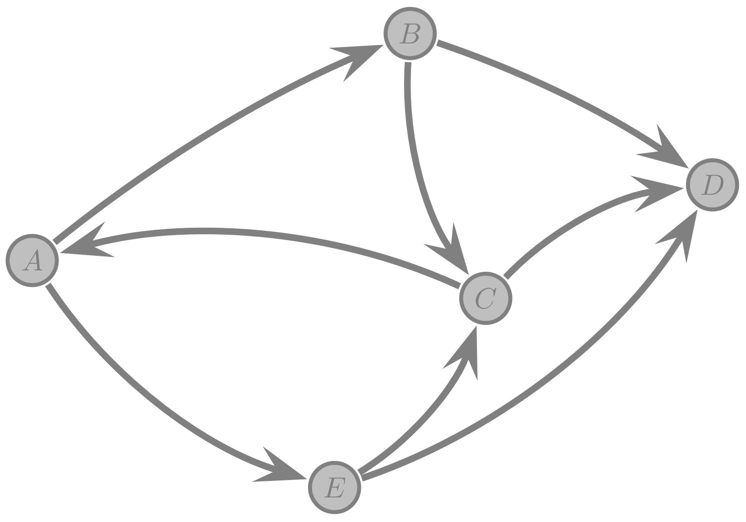
Networks and Combinatorial Optimization

Fall 2023

Thomas Rothvoss



UNIVERSITY of WASHINGTON

Last changes: December 26, 2024

Contents

Chapter 1

Minimum Spanning Trees

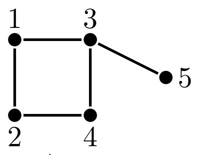
1.1 Foreword

Discrete optimization deals with finding the best solution out of a finite number of possibilities in a computationally efficient way. Typically the number of possible solutions is exponentially large and in order to obtain an optimum solution efficiently, insights into the *problem structure* are required to succeed. These lecture notes are to a good part based on the excellent lecture notes by Schrijver [Sch17] (in particular our earlier chapters). Other parts are based on different sources that will be mentioned at the beginning of the respective chapters. The reader should understand that these notes are solely created for reference for the students of the Math 514 course and we do not claim any originality in the exposition or in the results. In case of any correctness issues, the reader is refered to the original work. Sections that are somewhat optional and are not necessarily covered in every iteration of 514 are marked with *.

1.2 Undirected graphs

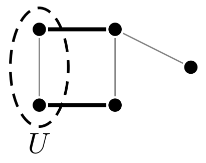
Many mathematical optimization problems can be naturally phrased as a problem on networks or graphs. For that reason we introduce the most common terms. Unfortunately, many graph theory terms are not standardized; we follow here the definitions of Diestel [Die12].

An *undirected graph* is of the form $G = (V, E)$ where V is a finite set and $E \subseteq \{\{u, v\} : u, v \in V, u \neq v\}$. We call V the *vertices* and E the *edges* of G. Here is a visualization of an example graph:



Graph $G = (V, E)$ with $V = \{1, 2, 3, 4, 5\}$
and $E = \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 4\}, \{3, 5\}\}$

We also write $G = (V(G), E(G))$. We will also use the terms *vertices* and *nodes* interchengably. Given a graph $G = (V, E)$, and a subset $U \subseteq V$, the edge set $\delta(U) = \{\{u, v\} \in E \mid |\{u, v\} \cap U| = 1\}$ is called a *cut*. For example, the bold edges in the figure below form a cut.

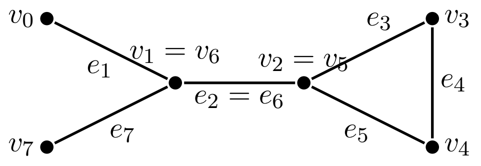


A *subgraph* of $G = (V(G), E(G))$ is a graph $H = (V(H), E(H))$ where $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ with the restriction that if $\{i, j\} \in E(H)$ then $i, j \in V(H)$.

If $V' \subseteq V(G)$, then the subgraph *induced* by V' is the graph $(V', E(V'))$ where $E(V')$ is the set of all edges in G for which both vertices are in V'.

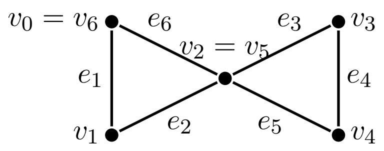
A subgraph H of G is a *spanning subgraph* of G if $V(H) = V(G)$, i.e. the subgraph contains all vertices of the original graph.

A *walk* in a graph $G = (V, E)$ is a sequence of vertices and edges $v_0, e_1, v_1, e_2, v_2, e_3, \ldots, e_k, v_k$, such that for $i = 0, \ldots, k$, $v_i \in V$, $e_i \in E$ where $e_i = \{v_{i-1}, v_i\}$.



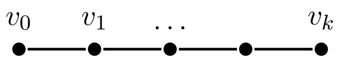
a walk of length 7

A walk in which start and end vertex are identical, i.e. $v_0 = v_k$, is a *closed walk*.



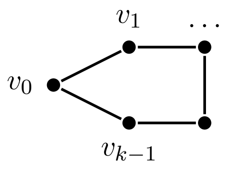
closed walk of length 6

A *path* is a graph $P = (V, E)$ with vertices $V = \{v_0, v_1, \ldots, v_k\}$ and edges $E = \{\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-1}, v_k\}\}$ and all $v_0, \ldots, v_k$ are distinct. The *length* of the path is the number of edges in the path which equals k. We refer to such a path as (v_0, v_k)-*path* if we want to emphasize the endpoints.



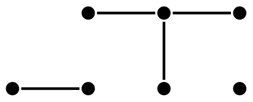
path

A *cycle* is a graph $G = (V, E)$ with $V = \{v_0, v_1, \dots, v_{k-1}\}$ and $E = \{\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{k-1}, v_0\}\}$ where $v_0, \dots, v_{k-1}$ are distinct and $k \geq 3$.

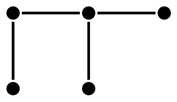


cycle with $k = 5$ vertices and edges

A graph G is *acyclic* if it contains no cycle as subgraphs. An acyclic graph is called a *forest*. A connected forest $T = (V(T), E(T))$ is a *tree*.

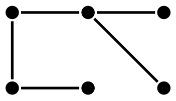


forest



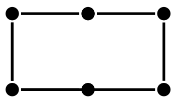
tree T

A subgraph $T = (V(T), E(T))$ is a *spanning tree* of G, if T is spanning, connected and acyclic.



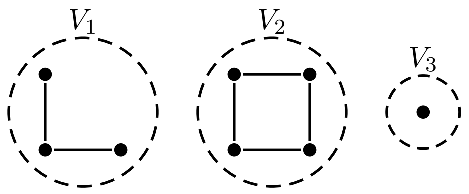
spanning tree

A *Hamiltonian circuit* of G is a subgraph that is a spanning cycle.



Hamiltonian circuit/tour

Given a graph $G = (V, E)$ we can define an *equivalence relation* so that $u \sim v$ if there exists a u-v path in G. Let $V_1, \dots, V_k$ be the *equivalence classes* of that relation. Then the induced subgraphs $G[V_1], \dots, G[V_k]$ are called *(connected) components* of G.



(connected) components

A set $M \subseteq E$ of edges with degree at most 1 for each vertex is called *matching.* A set $M \subseteq E$ of edges with degree exactly 1 for each vertex is called *perfect matching.*

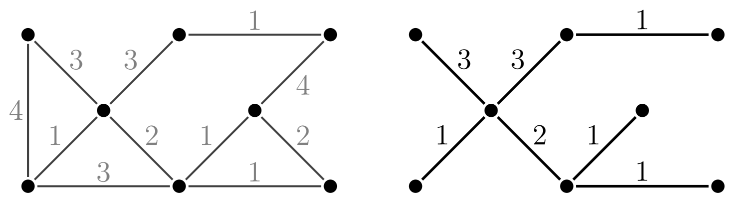
matching perfect matching

We want to make the following *convention:* Formally speaking paths / trees / spanning trees / cycles / Hamiltonian circuits are defined as *(sub)graphs* $H = (V(H), E(H))$. Often we will refer to an edge set $E(H)$ as paths (or trees etc) in which case the subgraph is meant that contains all the vertices incident to $E(H)$.

1.3 Minimum Spanning Trees

This section is a reproduction of Section 1.4 in [Sch17]. The first problem that we discuss will be the one of finding the cheapest spanning tree in a graph:

MINIMUM SPANNING TREE
Input: Undirected graph $G = (V, E)$, length function $\ell : E \to \mathbb{R}$
Goal: A spanning tree T of G (i.e. $E(T) \subseteq E(G)$) minimizing $\ell(T) := \sum_{e \in E(T)} \ell(e)$.



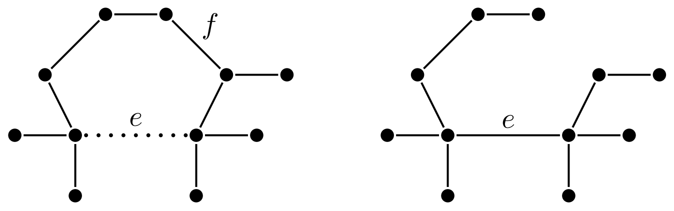
graph $G = (V, E)$ minimum spanning tree T

One may easily imagine applications to designing road systems, electrical power lines or telephone lines. We summarize two facts on spanning trees (we leave the proofs to the reader).

Lemma 1.1. *Let T be a spanning connected subgraph of G. The following conditions are equivalent*

- *T is a spanning tree (i.e. acyclic).*
- *One has $|E(T)| = |V(T)| - 1$.*
- *For each edge $e = \{u, v\} \in E$ there exists a unique u-v path in T.*

Lemma 1.2. *Let T be a spanning tree in G. Suppose $e \in E(G) \setminus E(T)$ and let f be any edge on the unique path in T between the end points of e. Then $T' = (V(T), E(T) \setminus \{f\} \cup \{e\})$ is again a spanning tree in G.*



spanning tree T spanning tree T'

The Dijkstra-Prim algorithm

We suggest the following algorithm:

Dijkstra-Prim Algorithm

Input: A connected graph G with edge costs $\ell : E \to \mathbb{R}$.
Output: A MST T of G.

```
(1) Choose any $v \in V$ and set $T := (\{v\}, \emptyset)$
(2) WHILE $V(T) \neq V$
        (3) Choose $e \in \delta(V(T))$ of minimal length
        (4) Add $e$ to $E(T)$ (and include endpoint of $e$)
```

We would like to note that the earliest algorithm for finding an MST is due to Boruvka (1926). Later variants appeared by Dijkstra [Dij59] and Prim [Pri57].

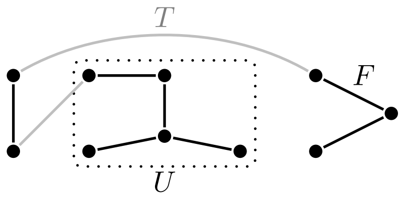
The algorithm above is also called a "Greedy algorithm". While there is no formal definition of what a greedy algorithm is, typically a greedy algorithm builds a solution iteratively by selecting the cheapest option in every step without looking ahead or revising past choices. We will now prove that the Disjkstra-Prim algorithm always finds an optimum solution. The following definition will be useful:

Definition 1.3. Fix a graph G and length function ℓ. A forest F (in G) is called *greedy* if there exists a minimum spanning tree T with $E(F) \subseteq E(T)$.

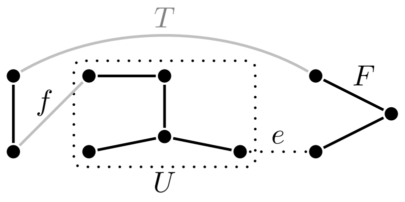
Intuitively, this means a forest is greedy if it can be extended to a minimum spanning tree. Now to the main technical claim that shows how one can grow a forest while keeping it greedy:

Proposition 1.4. *Let F be a greedy forest, U be one of the connected components. If $e \in \delta(U)$ is an edge of minimum length in $\delta(U)$, then $F \cup \{e\}$ is again a greedy forest.*

Proof. Let F be a greedy forest and let U be one of its connected components U. As F is greedy, there must be some MST T with $E(T) \supseteq E(F)$.



Select $e \in \delta(U)$ of minimum length. Since T is a spanning tree, there must be a unique path P in T between the endpoints of e. Since one endpoint of P lies in U and one endpoint lies outside of U, there must be an edge $f \in E(P) \cap \delta(U)$.



We have $\ell(e) \leq \ell(f)$ by assumption and moreover $T' := (V, (E(T) \setminus \{f\}) \cup \{e\})$ is a spanning tree. Then $\ell(T') = \ell(T) - \ell(f) + \ell(e) \leq \ell(T)$ and hence T' is an MST that includes $F \cup \{e\}$. □

Using this proposition it is easy to prove correctness of the Dijkstra-Prim algorithm.

