic construction of $\mathcal{P}$ from M^*.

Review of last lecture

- Given a directed graph, a set $\mathcal{P}$ of simple paths is a *path cover* if each node lies on at least one path in $\mathcal{P}$. A path cover $\mathcal{P}$ is a *disjoint-path cover* if each vertex lies on *exactly* one path in $\mathcal{P}$.
- The algorithm:
  - Construct a bipartite graph $G' = (L, R, F)$; for each node v of G, create $\ell_v \in L$ and $r_v \in R$;

Review of last lecture

- Given a directed graph, a set $\mathcal{P}$ of simple paths is a *path cover* if each node lies on at least one path in $\mathcal{P}$. A path cover $\mathcal{P}$ is a *disjoint-path cover* if each vertex lies on *exactly* one path in $\mathcal{P}$.
- The algorithm:
  - Construct a bipartite graph $G' = (L, R, F)$; for each node v of G, create $\ell_v \in L$ and $r_v \in R$;
  - For each edge (u, v) in G, create edge (ℓ_u, r_v) in G';

Review of last lecture

- Given a directed graph, a set $\mathcal{P}$ of simple paths is a *path cover* if each node lies on at least one path in $\mathcal{P}$. A path cover $\mathcal{P}$ is a *disjoint-path cover* if each vertex lies on *exactly* one path in $\mathcal{P}$.
- The algorithm:
  - Construct a bipartite graph $G' = (L, R, F)$; for each node v of G, create $\ell_v \in L$ and $r_v \in R$;
  - For each edge (u, v) in G, create edge (ℓ_u, r_v) in G';
  - Find the maximum matching M^* in G'.

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M^*|$.

Proof of reduction

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M|$.

Proof.

- Every matching M in G' of size k corresponds to a disjoint-path cover $\mathcal{P}$ in G of size $|V| - k$.

Proof of reduction

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ *has* $|\mathcal{P}^*| = |V| - |M|$.

Proof.

- Every matching M in G' of size k corresponds to a disjoint-path cover $\mathcal{P}$ in G of size $|V| - k$.
  - Construct a subgraph H of G with the same node set V: include the edge (u, v) in H if $(\ell_u, r_v) \in M$.

Proof of reduction

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M|$.

Proof.

- Every matching M in G' of size k corresponds to a disjoint-path cover $\mathcal{P}$ in G of size $|V| - k$.
  - Construct a subgraph H of G with the same node set V: include the edge (u, v) in H if $(\ell_u, r_v) \in M$.
  - Every vertex in H has at most one incoming edge and at most one outgoing edge.

Proof of reduction

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M|$.

Proof.

- Every matching M in G' of size k corresponds to a disjoint-path cover $\mathcal{P}$ in G of size $|V| - k$.
  - Construct a subgraph H of G with the same node set V: include the edge (u, v) in H if $(\ell_u, r_v) \in M$.
  - Every vertex in H has at most one incoming edge and at most one outgoing edge.
  - Hence H is a collection of disjoint paths; this collection is a path cover in G, and has k edges.

Proof of reduction

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M|$.

Proof.

- Every matching M in G' of size k corresponds to a disjoint-path cover $\mathcal{P}$ in G of size $|V| - k$.
  - Construct a subgraph H of G with the same node set V: include the edge (u, v) in H if $(\ell_u, r_v) \in M$.
  - Every vertex in H has at most one incoming edge and at most one outgoing edge.
  - Hence H is a collection of disjoint paths; this collection is a path cover in G, and has k edges.
  - The number of paths in this path cover is $|V| - k$.

Proof of reduction cont.

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M^*|$.

Proof.

- Every disjoint-path cover $\mathcal{P}$ in G with k paths corresponds to a matching in G' of size $|V| - k$.

Proof of reduction cont.

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M^*|$.

Proof.

- Every disjoint-path cover $\mathcal{P}$ in G with k paths corresponds to a matching in G' of size $|V| - k$.
  - Construct a subset M of edges in G: include in M the edge (ℓ_u, r_v) if (u, v) is in a path in $\mathcal{P}$.

Proof of reduction cont.

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M^*|$.

Proof.

- Every disjoint-path cover $\mathcal{P}$ in G with k paths corresponds to a matching in G' of size $|V| - k$.
  - Construct a subset M of edges in G: include in M the edge (ℓ_u, r_v) if (u, v) is in a path in $\mathcal{P}$.
  - Each node in G has at most one outgoing edge in $\mathcal{P}$, so each node in L is incident to at most one edge in M.

Proof of reduction cont.

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M^*|$.

Proof.

- Every disjoint-path cover $\mathcal{P}$ in G with k paths corresponds to a matching in G' of size $|V| - k$.
  - Construct a subset M of edges in G: include in M the edge (ℓ_u, r_v) if (u, v) is in a path in $\mathcal{P}$.
  - Each node in G has at most one outgoing edge in $\mathcal{P}$, so each node in L is incident to at most one edge in M.
  - Each node in G has at most one incoming edge in $\mathcal{P}$, so each node in R is incident to at most one edge in M.

Proof of reduction cont.

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M^*|$.

Proof.

- Every disjoint-path cover $\mathcal{P}$ in G with k paths corresponds to a matching in G' of size $|V| - k$.
  - Construct a subset M of edges in G: include in M the edge (ℓ_u, r_v) if (u, v) is in a path in $\mathcal{P}$.
  - Each node in G has at most one outgoing edge in $\mathcal{P}$, so each node in L is incident to at most one edge in M.
  - Each node in G has at most one incoming edge in $\mathcal{P}$, so each node in R is incident to at most one edge in M.
  - So M is a matching, and since $\mathcal{P}$ has $|V| - k$ edges, $|M| = |V| - k$.

□

Proof of reduction cont.

Claim

The smallest disjoint-path cover $\mathcal{P}^*$ has $|\mathcal{P}^*| = |V| - |M^*|$.

Proof.

- Every disjoint-path cover $\mathcal{P}$ in G with k paths corresponds to a matching in G' of size $|V| - k$.
  - Construct a subset M of edges in G: include in M the edge (ℓ_u, r_v) if (u, v) is in a path in $\mathcal{P}$.
  - Each node in G has at most one outgoing edge in $\mathcal{P}$, so each node in L is incident to at most one edge in M.
  - Each node in G has at most one incoming edge in $\mathcal{P}$, so each node in R is incident to at most one edge in M.
  - So M is a matching, and since $\mathcal{P}$ has $|V| - k$ edges, $|M| = |V| - k$.

□

Question: Where did we use that G is acyclic?