Corollary 1.5. *The Dijkstra-Prim algorithm yields a MST of G.*

Proof. At start $T = (\{v\}, \emptyset)$ is a greedy forest. In every step, the Dijkstra-Prim algorithm selects a cheapest edge in $\delta(V(T))$, hence by Proposition 1.4, T remains a greedy forest. At the end, T is a spanning tree that is greedy. By definition of greedyness, there must be an MST T^* with $E(T) \subseteq E(T^*)$, but then $T^* = T$. □

Kruskal's algorithm

One can also construct a different greedy-style algorithm that selects edges without keeping the edge set connected.

```
Kruskal's Algorithm
Input: A connected graph $G$ with edge costs $\ell : E \to \mathbb{R}$.
Output: A MST $T$ of $G$.
  (1) Sort the edges such that $\ell(e_1) \leq \ell(e_2) \leq \ldots \leq \ell(e_m)$.
  (2) Set $T = (V, \emptyset)$
  (3) For $i$ from 1 to $m$ do
          If $T \cup \{e_i\}$ is acyclic then update $T := T \cup \{e_i\}$.
```

We will also prove correctness for Kruskal's algorithm:

Theorem 1.6. *Kruskal's algorithm computes an MST.*

Proof. In fact, we will use the same Proposition 1.4. Again, at the beginning $(V, \emptyset)$ is trivially a greedy forest. In every step we add a cheapest edge crossing one of the connected components. So we again terminate with a connected greedy forest which must be an MST. □

1.4 The Maximum Reliability Problem

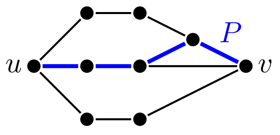
Minimum spanning trees have numerous applications. Here we will showcase one. We are given a graph $G = (V, E)$ and a function $s : E \to \mathbb{R}_{\geq 0}$ where we call $s(e)$ the *strength* of an edge. For a path P in G, we define the *reliability* as

$$r(P) := \min_{e \in E(P)} s(e)$$

Moreover, for a pair $u, v \in V$, define the *reliability* as

$$r_G(u, v) := \max\{r(P) : P \text{ is } u\text{-}v \text{ path in } G\}$$

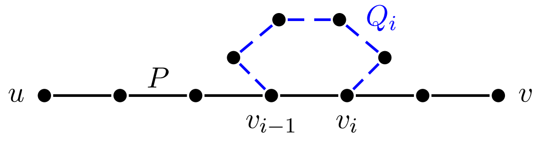
One can imagine that G is a communication network and if we send data between u and v along a path P then the minimal bandwidth on any edge determines the data rate. Naturally we would be interested in determining the u-v path P that maximizes the data rate, which would define $r_G(u, v)$.



We will not only prove that one can use an MST algorithm to solve the maximum reliability problem, but also show that an MST with respect to lengths $-s(e)$ provides a compact solution:

Theorem 1.7. *Let T be a spanning tree in G maximizing $\sum_{e\in E(T)} s(e)$. Then $r_T(u,v) = r_G(u,v)\ \forall u,v \in V$.*

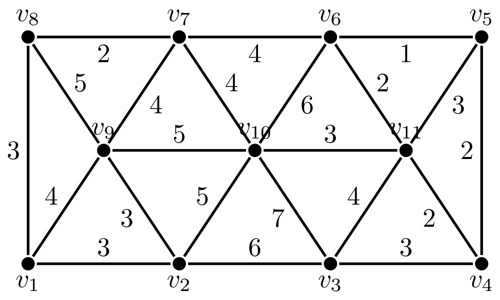
Proof. It suffices to prove that $r_T(u,v) \geq r_G(u,v)$. Fix a u-v path P in G. We need to prove that there is a path Q in T so that $r(Q) \geq r(P)$. Let $v_0,\ldots,v_m$ be the nodes of P with $\{v_{i-1},v_i\} \in E(P)$ and let Q_i be the unique v_{i-1}-v_i path in T. We know by optimality of T that $s(e) \geq s(v_{i-1},v_i)$ for all $e \in E(Q_i)$. Let Q be concatenation of $Q_1,\ldots,Q_m$. Then Q is a u-v walk in T with $r(Q) \geq r(P)$.



$\square$

1.5 Exercises

Exercise 1.1.
Find, both with the Dijkstra-Prim algorithm and with Kruskal's algorithm, a spanning tree of minimum length in the graph in the figure below.



Source: This exercise is taken from Schrijver [Sch17].

Exercise 1.2.
Let $G = (V,E)$ be a graph and let $\ell : E \to \mathbb{R}$ be a length function. Call a forest F *good* if $\ell(F') \geq \ell(F)$ for each forest satisfying $|F'| = |F|$.

Let F be a good forest and e be an edge not in F so that $F \cup \{e\}$ is a forest and such that (among all such e) $\ell(e)$ is as small as possible. Show that $F \cup \{e\}$ is good again.
Source: This exercise is taken from Schrijver [Sch17].

Chapter 2

Matroids — the basics

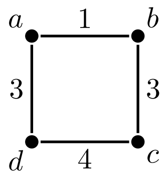
This chapter is a reproduction of Chapters 10.1 and 10.3 in Schrijver [Sch17]. Let us recall Kruskal's algorithm from Chapter 1.3 to find an MST in a graph $G = (V, E)$ with $|E| = m$ and length function $\ell : E \to \mathbb{R}$:

(1) Sort the edges such that $\ell(e_1) \leq \ell(e_2) \leq \ldots \leq \ell(e_m)$.

(2) Set $T = (V, \emptyset)$

(3) For i from 1 to m do
    If $T \cup \{e_i\}$ is acyclic then update $T := T \cup \{e_i\}$.

While Kruskal's algorithm is a *greedy algorithm* we have proven that for any graph and any weights ℓ, it always finds the optimum. Clearly one can design very similar greedy-style algorithms for other problems. For example, suppose we wanted to find a maximum weight matching in a graph $G = (V, E)$. If in the graph below we select the maximum weight edge first, then we will arrive at a solution of value $4 + 1 = 5$ while there is a better matching of value $3 + 3 = 6$.



Certainly some structure is needed so that greedy solutions are optimal. That structure is precisely captured by matroids!

Definition 2.1. A *matroid* is a pair $M = (X, \mathcal{I})$ where X is a finite set and $\mathcal{I} \subseteq 2^X$ so that the following holds:

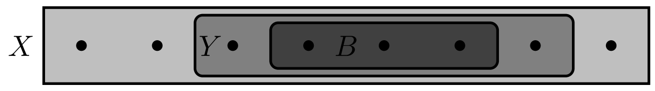
(i) *Non-emptyness:* $\emptyset \in \mathcal{I}$

(ii) *Monotonicity:* If $Y \in \mathcal{I}$ and $Z \subseteq Y$ then $Z \in \mathcal{I}$.

(iii) *Exchange property:* If $Y, Z \in \mathcal{I}$ and $|Y| < |Z|$ then for some $x \in Z \setminus Y$ one has $Y \cup \{x\} \in \mathcal{I}$.

The set X is called the *ground set* and the family of sets $\mathcal{I}$ is the family of *independent* sets.

Definition 2.2. Let $Y \subseteq X$. A set $B \subseteq Y$ is called a *basis of* Y if $B \in \mathcal{I}$ and for all $x \in Y \setminus B$ one has $B \cup \{x\} \notin \mathcal{I}$.



Phrased differently, a basis is an inclusion-wise maximal independent set (always with respect to some set $Y \subseteq X$ that does not need to be itself independent). We state a lemma that we will prove in the exercises:

Lemma 2.3. *Let $M = (X, \mathcal{I})$ be a matroid and let $Y \subseteq X$. Any two bases B_1, B_2 of Y have the same cardinality, i.e. $|B_1| = |B_2|$.*

Since bases of Y have the same cardinality it makes sense to give a name to that quantity. We define the *rank* of Y as

$$r_M(Y) = \max\{|B| : B \subseteq Y \text{ and } B \in \mathcal{I}\}$$

We define *rank of the matroid* M itself as $r_M(X)$. One can prove that the exchange property (iii) is equivalent the property that all bases have the same cardinality.

Lemma 2.4. *Let $M = (X, \mathcal{I})$. Then M is a matroid if and only if*

(i) $\emptyset \in \mathcal{I}$

(ii) If $Y \in \mathcal{I}$ and $Z \subseteq Y$ then $Z \in \mathcal{I}$

(iii') For all $Y \subseteq X$, all maximally independent subsets of Y have the same cardinality.

Again, we leave the proof for the exercises.

2.1 Examples of matroids

There are numerous examples of matroids. We will discuss two classes.

Graphic matroids

Let $G = (V, E)$ be an undirected graph. Then $(E, \mathcal{I})$ with $\mathcal{I} := \{F \subseteq E \mid F \text{ acyclic}\}$ is a matroid. Such matroids are called *graphic matroids.* We can also describe the rank function explicitly. For a set $Y \subseteq E$, let $H_1, \dots, H_k$ be connected components of the subgraph (V, Y). Then one can see that

$$\mathrm{rk}_M(Y) = |V| - k = \sum_{i=1}^{k}(|V(H_i)| - 1)$$

The linear matroid

Let V be a vectorspace (for example $\mathbb{R}^n$). Pick any subset $X := \{v_1, \dots, v_n\} \subseteq V$ and let

$$\mathcal{I} := \{Y \subseteq X \mid Y \text{ linearly independent}\}$$

Then $M = (X, \mathcal{I})$ is a matroid, called a *linear matroid.* In particular for $Y \subseteq X$ one has

$$\mathrm{rk}_M(Y) = \dim(\mathrm{span}(Y))$$

2.2 The Matroid Greedy Algorithm

We will now show that a Kruskal-type greedy algorithm can find a basis Y of a matroid that maximizes $\sum_{x \in Y} w(x)$ where $w : X \to \mathbb{R}$ is any weight function.

The Matroid Greedy Algorithm

Input: Matroid $M = (X, \mathcal{I})$ and weight function $w : X \to \mathbb{R}$.
Output: A basis Y maximizing $w(Y) := \sum_{x \in Y} w(x)$.

```
(1) Sort the elements in $X = \{e_1, \dots, e_n\}$ such that $w(e_1) \geq w(e_2) \geq \dots \geq w(e_n)$.
(2) Set $Y := \emptyset$
(3) For $i$ from 1 to $n$ do
        If $Y \cup \{e_i\} \in \mathcal{I}$ then update $Y := Y \cup \{e_i\}$.
```

Note that in case of a graphic matroid, the matroid greedy algorithm is identical to Kruskal's algorithm[1]. Now we will now prove that indeed for any matroid, the algorithm finds an maximum weight basis. First we prove a simple lemma which is a frequently used argument when dealing with matroids.

Lemma 2.5. *Let $M = (X, \mathcal{I})$ be a matroid and let $Y \subseteq Z \subseteq X$ with $Y \in \mathcal{I}$. Then there exists a basis $B \in \mathcal{I}$ of Z with $Y \subseteq B \subseteq Z$.*

[1]Note that since weights maybe negative, maximizing $w(Y)$ is equivalent to minimizing $-w(Y)$.

Proof. Start with $B := Y$. As long as B is not yet a basis of Z, there is some element $x \in Z \setminus B$ so that $B \cup \{x\} \in \mathcal{I}$. We iterate until B is a basis. $\square$

Now we come to the main theorem. Actually we will prove the stronger claim that if we had chosen any weaker property instead of (iii), then the greedy algorithm would fail for some weight function.

Theorem 2.6. *Suppose $M = (X, \mathcal{I})$ satisfies conditions (i) and (ii). Then the following is equivalent:*

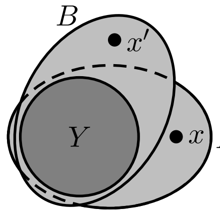
(A) M is a matroid.

(B) For any weight function $w : X \to \mathbb{R}$, the greedy algorithm finds a maximum weight basis.

Proof. $\boldsymbol{(A) \Rightarrow (B)}$. We fix a matrix M and a weight function $w : X \to \mathbb{R}$. We will generalize the analysis of Kruskal's algorithm.

Definition 2.7. We say an independent set $Y \in \mathcal{I}$ is *greedy* if there is a maximum weight basis B with $Y \subseteq B$.

We prove how to grow an independent set while keeping it greedy.
Claim I. *Suppose $Y \in \mathcal{I}$ is greedy and $x \in X \setminus Y$ so that $Y \cup \{x\} \in \mathcal{I}$ and $w(x)$ maximal. Then $Y \cup \{x\}$ is greedy.*
Proof of Claim. By assumption, Y is greedy, hence there is a maximum weight basis B with $B \supseteq Y$. If $x \in B$, then $Y \cup \{x\} \subseteq B$ and we are done! So suppose $x \notin B$. Construct basis B' with $Y \cup \{x\} \subseteq B' \subseteq B \cup \{x\}$ (see Lemma 2.5). As $|B'| = |B|$ there is a unique element $x' \in B \setminus B'$. By choice of x, $w(x) \geq w(x')$ and so $w(B') \geq w(B)$. Then $Y \cup \{x\}$ is greedy! $\square$



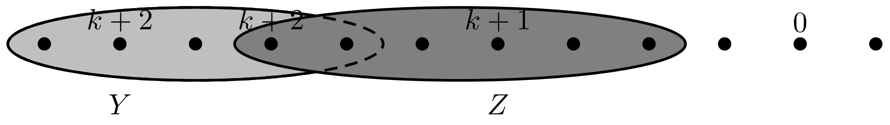
Let $Y_i \subseteq \{e_1, \dots, e_i\}$ be the set Y after iteration i in the algorithm.
Claim II. *For all i, Y_i is a basis of $\{e_1, \dots, e_i\}$.*
Proof of Claim II. By construction, each Y_i is an independent set. Suppose for the sake of contradiction that for some i, Y_i is not a basis of $\{e_1, \dots, e_i\}$. Then there is some $e_j \notin Y_i$ with $j \leq i$ so that $Y_i \cup \{e_j\} \in \mathcal{I}$. In particular we did not select e_j in iteration j but from $Y_j \subseteq Y_i$ we know that $Y_j \cup \{e_j\} \in \mathcal{I}$ which is a contradiction.
To finish the direction, note that Kruskal's algorithm always adds the maximum weight element $x \in X \setminus Y$ so that $Y \cup \{x\} \in \mathcal{I}$. Hence the final independent set Y will be greedy. By Claim II, the final independent set Y is also a basis of the matroid.

$\neg(A) \Rightarrow \neg(B)$. Now assume that M is not a matroid and in particular (iii) is not satisfied. That means there are $Y, Z \in \mathcal{I}$ with $k := |Y| < |Z|$ so that $Y \cup \{z\} \notin \mathcal{I}$ for all $z \in Z \setminus Y$. We claim that there is a weight function $w : X \to \mathbb{R}$ for which the greedy algorithm does not find the maximum weight basis. In fact, we define

$$w(x) := \begin{cases} k+2 & \text{if } x \in Y \\ k+1 & \text{if } x \in Z \setminus Y \\ 0 & \text{if } x \in X \setminus (Y \cup Z) \end{cases}$$

The greedy algorithm will pick all the elements in Y plus potentially some elements in $X \setminus (Y \cup Z)$ (which have value 0. Then the value of the greedy solution is

$$w(Y) = k(k+2) < (k+1)^2 \leq w(Z)$$



□

2.3 Exercises

Exercise 2.1.
Let $(X, \mathcal{I})$ be a pair with X finite and $\mathcal{I} \subseteq 2^X$. Consider the following properties:

(i) $\emptyset \in \mathcal{I}$

(ii) If $Y \in \mathcal{I}$ and $Z \subseteq Y$, then $Z \in \mathcal{I}$.

(iii) If $Y, Z \in \mathcal{I}$ and $|Y| < |Z|$, then $Y \cup \{x\} \in \mathcal{I}$ for some $x \in Z \setminus Y$.

(iv) For any $Y \subseteq X$, any two bases of Y have the same cardinality.

Assume that (i)+(ii) hold. Prove that (iii) and (iv) are equivalent.
Source: This exercise is taken from Schrijver [Sch17].

Exercise 2.2.
Let $M = (X, \mathcal{I})$ be a matroid. An inclusion-wise minimal dependent set $Y \subseteq X$ is called a *circuit*. Two elements $x, y \in X$ are called *parallel* if $\{x, y\}$ is a circuit. Show that if x and y are parallel and $Y \in \mathcal{I}$ with $x \in Y$, then $(Y \setminus \{x\}) \cup \{y\} \in \mathcal{I}$.
Source: This exercise is taken from Schrijver [Sch17].

Chapter 3

Polytopes, Polyhedra, Farkas Lemma and Linear Programming

Discrete optimization is typically about optimization with a *finite* number of candidate solutions. On the other hand, convex programs and linear programs contain in general an infinite number of points. We will later make the connection in Chapter 5. First we give an introduction into linear programming and the related geometry. Here we very closely follow Chapter 2 in Schrijver [Sch17].

3.1 Convex sets

We begin with a basic definition:

Definition 3.1. A set $C \subseteq \mathbb{R}^n$ is *convex* if for all $x, y \in C$ and $0 \leq \lambda \leq 1$ one has $\lambda x + (1-\lambda)y \in C$.

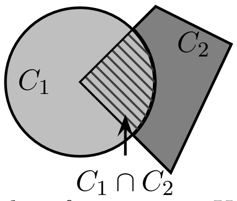
Intuitively, for any pair of points $x, y \in C$ in a convex set, the line segment connecting them must lie inside C.

convex not convex

The point $\lambda x + (1-\lambda)y$ is called a *convex combination* of x and y. It is not difficult to see that intersections of convex sets are again convex:

Lemma 3.2. *Let $C_i \subseteq \mathbb{R}^n$ be convex for $i \in I$. Then $\bigcap_{i \in I} C_i$ is convex.*

Proof. Let $x, y \in \bigcap_{i \in I} C_i$ and $0 \leq \lambda \leq 1$. For any $i \in I$ we have $x, y \in C_i$ and hence $\lambda x + (1-\lambda)y \in C_i$. Then $\lambda x + (1-\lambda)y \in \bigcap_i C_i$. $\square$



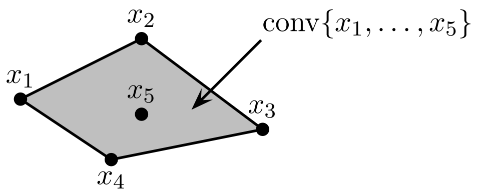
A consequence of this lemma is that for any set $X \subseteq \mathbb{R}^n$ there is a *unique* smallest set containing X, which we denote by

$$\text{conv}(X) := \bigcap_{C \supseteq X : C \text{ is convex}} C$$

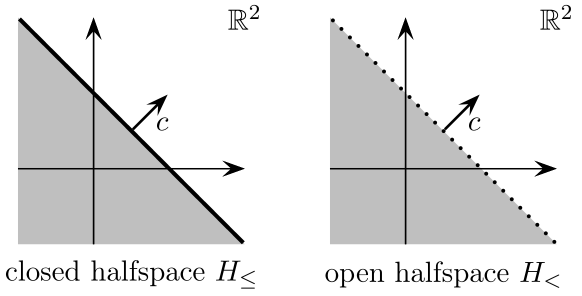
We can also give an alternative characterization of what the convex hull of X is:

Lemma 3.3. *For any $X \subseteq \mathbb{R}^n$ one has*

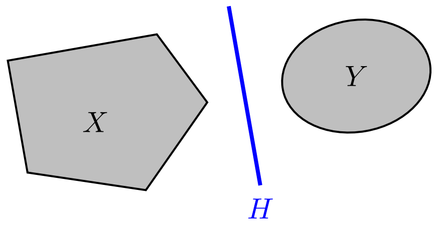
$$\text{conv}(X) = \left\{ \sum_{i=1}^{t} \lambda_i x_i \mid \begin{array}{c} t \in \mathbb{N},\ x_1, \ldots, x_t \in X, \\ \lambda_i \geq 0\ \forall i = 1, \ldots, t \textit{ and } \sum_{i=1}^{t} \lambda_i = 1 \end{array} \right\}$$



Definition 3.4. For $c \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ and $\delta \in \mathbb{R}$, the set $H = \{x \in \mathbb{R}^n \mid c^T x = \delta\}$ is called an *affine hyperplane*. The set $H_{\leq} := \{x \in \mathbb{R}^n \mid c^T x \leq \delta\}$ is a *(closed) half-space* and $H_{<} := \{x \in \mathbb{R}^n \mid c^T x < \delta\}$ is a *(open) half-space.*



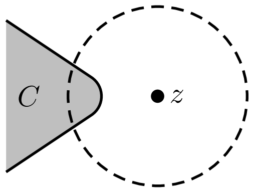
Definition 3.5. We say that a hyperplane H *separates* two sets $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^n$, if X and Y lie in different open halfspaces of H.



It is an important result that disjoint convex sets can always be separated by a hyperplane. The reader may note that this result has an infinite-dimensional analogue, called the *Hahn-Banach Theorem.* We denote $B(z,r) := \{x \in \mathbb{R}^n \mid \|x - z\|_2 \leq r\}$ as the Euclidean ball of radius r with center z.

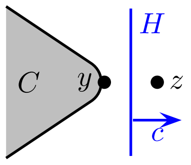
Theorem 3.6 (Separating Hyperplane Theorem)**.** *Let $C \subseteq \mathbb{R}^n$ be a closed convex set and $z \in \mathbb{R}^n \setminus C$. Then there is a hyperplane separating z and C.*

Proof. The claim is true if $C = \emptyset$, so suppose $C \neq \emptyset$.
Claim I. *The minimum* $\min\{\|z - y\|_2 : y \in C\}$ *is attained.*
Proof of Claim I. Fix $r > 0$ with $B(z,r) \cap C \neq \emptyset$. Then $\min\{\|z - y\|_2 : y \in C\} = \min\{\|z - y\|_2 : y \in B(z,r) \cap C\}$.

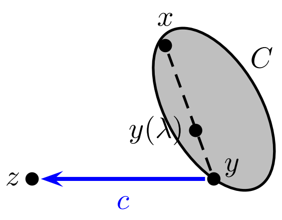


Moreover the set $B(z,r) \cap C$ is *compact* and the map $y \mapsto \|z - y\|_2$ is *continuous.* The claim then follows. □

Now to the main statement. Fix the point $y \in C$ minimizing $\|z-y\|_2$. We choose the hyperplane $H := \{x \in \mathbb{R}^n \mid c^T x = \delta\}$ with $c := z - y$ and $\delta := c^T(\frac{z+y}{2})$.



Claim II. *One has $c^T z > \delta$ and $c^T x < \delta \ \forall x \in C$*
Proof of Claim II. First, we verify that indeed $c^T z = \delta + \frac{1}{2}\|c\|_2^2 > \delta$. Next, suppose for sake of contradiction that there is an $x \in C$ with $c^T x \geq \delta$. Note that in particular one has $c^T x > c^T y$. Consider $y(\lambda) := (1 - \lambda)y + \lambda x$ for some parameter $0 \leq \lambda \leq 1$ that we determine later.



Then using that $c = z - y$ we have

$$\begin{aligned}\|z - y(\lambda)\|_2^2 &= \|c + \lambda(y-x)\|_2^2 \\ &= \|c\|_2^2 + 2\lambda \underbrace{c^T(y-x)}_{<0} + \lambda^2\|y-x\|_2^2 \overset{!}{<} \|c\|_2^2 = \|z-y\|_2^2\end{aligned}$$

if we pick $\lambda > 0$ small enough. $\square$

Definition 3.7. Vectors $x_1, \dots, x_m \in \mathbb{R}^n$ are *affinely independent* if

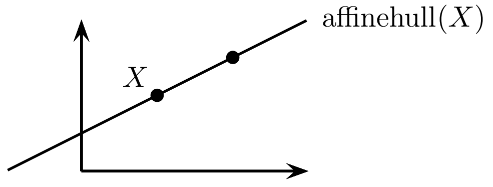
$$\Big(\sum_{i=1}^m \lambda_i x_i = 0 \text{ and } \sum_{i=1}^m \lambda_i = 0\Big) \Rightarrow \Big(\lambda_1 = \ldots = \lambda_m = 0\Big)$$

The relationship of affine to linear independence is as follows:

Lemma 3.8. *$x_1, \dots, x_m \in \mathbb{R}^n$ affinely independent $\Leftrightarrow$ $\binom{x_1}{1}, \dots, \binom{x_m}{1}$ are linearly independent.*

From Lemma 3.8 we can see that at most $n+1$ points in $\mathbb{R}^n$ can be affinely independent. Also note that affine invariance is invariant under translation. We define the unique smallest affine subspace containing X as

$$\text{affinehull}(X) := \Big\{ \sum_{i=1}^t \lambda_i x_i \mid x_1, \dots, x_t \in X \text{ and } \sum_{i=1}^t \lambda_i = 1 \Big\}$$

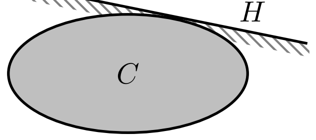


3.2 Polytopes and polyhedra

One can prove that any closed convex set is the intersection of a set of halfspaces:

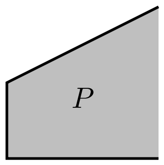
Lemma 3.9. *For any closed convex set $C \subseteq \mathbb{R}^n$ one has $C = \bigcap_{H: C \subseteq H_\leq} H_\leq$.*

Here the intersection is over all closed halfspaces that contain C.



We leave the proof for the exercises. It is not hard to see that possibly an infinite number of halfspaces are needed — see for example the Euclidean ball in dimension at least 2. We give a name to those convex sets where finitely many halfspace suffice:

Definition 3.10. The intersection of a finite number of closed half-spaces is called a *polyhedron.*



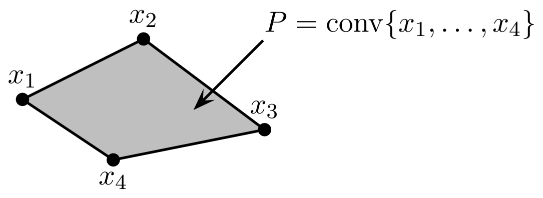
polyhedron $P \subseteq \mathbb{R}^2$

Note that every polyhedron is closed and convex. It will be useful to any polyhedron P can be repesented in the form

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\} = \left\{ x \in \mathbb{R}^n \mid \begin{array}{rcl} A_1^T x & \leq & b_1 \\ A_2^T x & \leq & b_2 \\ & \vdots & \\ A_m^T x & \leq & b_m \end{array} \right\}$$

for a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$ (here A_i is the ith row of matrix A interpreted as a column vector). We provide another definition:

Definition 3.11. $P \subseteq \mathbb{R}^n$ is a *polytope* if $P = \text{conv}\{x_1, \dots, x_t\}$ for a finite number of points $x_1, \dots, x_t \in \mathbb{R}^n$.



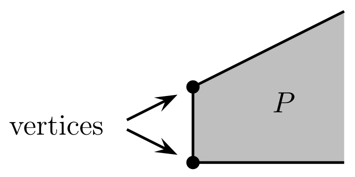
Assuming boundedness, both concepts are identical.

Theorem 3.12. *Let $P \subseteq \mathbb{R}^n$. Then P is a polytope if and only if P is a bounded polyhedron.*

For the proof we refer to Section 2.2 in Schrijver [Sch17].

Definition 3.13. Let $C \subseteq \mathbb{R}^n$ be a convex set. A point $z \in C$ is called *vertex* (or *extreme point*) if there are no $x, y \in C$ and $0 < \lambda < 1$ with $x \neq y$ so that $z = \lambda x + (1 - \lambda)y$.

In other words, a point $z \in C$ is a vertex if it is not a non-trivial convex combination of points of C.

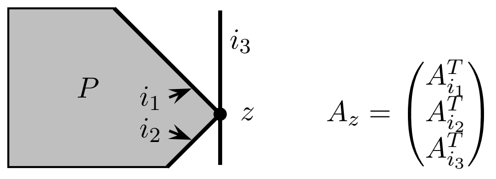


vertices of polyhedron $P \subseteq \mathbb{R}^2$

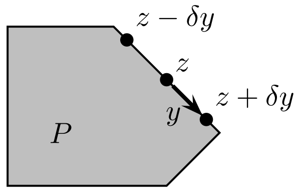
Characterization of vertices

We can characterize the vertices of a polyhedron using the constraint system:

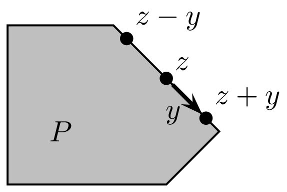
Lemma 3.14. *Suppose $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ be a polyhedron with $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and let $z \in P$. Let A_z be the submatrix of A consisting of those rows i s.t. $A_i^T z = b_i$. Then z is a vertex of $P \Leftrightarrow$ rank$(A_z) = n$.*


P
i_1
i_2
i_3
z
$A_z = \begin{pmatrix} A_{i_1}^T \\ A_{i_2}^T \\ A_{i_3}^T \end{pmatrix}$

Proof. We fix a point $z \in P$ and prove both directions separately.
Claim I. rank$(A_z) < n \Rightarrow z$ not a vertex
Proof of Claim I. Since rank$(A_z) < n$, there is a vector $y \in \ker(A_z)\setminus\{\mathbf{0}\}$ (meaning that $A_z y = \mathbf{0}$). For some small enough $\delta > 0$, we have $A_i^T(z + \delta y) \leq b_i$ and $A_i^T(z - \delta y) \leq b_i$ for any non-tight constraint (i.e. any constraint with $A_i^T z < b_i$). Then $z + \delta y, z - \delta y \in P$ and hence z is not a vertex □



Claim II. z not a vertex $\Rightarrow$ rank$(A_z) < n$
Proof of Claim II. Suppose z is not a vertex. By definition and convexity there is some $y \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ so that $z + y \in P$ and $z - y \in P$.



Consider an index i with $A_i^T z = b_i$. Then

$$(A_i^T(z+y) \leq b_i \quad \& \quad A_i^T(z-y) \leq b_i) \Rightarrow A_i^T y = 0$$

Hence $y \in \ker(A_z)$ and so $\text{rank}(A_z) < n$. □

We can prove a useful variant of this lemma. For an index set $I \subseteq [m]$, we denote A_I as the submatrix of A containing all row vectors.

Lemma 3.15. *Let $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ be a polyhedron with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. For each vertex $z \in P$, there is a subset $I \subseteq [m]$ with $|I| = n$ so that rank$(A_I) = n$ and $z = A_I^{-1} b_I$.*

Proof. By Lemma 3.14 there is a set $J \subseteq [m]$ so that $\text{rank}(A_J) = n$ and all inequalities in J are tight for z. Select any maximal subset $I \subseteq J$ so that $\{A_i\}_{i \in I}$ are linearly independent. Then $|I| = n$, A_I is non-singular and $A_I z = b_I$ can be inverted to $z = A_I^{-1} b_I$. □

As there are only $\binom{m}{n}$ many choices to select $I \subseteq [m]$ with $|I| = n$, we conclude the following:

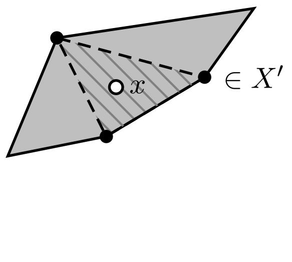
Corollary 3.16. *A polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ has at most $\binom{m}{n}$ vertices.*

For $n = 2$, Cor 3.16 gives an upper bound of $O(m^2)$ while it is not hard to see that in reality the tight bound must be m. And in fact, one can prove a stronger bound. The so-called *Upper Bound Theorem* by McMullen [McM70] shows that the number of vertices is indeed at most $O(m^{\lfloor n/2 \rfloor})$.

The following result is often quite useful:

Theorem 3.17 (Carathéodory's Theorem)**.** *If $X \subseteq \mathbb{R}^n$ and $x \in conv(X)$, then there is a subset $X' \subseteq X$ with $|X'| \leq n+1$ so that $x \in conv(X')$.*

The proof is similar to the argument in Lemma 3.14 and we defer it to the exercises. For example the statement says that any point x in a 2-dimensional polytope is contained in the convex hull of at most 3 of its vertices:

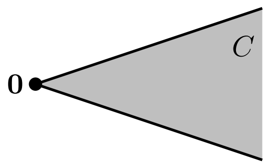


Convex cones

A special type of convex set is the following:

Definition 3.18. A set $C \subseteq \mathbb{R}^n$ is a *convex cone* if

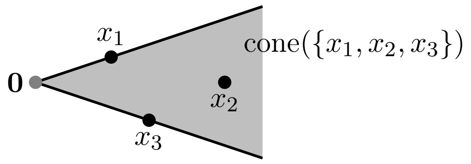
$$\lambda x + \mu y \in C \quad \forall x, y \in C \;\; \forall \lambda, \mu \geq 0$$



Again it will be useful to consider the unique minimal cone containing a set X:

Definition 3.19. For $X \subseteq \mathbb{R}^n$ we define

$$\text{cone}(X) \quad := \quad \Big\{ \quad \underbrace{\sum_{i=1}^{t} \lambda_i x_i}_{\substack{\text{conical combination}\\ \text{of } x_1,\ldots,x_t}} \quad \mid x_1, \ldots, x_t \in X;\; \lambda_1, \ldots, \lambda_t \geq 0 \Big\}$$



We can also provide a variant of Carathéodory's Theorem:

Theorem 3.20 (Carathéodory's Theorem — Conic version)**.** *If $X \subseteq \mathbb{R}^n$ and $x \in$ cone(X), then there is a subset $X' \subseteq X$ with $|X'| \leq n$ so that $x \in$ cone(X').*

3.3 Farkas Lemma

We come to a classical result in the theory of linear inequalities which says that if a linear system is infeasibly, then there must be an obvious certificate for that infeasibility. In the following we write $\dot{\vee}$ as the *exclusive or*. For example, if S and T are statements, then $S\dot{\vee}T$ means that exactly one of the two statements is true.

Lemma 3.21 (Farkas' Lemma 1902)**.** *For any $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ one has*

$$\big(\exists x \geq \mathbf{0} : Ax = b\big) \quad \dot{\vee} \quad \big(\exists y : y^T A \geq \mathbf{0} \text{ and } y^T b < 0\big).$$

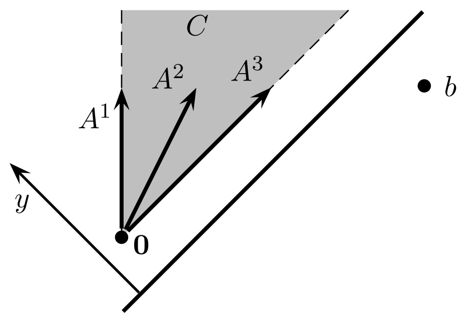
Proof. We prove both directions separately.
Claim I. *It is impossible that both systems have a solution.*
Proof of Claim I. Suppose for sake of contradiction that there are solutions x, y to both systems. Then

$$0 \leq \underbrace{(y^T A)}_{\geq \mathbf{0}} \underbrace{x}_{\geq \mathbf{0}} = y^T \underbrace{(Ax)}_{=b} = y^T b < 0$$

which is a contradiction. $\square$

Claim II. *Assume there is no $x \geq \mathbf{0}$ with $Ax = b$. Then there is a $y^T A \geq \mathbf{0}$ and $y^T b < 0$.*

Proof of Claim II. Let $A^1, \dots, A^n$ be the columns of A. Consider the cone $C := \text{cone}(\{A^1, \dots, A^n\}) = \{Ax \mid x \in \mathbb{R}^n_{\geq 0}\}$ spanned by those columns. Then C is convex and closed[1]. By assumption we know that $b \notin C$. Then by Theorem 3.6, there is a hyperplane $y^T c = \gamma$ separating C and b.



Separation here means that

$$\forall c \in C: \;\; y^T c > \gamma > y^T b$$

As $\mathbf{0} \in C$, we must have $\gamma < 0$. For any $x_i \geq 0$ we have $x_i A^i \in C$ and so $x_i \cdot y^T A^i > \gamma$. Then indeed it must be that $y^T A^i \geq 0$ for each $i \in [n]$. More compactly this means $y^T A \geq \mathbf{0}$ as claimed. $\square$

There are many variants of Farkas' Lemma. We mention a few here:

Proposition 3.22. *Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Then*

(I) $\exists x : Ax \leq b \quad \Leftrightarrow \quad \nexists y \geq \mathbf{0} : y^T A = \mathbf{0},\ y^T b < 0$

(II) $\exists x \geq \mathbf{0} : Ax = b \quad \Leftrightarrow \quad \nexists y : y^T A \geq \mathbf{0},\ y^T b < 0$

(III) $\exists x \geq \mathbf{0} : Ax \leq b \quad \Leftrightarrow \quad \nexists y \geq \mathbf{0} : y^T A \geq \mathbf{0},\ y^T b < 0$

[1]The latter fact actually requires some thought. Abbreviate cone(A) as the cone spanned by the columns of A and let us assume that rank$(A) = m$ otherwise the argument reduces to a subspace containing all the columns of A. Let us write A^J as the submatrix with columns indexed by $J \subseteq [n]$. Then by *Carathéodory's Theorem* (Theorem 3.20) one has $C = \bigcup_{|J|:\text{rank}(A^J)=m} \text{cone}(A^J)$ which is a finite union and so it suffices to prove that cone(B) is closed for some regular matrix $B \in \mathbb{R}^{m \times m}$. And indeed if $z^{(t)} \in \text{cone}(B)$ is a convergent sequence, then $B^{-1} z^{(t)} \geq \mathbf{0}$ and by linearity the limit $y := \lim_{t \to \infty} B^{-1} z^{(t)}$ exists and $y \geq \mathbf{0}$. Then $\|By - z^{(t)}\|_2 = \|By - BB^{-1} z^{(t)}\|_2 \leq \|B\|_{\text{op}} \|y - B^{-1} z^{(t)}\|_2 \to 0$ for $t \to \infty$.

Proof. We know from Lemma 3.21 that (II) is true. We will now prove (I) and leave (III) for the exercises. In fact, one has

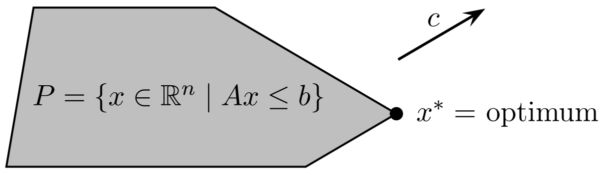
$$\begin{aligned}
& \exists x : Ax \leq b \\
\Leftrightarrow \quad & \exists x, s \geq \mathbf{0} : Ax + Is = b \\
\Leftrightarrow \quad & \exists x' \geq \mathbf{0}, x'' \geq \mathbf{0}, s \geq \mathbf{0} : Ax' - Ax'' + Is = b \\
\Leftrightarrow \quad & \exists \begin{pmatrix} x' \\ x'' \\ s \end{pmatrix} \geq \mathbf{0} : [A, -A, I] \begin{pmatrix} x' \\ x'' \\ s \end{pmatrix} = b \\
\overset{(II)}{\Leftrightarrow} \quad & \nexists y : y^T[A, -A, I] \geq \mathbf{0},\ y^T b < 0 \\
\Leftrightarrow \quad & \nexists y : y^T A \geq \mathbf{0},\ y^T(-A) \geq \mathbf{0},\ y^T I \geq \mathbf{0},\ y^T b < 0 \\
\Leftrightarrow \quad & \nexists y : y^T A = \mathbf{0},\ y \geq \mathbf{0},\ y^T b < 0 \qquad \square
\end{aligned}$$

□

3.4 Linear Programming

Now we come to the problem of optimizing linear functions over a polyhedron:

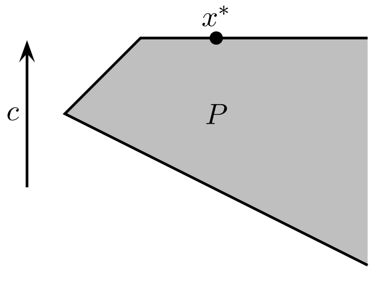
Definition 3.23. The optimization problem $\max\{c^T x \mid Ax \leq b\}$ is called *linear program* where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.



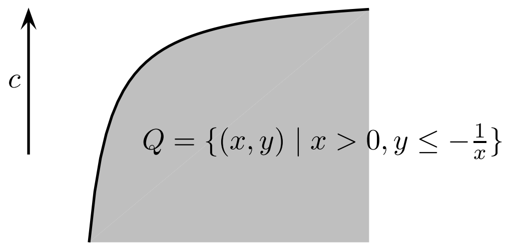
First we prove a somewhat technical result that an LP is either unbounded or an optimum is attained.

Lemma 3.24. *Let $P \subseteq \mathbb{R}^n$ be a polyhedron and $c \in \mathbb{R}^n$. If* $\sup\{c^T x \mid x \in P\} < \infty$ *then* $\max\{c^T x \mid x \in P\}$ *is attained.*

Before we come to the proof, we should note that the claim is trivial for polytopes as each polytope is compact and linear functions are continuous. But in the unbounded case the claim crucially relies on the fact that LPs have finitely many inequalities. In fact, the claim is false for arbitrary closed convex sets. See the figure below for an example.



unbounded P, bounded LP



$\max\{c^T x : x \in Q\}$ not attained

Proof. Set $\delta := \sup\{c^T x \mid x \in P\} < \infty$. Suppose for sake of contradiction that there is no $x \in P$ with $c^T x \geq \delta$. We can equivalently rewrite this as

$$\begin{aligned}
&\Leftrightarrow \quad \nexists x : \begin{pmatrix} A \\ -c^T \end{pmatrix} x \leq \begin{pmatrix} b \\ -\delta \end{pmatrix} \\
&\overset{(*)}{\Leftrightarrow} \quad \exists (y,\lambda) \geq \mathbf{0} : (y^T, \lambda)\begin{pmatrix} A \\ -c^T \end{pmatrix} = \mathbf{0}, (y^T,\lambda)\begin{pmatrix} b \\ -\delta \end{pmatrix} < 0 \\
&\Leftrightarrow \quad \exists y \geq \mathbf{0}, \lambda \geq 0 : y^T A = \lambda c^T, y^T b < \lambda\delta
\end{aligned}$$

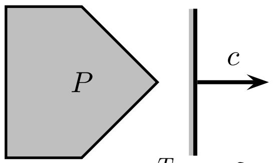
where in $(*)$ we have used Farkas variant I from Proposition 3.22. Then

$$\lambda\delta = \lambda \cdot \sup\{c^T x \mid x \in P\} = \sup\{y^T A x \mid x \in P\} \leq y^T b < \lambda\delta$$

which is a contradiction! □

Feasible inequalities

Next, we prove a very important property of polyhedra: an inequality is feasible for the polyhedron if and only if it can be obtained by adding up non-negative multiples of defining inequalities.



$c^T x \leq \delta$

Lemma 3.25. *Let $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and assume $P := \{x \in \mathbb{R}^n \mid Ax \leq b\}$ is non-empty. Then*

$$\Big(c^T x \leq \delta \ \ \forall x \in P\Big) \Leftrightarrow \Big(\exists y \geq \mathbf{0} : y^T A = c^T \text{ and } y^T b \leq \delta\Big)$$

Proof. We prove both directions separately.
($\Leftarrow$). Suppose there is a $y \geq \mathbf{0}$ with $y^T A = c^T$ and $y^T b \leq \delta$. Fix any $x \in P$. Then

$$c^T x = y^T A x \overset{y \geq \mathbf{0}, Ax \leq b}{\leq} y^T b \leq \delta$$

($\Rightarrow$). Suppose that there is <u>no</u> $y \geq \mathbf{0}$ with $y^T A = c^T, y^T b \leq \delta$. Then

$$\begin{aligned}
& \nexists y \geq \mathbf{0}, \lambda \geq 0 : y^T A = c^T,\ y^T b + \lambda = \delta \\
\Leftrightarrow \quad & \nexists (y,\lambda) \geq \mathbf{0} : (y^T, \lambda) \begin{pmatrix} A & b \\ \mathbf{0} & 1 \end{pmatrix} = (c^T, \delta) \\
\overset{\text{Farkas}}{\Leftrightarrow} \quad & \exists \begin{pmatrix} z \\ u \end{pmatrix} : \begin{pmatrix} A & b \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} z \\ u \end{pmatrix} \geq \begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix} \& (c^T, \delta) \begin{pmatrix} z \\ u \end{pmatrix} < 0 \\
\Leftrightarrow \quad & \exists z, u : Az + bu \geq \mathbf{0}, u \geq 0, c^T z + \delta u < 0 \\
\overset{\text{flip sign}}{\Leftrightarrow} \quad & \exists z, u \geq 0 : Az \leq bu, c^T z > \delta u
\end{aligned}$$

Here we have used Farkas Lemma (Lemma 3.21)[2]. We distinguish two cases:

- *Case $u = 0$:* Then $Az \leq \mathbf{0}$, $c^T z > 0$. By assumption $P \neq \emptyset$ and so we may fix a point $x_0 \in P$. Then for τ large enough one has $A(x_0 + \tau z) \leq b$ and $c^T(x_0 + \tau z) > \delta$ which is a contradiction!

- *Case $u > 0$:* After scaling (z, u) we may assume that $u = 1$. Then $Az \leq b$ and $c^T z > \delta$ which again is a contradiction!

$\square$

The strong duality theorem

Given a matrix $A \in \mathbb{R}^{m \times n}$, a right hand side $b \in \mathbb{R}^m$ and an objective function vector $c \in \mathbb{R}^n$, we consider the following pair of LPs:

$$\begin{aligned}
\text{primal LP:} \quad & \max\{c^T x \mid Ax \leq b\} \\
\text{dual LP:} \quad & \min\{y^T b \mid y^T A = c^T,\ y \geq \mathbf{0}\}
\end{aligned}$$

Any feasible solution y for the dual gives us a feasible ineqality for the primal of the form $c^T x = (y^T A) x \leq y^T b$. In other words, solutions to the dual prove an upper bound on the objective function value of the primal. A fundamental result in the theory of linear programming tells us that the best upper bound matches the value of the primal:

Theorem 3.26 (Duality Theorem)**.** *Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. Then*

$$\max\{c^T x \mid Ax \leq b\} = \min\{y^T b \mid y^T A = c^T,\ y \geq \mathbf{0}\}$$

provided both LPs are feasible.

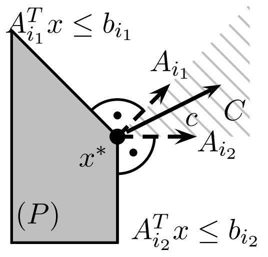
[2]Which states that $(\exists x \geq \mathbf{0} : Ax = b) \dot\vee (\exists y : y^T A \geq \mathbf{0} \ \&\ y^T b < 0)$

Proof. First we prove *weak duality*. Suppose x and y are feasible solutions to the primal and dual, resp. Then indeed

$$c^T x = (y^T A)x = y^T(Ax) \overset{y\geq \mathbf{0}, Ax\leq b}{\leq} y^T b$$

Next, we prove *strong duality*, meaning that there are indeed solutions x, y with $y^T b \leq c^T x$. We set $\delta := \sup\{c^T x \mid Ax \leq b\}$. If $\delta = \infty$, then by weak duality, the dual is infeasible which contradicts the assumption. So suppose $\delta < \infty$, i.e. the primal LP is bounded. We abbreviate the feasible region of the primal by $P := \{x \in \mathbb{R}^n \mid Ax \leq b\}$. Then $c^T x \leq \delta$ is a feasible inequality for P. Then by Lemma 3.25, we know that there is a $y \geq \mathbf{0}$ with $y^T A = c^T$ and $y^T b \leq \delta$. This is a solution for dual with objective value at most δ! □

We also want to give a more geometric interpretation of the Duality Theorem. Let x^* be the primal optimum solution to $\max\{c^T x \mid Ax \leq b\}$ (which exists by Lemma 3.24). Let $I := \{i \in [m] \mid A_i^T x^* = b_i\}$ be the *tight* inequalities for x^*. Consider the cone $C := \{\sum_{i\in I} A_i y_i \mid y_i \geq 0 \; \forall i \in I\}$ spanned by the normal vectors of the tight constraints.



If $c \notin C$ then by Farkas Lemma, there must be a vector $\lambda \in \mathbb{R}^n$ with $A_i^T \lambda \leq 0$ for all $i \in I$ and $c^T \lambda > 0$. Hence $x^* + \varepsilon\lambda$ for some $\varepsilon > 0$ is a feasible solution with $c^T(x^* + \varepsilon\lambda) > c^T x^*$ which contradicts the optimality of x^*. Hence indeed one must have that $c \in C$. Phrased differently there is a vector $y \geq \mathbf{0}$ with $y^T A = c^T$ and $y_i = 0 \; \forall i \notin I$. Note that this vector y is a feasible solution for the dual LP. We claim that indeed it is optimal. To see this, consider the *duality gap* which is

$$y^T b - c^T x^* \quad = \quad y^T b - \underbrace{y^T A}_{=c^T} x^* = \sum_{i=1}^m \underbrace{y_i}_{=0 \text{ if } i\notin I} \cdot \underbrace{(b_i - A_i^T x^*)}_{=0 \text{ if } i\in I} = 0$$

There are many possible primal-dual pairs of LPs. One state one popular variant and leave the proof for the exercises:

Theorem 3.27 (Duality Theorem II). *Let $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. Then*

$$\max\left\{c^T x \mid Ax \leq b, \; x \geq \mathbf{0}\right\} = \min\left\{y^T b \mid y^T A \geq c^T, y \geq \mathbf{0}\right\}$$

provided both LPs are feasible.

3.4.1 Complementary slackness

While finding optimum solutions is non-trivial, verifying optimality is simple even without explicitly computing the objective function values. Again we state two popular variants:

Theorem 3.28 (Complementary Slackness I). *Consider the systems*

$$(\textsc{Primal}) \quad \max\left\{c^T x \mid Ax \leq b\right\} \quad \textit{and} \quad (\textsc{Dual}) \quad \min\left\{y^T b \mid y^T A = c^T,\ y \geq \mathbf{0}\right\}$$

and suppose (x^*, y^*) *is a pair of feasible solutions for* (Primal) *and* (Dual). *Then the following is equivalent:*

(A) Both x^* *and* y^* *are optimal.*
(B) For all $i \in [m]$ *one has:* $y_i^* > 0 \Rightarrow A_i^T x^* = b_i$.

Proof. We can write the difference between the objective functions as

$$b^T y^* - c^T x^* = (y^*)^T b - (y^*)^T A x^* = \sum_{i=1}^{m} \underbrace{y_i^*}_{\geq 0} \cdot \underbrace{(b_i - A_i^T x^*)}_{\geq 0} \geq 0$$

Then this gap is actually equal to 0 if and only if for each i either $y_i^* = 0$ or $b_i - A_i^T x^* = 0$. □

Note that it is possible that for an optimum pair (x^*, y^*) one has $y_i^* = 0$ and $A_i^T x^* = b_i$ for some indices. However, the *strict complementary slackness theorem* says that there *exists* an optimum pair (x^*, y^*) so that for each i one has $(y_i^* = 0 \dot\vee A_i^T x^* = b_i)$. For example the points x^* and y^* an be taken as any interior point of the optimum face of the primal and dual, resp.

For the sake of completeness, we also list the complementary slackness condition for the pair in Theorem 3.27:

Theorem 3.29 (Complementary Slackness II). *Consider the systems*

$$(\textsc{Primal}) \quad \max\left\{c^T x \mid Ax \leq b,\ x \geq \mathbf{0}\right\} \quad \textit{and} \quad (\textsc{Dual}) \quad \min\left\{y^T b \mid y^T A \geq c^T, y \geq \mathbf{0}\right\}$$

and suppose (x^*, y^*) *is a pair of feasible solutions for* (Primal) *and* (Dual). *Then the following is equivalent:*

(A) Both x^* *and* y^* *are optimal.*
(B) For all $i \in [m]$ *one has* $y_i^* > 0 \Rightarrow A_i^T x^* = b_i$ *and for all* $j \in [n]$ *one has* $x_j^* > 0 \Rightarrow (y^*)^T A^j = c_j$.

3.5 Exercises

Exercise 3.1.
Let $C \subseteq \mathbb{R}^n$. Prove that C is a closed convex set if and only if there is a collection $\mathcal{F}$ of closed affine halfspaces so that $C = \bigcap_{H \in \mathcal{F}} H$.
Source: This exercise is taken from Schrijver [Sch17].

Exercise 3.2.
Goal of this exercise is to prove Carathéodory's Theorem[3].

(a) Consider $P := \{x \in \mathbb{R}^n \mid Ax = b, x \geq \mathbf{0}\}$ for $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Fix a point $y \in P$ that minimizes $|\mathrm{supp}(y)|$ where $\mathrm{supp}(y) := \{j \in \{1, \ldots, n\} \mid y_j \neq 0\}$. Prove that $|\mathrm{supp}(y)| \leq m$.

(b) Let $X \subseteq \mathbb{R}^n$. Prove that for any $x \in \mathrm{conv}(X)$, there is a subset $X' \subseteq X$ with $|X'| \leq n+1$ so that $x \in \mathrm{conv}(X')$.

Exercise 3.3.
Let $A \in \mathbb{R}^{m \times n}$ and let $b \in \mathbb{R}^m$ with $m \geq n+1$. Suppose that $Ax \leq b$ has no solution x. Prove that there are indices $i_0, \ldots, i_n$ so that the system $A_{i_0}^T x \leq b_{i_0}, \ldots, A_{i_n}^T x \leq b_{i_n}$ has no solution x.
Source: This exercise is taken from Schrijver [Sch17].

[3] In particular you are not allowed to use Carathéodory's Theorem itself in the argument.

Chapter 4

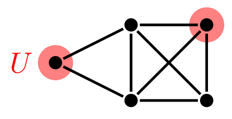
Matchings and Covers in Bipartite Graphs

In this chapter, we will get more familiar with graph theory and popular graph optimization questions. Again, we follow Chapter 3 in Schrijver [Sch17].

4.1 Matchings, stable sets and vertex cover

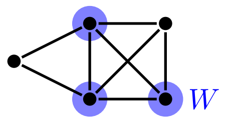
In the following, let $G = (V, E)$ be an undirected graph.

Definition 4.1. $U \subseteq V$ is a *stable set / independent set* if for all $i, j \in U$ one has $\{i, j\} \notin E$.



stable set U

Definition 4.2. $W \subseteq V$ is a *vertex cover* if $e \cap W \neq \emptyset$ for all $e \in E$.

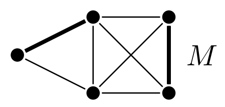


vertex cover W

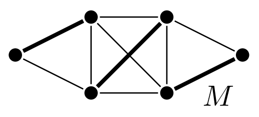
One can make the observation that these notions are complementary:

Fact 4.3. *U is stable set in $G \Leftrightarrow V \setminus U$ is vertex cover.*

Definition 4.4. $M \subseteq E$ is a *matching* in G if $e \cap e' = \emptyset$ for any distinct edges $e, e' \in M$. A matching is *perfect* if $|M| = \frac{1}{2}|V|$.



non-perfect matching



perfect matching

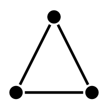
We will be particularly interested in the "best" stable set, vertex cover and matching in a given graph. For this sake, we define

$$\begin{aligned} \alpha(G) &:= \max\{|C| : C \text{ stable set in } G\} = \text{stability } \# \\ \tau(G) &:= \min\{|W| : W \text{ vertex cover}\} = \text{vertex cover } \# \\ \nu(G) &:= \max\{|M| : M \text{ matching}\} = \text{matching } \# \end{aligned}$$

One can easily observe the following:

Fact 4.5. *In any undirected graph G one has $\nu(G) \leq \tau(G)$.*

One can also see that this inequality may be strict for some graphs:



graph with $\nu(G) = 1 < 2 = \tau(G)$

4.2 M-augmenting paths

The first step will be to characterize when a matching in a graph is maximal. For this we need the following concept:

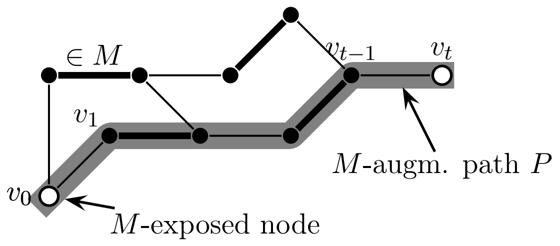
Definition 4.6. Let M be a matching in $G = (V, E)$. A path $P = (v_0, \ldots, v_t)$ in G is *M-augmenting* if

(i) t is odd

(ii) $\{v_1, v_2\}, \{v_3, v_4\}, \ldots, \{v_{t-2}, v_{t-1}\} \in M$

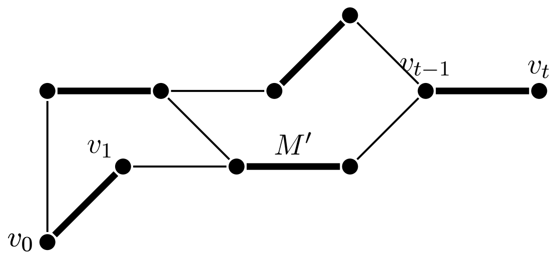
(iii) $v_0, v_t \notin V(M)$

Vertices that are not covered by the matching M (that means vertices in $V(M) \setminus M$) are also called *M-exposed*.



For sets A, B, let $A\Delta B := (A \setminus B) \cup (B \setminus A)$ be the *symmetric difference*. We can now see the usefulness of an M-augmenting path: flipping all the edges along the path will lead to a matching that has one edge more.

Fact 4.7. *If P is an M-augmenting path in G, then $M' := M\Delta E(P)$ is a matching in G with $|M'| = |M| + 1$.*



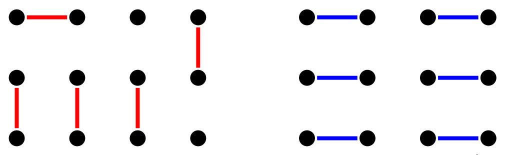
matching $M' = M\Delta E(P)$

We can prove the following:

Theorem 4.8. *Let $G = (V, E)$ be an undirected graph with matching $M \subseteq E$. Either M is a matching of maximum cardinality, or there exists an M-augmenting path.*

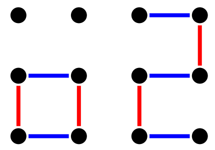
Proof. From Fact 4.7 we know that if there is an M-augmenting path, then M not optimal. It remains to prove the reverse direction: M not maximal $\Rightarrow \exists M$-augmenting path.

So, suppose there is a matching $M' \subseteq E$ with $|M'| > |M|$.



matching M matching M'

Consider the graph $G' := (V, M\Delta M')$ formed by taking the symmetric difference of M and M'. Then the vertices in G' must have degrees in $\{0, 1, 2\}$. That means the connected components of G' are either paths (which allows singletons as these are 0-length paths) or circuits.


difference $M \Delta M'$

As $|M'| > |M|$, there is a component with more edges from M' than from M. This component has to be a path with endpoints in M', meaning it is an M-augmenting path. $\square$

4.3 Kőnig's Theorem

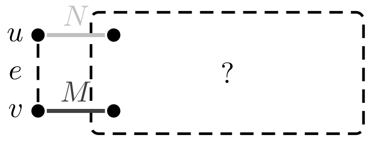
Next, we prove one of the most famous classical results in graph theory.

Theorem 4.9 (Kőnig 1931)**.** *In any bipartite graph G, one has $\nu(G) = \tau(G)$.*

Proof. We know from Fact 4.5 that $\nu(G) \leq \tau(G)$ in any graph. Hence it suffices to prove that $\nu(G) \geq \tau(G)$. The key argument is a structural insight on bipartite graphs:

Claim I. *Let $G = (V, E)$ be bipartite with $|E| \geq 1$. Then G has a vertex that is covered by* every *maximum matching.*

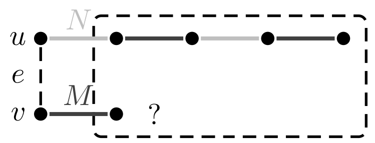
Proof of Claim I. Suppose for the sake of contradiction that the claim is false. Fix any edge $e = \{u, v\} \in E$. Let M be a maximum matching with $u \notin V(M)$. As M is maximal and $\{u, v\} \in E$ this means that $v \in V(M)$. Similarly there is a maximum matching N with $v \notin V(N)$ and $u \in V(N)$.



Let P be the component of $(V, M \Delta N)$ containing u. Then one can see that P must be a path of even length (in fact, degrees in P must be in $\{0, 1, 2\}$ but P cannot be a circuit since u is M-exposed and P cannot have odd length since then P would be M-augmenting). We make a case distinction as to what the other endpoint of P is:

- *Case: P contains v.* Then $P \cup \{e\}$ is an odd length cycle which contradicts that the graph is bipartitie.

- *Case: P does not contain v.* Then $P \cup \{u, v\}$ is N augmenting, which contradicts the maximality of N.



Either way we arrive at a contradiction. □

Now we can finish the main claim.

Claim II. *Any bipartite graph $G = (V, E)$ contains a vertex cover of size $\nu(G)$.*

Proof of Claim II. We prove the claim by induction over $|V|$. Let u be the vertex from Claim I that contained in every maximum matching. Then $G' := G \setminus u$ has $\nu(G') = \nu(G) - 1$. By induction, let U' be vertex cover for G' of size $|U'| = \nu(G') = \nu(G) - 1$. Then $U' \cup \{v\}$ is vertex cover for G of size $\nu(G)$. □

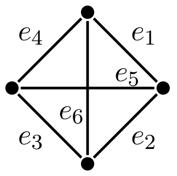
The concept of M-augmenting paths also provides a natural algorithm that iteratively finds M-augmenting paths until a maximum matching is reached. The algorithm is simple in bipartite graphs and more involved (but still polynomial time) in general graphs. For bipartite graphs, one can also find minimum vertex covers in polynomial time while this is **NP**-hard problem in general graphs. We will not go into details here but later discuss a more general concept in the chapter on flows.

4.4 The Matching Polytope of a Graph

In this section we want to motivate a connection between linear programming that we investigate in the next chapter. Fix a graph $G = (V, E)$. For a subset $M \subseteq E$, we define the *characteristic vector / incidence vector* as $\chi_M \in \mathbb{R}^E$ with

$$\chi^M(e) = \begin{cases} 1 & \text{if } e \in M \\ 0 & \text{if } e \notin M \end{cases}$$

For example in the following graph



we have

$$\chi^{\{e_1,e_3\}} = \begin{pmatrix}1\\0\\1\\0\\0\\0\end{pmatrix}, \quad \chi^{\{e_5,e_6\}} = \begin{pmatrix}0\\0\\0\\0\\1\\1\end{pmatrix}, \quad \chi^{\emptyset} = \begin{pmatrix}0\\0\\0\\0\\0\\0\end{pmatrix}$$

Definition 4.10. For an undirected graph G, the *matching polytope* is

$$P_{\text{matching}}(G) := \text{conv}\{\chi^M \in \mathbb{R}^E \mid M \subseteq E \text{ is matching}\}$$

We note that $P_{\text{matching}}(G)$ is an $|E|$-dimensional polytope whose vertices are from $\{0,1\}^E$. We know from Theorem 3.12 that $P_{\text{matching}}(G)$ can be described with finitely many inequalities. But it is less obvious what those constraints are. A natural guess might be

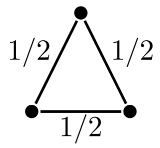
$$Q_{\text{matching}}(G) \quad := \quad \left\{ x \in \mathbb{R}^E \mid \begin{array}{rcll} \sum_{e: v \in e} x_e & \leq & 1 & \forall v \in V \\ x_e & \geq & 0 & \forall e \in E \end{array} \right\}$$

It is not hard to check that indeed

$$P_{\text{matching}}(G) \subseteq Q_{\text{matching}}(G) \quad \text{and} \quad P_{\text{matching}}(G) = \text{conv}\{Q_{\text{matching}}(G) \cap \mathbb{Z}^E\}$$

On the other hand, for arbitrary graphs the inclusion might be strict.

Example 4.11. Let G be a triangle. Then $x^* := \begin{pmatrix} 1/2 \\ 1/2 \\ 1/2 \end{pmatrix}$ lies in $Q_{\text{matching}}(G)$ but not in $P_{\text{matching}}(G)$.



Note that in this case

$$P_{\text{matching}}(G) = \text{conv}\left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

However, we will see that for all bipartite graphs, $Q_{\text{matching}}(G)$ contains all required inequalities:

Theorem 4.12. *If G is bipartite, then $P_{\text{matching}}(G) = Q_{\text{matching}}(G)$.*

To draw the connection back to optimization, assuming this result, the following algorithm finds a matching M in a bipartite graph maximizing $\sum_{e \in M} w_e$ where $w: E \to \mathbb{R}$ is some weight function:

(1) Find an optimum vertex solution x^* for the LP

$$\max\Big\{ \sum_{e \in E} w_e x_e \mid \sum_{e: v \in e} x_e \leq 1 \; \forall v \in V, \;\; x_e \geq 0 \; \forall e \in E \Big\}$$

(2) Return $\{e \in E \mid x^*_e = 1\}$

4.5 Exercises

Exercise 4.1.
Prove that in a matrix, the maximum number of non-zero entries with no two in the same line (=row or column), is equal to the minimum number of lines that include all nonzero entries.
Source: This exercise is taken from Schrijver [Sch17].

Example: Consider the following matrix

$$A = \begin{pmatrix} * & 0 & 0 \\ * & * & * \\ * & 0 & 0 \end{pmatrix}$$

where $*$ means any non-zero entry. Then the non-zero entries can be covered by two lines (2nd row and first column) and this is optimal. Also we can select at most 2 non-zero entries that have all rows and columns distinct — for example the two entries (1,1) and (2,2) on the diagonal.

Exercise 4.2.
Let $\mathcal{A} = (A_1, \ldots, A_n)$ be a family of subsets of some finite set X. Prove that $\mathcal{A}$ has an SDR if and only if

$$\Big| \bigcup_{i \in I} A_i \Big| \geq |I|$$

for each subset $I \subseteq \{1, \ldots, n\}$.
Remark: Recall that an SDR is an injective map $\pi : [n] \to X$ with $\pi(i) \in A_i$ for all $i = 1, \ldots, n$.
Source: This exercise is taken from Schrijver [Sch17].

Exercise 4.3.
A matrix is called doubly-stochastic if it is nonnegative and each row sum and each column sum is equal to 1. A matrix is palled a permutation matrix if each entry is 0 or 1 and each column and each row contains exactly one 1.

i) Show that for each doubly stochastic matrix $A = (a_{ij})_{i,j=1,\ldots,n}$, there exists a permutation $\pi \in S_n$ so that $a_{i,\pi(i)} \neq 0$ for all $i = 1, \ldots, n$.

ii) Derive that each doubly stochastic matrix is a convex linear combination of permutation matrices.

Hint: Set up a bipartite graph and prove the claim using König's Theorem.
Source: This exercise is taken from Schrijver [Sch17].

Chapter 5

Integer Linear Programming and Totally Unimodular Matrices

For this chapter we mostly follow Chapter 8 in Schrijver [Sch17].

5.1 Integer linear programming

Probably, the most versatile problem in combinatorial optimization is the following:

Definition 5.1. A problem of the form

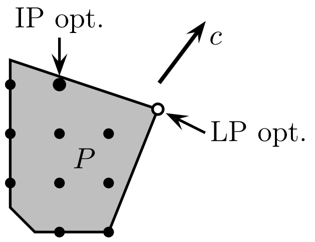
$$\max\left\{c^T x \mid Ax \leq b; x \in \mathbb{Z}^n\right\}$$

is called an *integer linear program*.

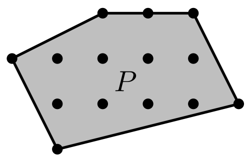
In general, solving such a problem is **NP**-hard, hence our focus here will be on substantial special cases that can be solved in polynomial time. First, we can make the observation that one always has

$$\max\{c^T x \mid Ax \leq b, x \in \mathbb{Z}^n\} \leq \max\{c^T x \mid Ax \leq b\}$$

(if we think of an infeasible problem as having value $-\infty$). Even in dimension 2 it is easy to find examples where the inequality is strict:



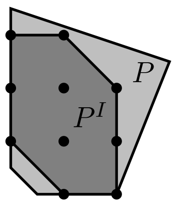
Definition 5.2. A polytope P is *integer / integral*, if all vertices are integer vectors.



integral polytope

For example $P_{\text{matching}}(G)$ is integral by construction.

Definition 5.3. For a polyhedron $P \subseteq \mathbb{R}^n$ we define the *integer hull* as $P^I := \text{conv}(P \cap \mathbb{Z}^n)$.



We leave the proof of the following as an exercise:

Lemma 5.4. *For a polytope $P \subseteq \mathbb{R}^n$, the following is equivalent*

(A) P integer

(B) $P^I = P$

(C) For all $c \in \mathbb{R}^n$ one has $\max\{c^T x \mid x \in P\} = \max\{c^T x \mid x \in P \cap \mathbb{Z}^n\}$.

Note that for the moment we have restricted the concept of integrality to bounded polyhedra where definitions are simpler.

5.2 Totally unimodular matrices

Next, we approach the question when we can guarantee that a polytope P is integral. The following definition is crucial:

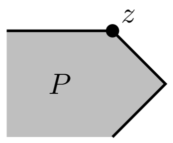
Definition 5.5. A matrix $A \in \mathbb{R}^{m \times n}$ is called *totally unimodular* (TU) if each square submatrix of A has determinant in $\{-1, 0, 1\}$.

If a matrix A is TU, then in particular one must have $A \in \{-1, 0, 1\}^{m \times n}$ (though this is not sufficient). A simple example is that the identity matrix I_n is totally unimodular. We recall the following fact from linear algebra:

Lemma 5.6 (Cramer's Rule)**.** *Let $B \in \mathbb{Z}^{n \times n}$ be invertible. Then $B^{-1} = \frac{1}{\det(B)} C$ where $C \in \mathbb{Z}^{n \times n}$.*

Theorem 5.7. *If $A \in \mathbb{R}^{m\times n}$ is TU and $b \in \mathbb{Z}^m$, then every vertex of the polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \le b\}$ is integral.*

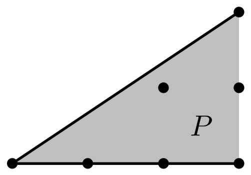
Proof. Let z be a vertex of P. We have proven in Lemma 3.15 that there is $I \subseteq [m]$ with $|I| = n$ so that $\text{rank}(A_I) = n$ and $z = A_I^{-1}b_I$. Since A is TU we know that $\det(A_I) \in \{-1, 1\}$. Then $A_I^{-1} \in \mathbb{Z}^{n\times n}$ by Cramers Rule and $b_I \in \mathbb{Z}$ by assumption. Hence $z = A_I^{-1}b_I \in \mathbb{Z}^n$.



□

Note that if in Theorem 5.7 we know that P is bounded, then in fact P itself is integer by Lemma 5.4. But an unbounded polyhedron may not even have any vertices and the statement of Theorem 5.7 might be vacuous. First we extend the definition of being integer to unbounded polyhedra:

Definition 5.8. A polyhedron P is *integer* if for all $c \in \mathbb{R}^n$ where $\max\{c^Tx \mid x \in P\} < \infty$, the maximum is attained by some integer vector.

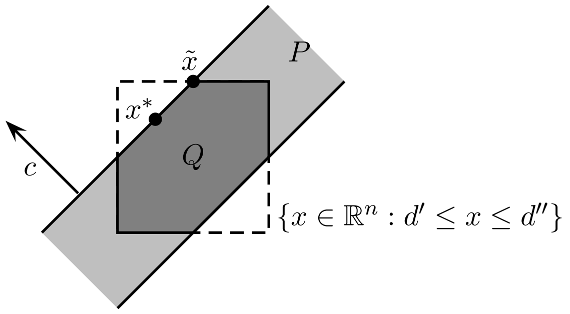


Lemma 5.9. *Let $A \in \mathbb{R}^{m\times n}$ be TU and $b \in \mathbb{Z}^m$. Then the polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \le b\}$ is integral.*

Proof. Fix a $c \in \mathbb{R}^n$ so that $\max\{c^Tx \mid x \in P\} < \infty$ and let x^* be an optimum solution (possibly fractional). Choose $d', d'' \in \mathbb{Z}^n$ so that $d' \le x^* \le d''$. Consider the polytope

$$Q := \{x \in \mathbb{R}^n \mid Ax \le b, d' \le x \le d''\} = \left\{ x \in \mathbb{R}^n \mid \begin{pmatrix} A \\ -I_n \\ I_n \end{pmatrix} x \le \begin{pmatrix} b \\ -d' \\ d'' \end{pmatrix} \right\}$$

We can verify that if A is TU then also the stacked matrix $\left(\begin{smallmatrix} A \\ -I_n \\ I_n \end{smallmatrix}\right)$ is TU. Moreover, the right hand side vector defininingdefinining Q is integral. Since Q is bounded, the must be an optimum solution $\tilde{x}$ for $\max\{c^Tx \mid x \in Q\}$ that is a vertex.



By Theorem 5.7 we have $\tilde{x} \in \mathbb{Z}^n$. As $x^* \in Q$, one has $c^T\tilde{x} \geq c^T x^*$. □

This result will suffice for us to handle the case of unbounded polyedra[1]. Combining total unimodularity with strong duality we obtain the following:

Lemma 5.10. *Let $A \in \mathbb{R}^{m\times n}$ TU, $b \in \mathbb{Z}^m$, $c \in \mathbb{Z}^n$. Both LPs*

$$\max\{c^T x \mid Ax \leq b\} = \min\{y^T b \mid y^T A = c^T,\ y \geq \mathbf{0}\}$$

have integral optimum solutions (assuming the LPs are feasible).

Proof. By Lemma 5.9, the primal has an integral optimum solution. The dual can be written in the form

$$\min\left\{ y^T b \mid \begin{pmatrix} A^T \\ -A^T \\ -I_m \end{pmatrix} y \leq \begin{pmatrix} c^T \\ -c^T \\ \mathbf{0} \end{pmatrix} \right\}$$

We note that also the stacked matrix $\left(\begin{smallmatrix} A^T \\ -A^T \\ -I_m \end{smallmatrix}\right)$ is TU and the right hand side is integral by assumption. Then also the dual has an integral optimum solution. □

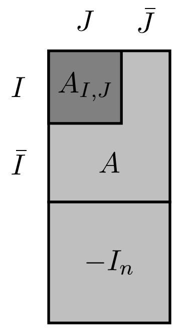
We can also prove that there is no property guaranteeing integrality that is more general than total unimodularity without depending on the right hand side vector.

Theorem 5.11 (Hoffman-Kruskal Theorem)**.** *Let $A \in \mathbb{Z}^{m\times n}$. Then A is TU $\Leftrightarrow \forall b \in \mathbb{Z}^m$, $P_b = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq \mathbf{0}\}$ is integer.*

The standard proof for the Hoffman-Kruskal Theorem takes a detour via the concept of *unimodularity*. To avoid this, we follow the exposition in Korte and Vygen [KV12].

[1]The proper generalization for the case of arbitrary polyhedra is as follows. A polyhedron P is called *rational* if it can be written as $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ so that A and b have only rational entries. Then for a rational polyhedron P, the following conditions are equivalent (and either can be used as definition for P being integer): (A) each minimal face of P contains an integer point; (B) $P^I = P$; (C) For all $c \in \mathbb{R}^n$ with $\max\{c^T x \mid x \in P\} < \infty$, one has $\max\{c^T x \mid x \in P\} = \max\{c^T x \mid x \in P \cap \mathbb{Z}^n\}$. For details we refer to Chapter 16 of Schrijver [Sch99].

Proof. We already proved the direction ($\Rightarrow$) in Lemma 5.9 hence it remains to show ($\Leftarrow$). Let $I \subseteq [m]$ be row indices and let $J \subseteq [n]$ be column indices with $k := |I| = |J|$. We write $A_{I,J}$ as the corresponding $k \times k$ submatrix of A and we need to prove that $\det(A_{I,J}) \in \{-1, 0, 1\}$. We assume that $\det(A_{I,J}) \neq 0$ since otherwise there is nothing to show. Note that the contraint matrix of P_b looks then as follows:



Claim. *For any $v \in \mathbb{Z}^k$, $A_{I,J}^{-1} v \in \mathbb{Z}^k$.*

Proof of Claim. Fix $v \in \mathbb{Z}^k$. We will construct a right hand side vector b and a point z so that z is a vertex of P_b and $A_{I,J}$ is part of the basis defining z. First, let $y \in \mathbb{Z}^k$ be a vector so that $z_J := y + A_{I,J}^{-1} v > \mathbf{0}$. We set $z_{\bar{J}} := \mathbf{0}$. Set $b_I := A_I z = A_{I,J} z_J = A_{I,J} y + A_{I,J} A_{I,J}^{-1} v = A_{I,J} y + v$. We pick $b_{\bar{I}}$ integral so that $A_{\bar{I}} z < b_{\bar{I}}$. Then z is a vertex of P_b and by assumption z is integral which implies that $A_{I,J}^{-1} v$ is integral. □

From the claim we know that $A_{I,J}^{-1} e_i \in \mathbb{Z}^k$ for all $i = 1, \ldots, k$ and so $A_{I,J}^{-1} \in \mathbb{Z}^{k \times k}$. From $\det(A_{I,J}), \det(A_{I,J}^{-1}) \in \mathbb{Z}^{k \times k}$ and $\det(A_{I,J}) \cdot \det(A_{I,J}^{-1}) = 1$ we then know that $\det(A_{I,J}) \in \{-1, 1\}$. □

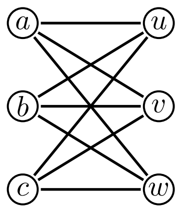
5.3 TU matrices from bipartite graphs

We will now come to the first natural family of matrices that are totally unimodular.

Definition 5.12. The *node-edge incidence matrix* of graph G is the matrix $A \in \{0,1\}^{V \times E}$ with

$$A_{v,e} = \begin{cases} 1 & v \text{ incident to } e \\ 0 & \text{otherwise} \end{cases}$$

An example for a node edge incidence matrix can be found below:



graph G

$$\begin{array}{cc|ccccccccc|}
 & & \{a,u\} & \{a,v\} & & & \textbf{edges} & & & & \\
\hline
 & a & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & b & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
 & c & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
\textbf{nodes} & u & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
 & v & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
 & w & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
\hline
\end{array}$$

node edge incidence matrix A

We can prove that incidence matrices of bipartite graphs are totally unimodular (and again, these are the only undirected graphs with TU incidence matrices).

Theorem 5.13. *Let $G = (V, E)$ be a graph with incidence matrix A_G. Then G is bipartite $\Leftrightarrow$ A_G is TU.*

Proof. We prove both directions separately.
Claim I. *G bipartite $\Rightarrow$ A_G TU.*
Proof of Claim I. Let B be a $k \times k$ submatrix of A_G. We prove by induction over k that $\det(B) \in \{-1, 0, 1\}$. The case $k = 1$ is trivial, hence suppose $k \geq 2$. We make a case distinction:

- *Case: B has a column with only 0's.* Then $\det(B) = 0$.
- *Case: B has a column with exactly one* 1. After permuting rows and columns,

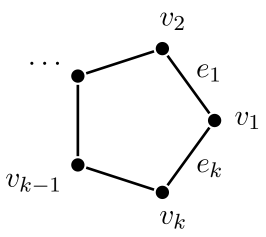
$$B = \begin{pmatrix} 1 & * \\ \mathbf{0} & B' \end{pmatrix}$$

  with $\det(B') \in \{-1, 0, 1\}$ by induction. Then $\det(B) = 1 \cdot \det(B') \in \{-1, 0, 1\}$.
- *Case: Every column of B has exactly 2 ones.* Partition the row indices $I \cup J$ so that each column has exactly one row in I and one row in J (uses bipartiteness!)

$$\begin{array}{c} I \\ \\ J \end{array} \left[\begin{array}{cc} 1 & \cdots \\ \hdashline 1 & \cdots \end{array}\right]$$

  Then $\sum_{i \in I} B_i - \sum_{i \in J} B_j = \mathbf{0}$. Hence $\det(B) = 0$. $\square$

Claim II. *G not bipartite $\Rightarrow$ A_G not TU.*
Proof of Claim II. Consider an odd cyle H in G and denote its vertices by $v_1, \ldots, v_k$ and its edges $e_1, \ldots, e_k$. Then A_H is a square submatrix of A_G. After permuting rows and columns A_H is of the following form:



$$\begin{array}{c|ccccc} & e_1 & \cdots & & & e_k \\ \hline v_1 & 1 & & & & 1 \\ v_2 & 1 & 1 & & & \\ \vdots & & 1 & 1 & & \\ & & & 1 & 1 & \\ v_k & & & & 1 & 1 \end{array} = A_H$$

One may check that indeed $|\det(B)| = 2$. □

Now, we can finally prove the claim that we made at the end of Chapter 4.

Theorem 5.14. *If G is bipartite, then $P_{matching}(G) = Q_{matching}(G)$.*

Proof. Recall that

$$\begin{aligned} Q_{\text{matching}}(G) &:= \left\{ x \in \mathbb{R}^E \mid \begin{array}{rcll} \sum_{e:v\in e} x_e & \leq & 1 & \forall v \in V \\ x_e & \geq & 0 & \forall e \in E \end{array} \right\} \\ &= \left\{ x \in \mathbb{R}^E \mid \begin{pmatrix} A_G \\ -I_n \end{pmatrix} x \leq \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix} \right\} \end{aligned}$$

The matrix $\binom{A_G}{-I_n}$ is TU and hence every vertex of $Q_{\text{matching}}(G)$ is integral. Then $Q_{\text{matching}}(G) = \text{conv}\{Q_{\text{matching}} \cap \mathbb{Z}^E\} = P_{\text{matching}}(G)$. □

Recall that Kőnig's Theorem (Theorem 4.9) can be seen as a type of *duality theorem*. And indeed, one can also prove Kőnig's Theorem using LP duality:

Theorem (Kőnig's Theorem)**.** *Let G be a bipartite graph. Then $\nu(G) = \tau(G)$.*

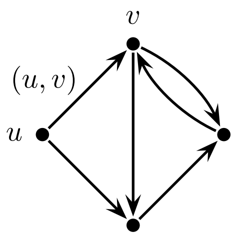
Proof. We write

$$\begin{aligned} \nu(G) &= \max\{\mathbf{1}^T x \mid A_G x \leq \mathbf{1},\ x \geq \mathbf{0}\} \\ &\overset{(*)}{=} \min\{y^T \mathbf{1} \mid y^T A_G \geq \mathbf{1},\ y \in \mathbb{Z}^V_{\geq 0}\} \\ &\overset{(**)}{=} \min\Big\{\sum_{i\in V} y_i \mid y^T A_G \geq \mathbf{1},\ y \in \{0,1\}^V\Big\} \overset{(***)}{=} \tau(G) \end{aligned}$$

In $(*)$ we use that by Lemma 5.10 and Theorem 5.13, both LPs have optimum solutions that are integral. In $(**)$ we use that optimum solution would have $y_i \leq 1$. Finally for $(***)$ we note that the problem of selecting minimum number of rows of A_G to cover $\mathbf{1} \in \mathbb{R}^E$ is exactly the minimum vertex cover problem. □

5.4 TU matrices from directed graphs

We call $D = (V, A)$ a *directed graph* if V is a finite set and A is a subset of ordered pairs from V. We think of $(u, v) \in A$ as the *arc / edge* that goes from node u to node v.



Then the *node-edge incidence matrix of D* is the matrix $A_D \in \{-1,0,1\}^{V \times A}$ with

$$(A_D)_{v,a} = \begin{cases} +1 & \text{if } a = (v,*) \\ -1 & \text{if } a = (*,v) \\ 0 & \text{otherwise} \end{cases} \qquad A_D = \begin{array}{cc} (u,v) & \\ \downarrow & \\ \left[\begin{array}{ccc} & & \\ & +1 & \\ & -1 & \\ & & \end{array}\right] & \begin{array}{c} \\ \leftarrow u \\ \leftarrow v \\ \\ \end{array} \end{array}$$

Then we can prove an analogue of Theorem 5.13 for directed graphs (where a bit surprisingly no condition on the graph is needed):

Theorem 5.15. *For any directed graph $D = (V, A)$, the node edge incidence matrix A_D is totally unimodular.*

The proof is very similar to the proof of Theorem 5.13 and we leave it as an exercise.

5.5 The Theorem of Ghouila-Houri

For a matrix $A \in \mathbb{R}^{m \times n}$ we define the *discrepancy* as

$$\text{disc}(A) := \min_{x \in \{-1,1\}^n} \|Ax\|_\infty.$$

The *hereditary discrepancy* is the maximum discrepancy for any submatrix, i.e.

$$\text{herdisc}(A) := \max_{J \subseteq [n]} \text{disc}(A^J)$$

where A^J is the $m \times |J|$ submatrix of A indexed by J.

Theorem 5.16 (Ghouila-Houri 1962)**.** *Let $A \in \mathbb{Z}^{m \times n}$. Then the following is equivalent:*

(A) A is totally unimodular.

(B) herdisc$(A) \leq 1$.

Proof. $(A) \Rightarrow (B)$. Since a submatrix of a TU matrix is also TU, it suffices to show that for a TU matrix A one has $\text{disc}(A) \leq 1$. That means we need to show there is a point $y \in \{-1,1\}^n$ so that $-\mathbf{1} \leq Ay \leq \mathbf{1}$. This is equivalent to finding $x \in \{0,1\}^n$ so that $-\mathbf{1} \leq A(2x - \mathbf{1}) \leq \mathbf{1}$. For a reason that will become clear later we instead ask for a $x \in \{0,1\}^n$ so that $-p \leq A(2x-\mathbf{1}) \leq p$ where $p \in \{0,1\}^m$ (which clearly is still sufficient). So consider the polytope

$$\begin{aligned} P &:= \{x \in \mathbb{R}^n \mid -p \leq A(2x - \mathbf{1}) \leq p, \mathbf{0} \leq x \leq \mathbf{1}\} \\ &= \left\{x \in \mathbb{R}^n \mid \frac{-p + A\mathbf{1}}{2} \leq Ax \leq \frac{p + A\mathbf{1}}{2},\ \ \mathbf{0} \leq x \leq \mathbf{1}\right\} \end{aligned}$$

From the first representation we know that $\frac{1}{2}\mathbf{1} \in P$ and so $P \neq \emptyset$ (no matter how we choose p). We know that A is TU, so the polytope will have only integral vertices if the right hand side vector is integral. We choose $p \in \{0,1\}^m$ to have the same parity as $A\mathbf{1}$ so that $p + A\mathbf{1}$ and $-p + A\mathbf{1}$ are even integers. This settles the claim.

$(B) \Rightarrow (A)$. Suppose A is a minimal matrix for which the claim is false, i.e. $A \in \mathbb{Z}^{k \times k}$ is non-singular, $\text{herdisc}(A) \leq 1$ and for each proper submatrix B one has $\det(B) \in \{-1,0,1\}$. Fix the unique vector y so that $Ay = \det(A)e_1$. By Cramer's rule, each entry y_i equals the determinant of A with the ith column being replaced by e_i. That determinant is $\pm\det(B)$ for some $(k-1) \times (k-1)$ submatrix B of A. Hence indeed $y \in \{-1,0,1\}^k$.

Next, let $x \in \{-1,0,1\}^k$ so that $supp(x) = supp(y)$ and $\|Ax\|_\infty \leq 1$. Then $x + y \in \{-2,0,2\}^k$ and for $i \in \{2,\ldots,k\}$ we have $|\langle A_i, x+y\rangle| \leq 1$ which using parity in fact means that $\langle A_i, x\rangle = \langle A_i, x+y\rangle = 0$. As A is non-singular this means $\langle A_1, x\rangle \neq 0$ which only leaves $\langle A_i, x\rangle \in \{-1,1\}$, i.e. $Ax \in \{-e_1, e_1\}$. Then $Ay \in \{-\det(A)Ax, \det(A)Ax\}$ and again by non-singularity $y \in \{-\det(A)x, \det(A)x\}$. Since $x, y \in \{-1,0,1\}^n$ this only leaves $|\det(A)| = 1$. $\square$

The proof is (partly) taken from Korte and Vygen [KV12].

5.6 The Integer Decomposition Property

We make the following definition.

Definition 5.17. A polyhedron $P \subseteq \mathbb{R}^n$ has the *integer decomposition property* if for any $k \in \mathbb{Z}_{\geq 1}$ and $x \in kP \cap \mathbb{Z}^n$ there are points $x_1, \ldots, x_k \in P \cap \mathbb{Z}^n$ so that $x = x_1 + \ldots + x_k$.

The literature also uses the term *normal polytope* instead of integer decomposition property. The condition implies that any rational vector $\frac{x}{k} \in P$ (i.e. $k \in \mathbb{Z}_{\geq 1}$ and $x \in \mathbb{Z}^n$) must lie in the convex hull of k integer points in P. In particular that means every polytope (or rational polyhedron) with integer decomposition polytope must be integral. But the reverse is not necessarily true. We provide an example from Francisco Santos:

Example 5.18. Consider the points $X := \{\mathbf{0}, e_1+e_2, e_1+e_3, e_2+e_3\} \subseteq \mathbb{R}^3$ and the polytope $P = \mathrm{conv}(X)$ which by definition is integral. Then $(1,1,1)^T \in 2P$. Since $X \subseteq [0,1]^3$, it is not hard to see that in fact $P \cap \mathbb{Z}^3 = X$. Then one can verify that no two points of X sum up to $(1,1,1)^T$.

But in fact, there are large classes of polyhedra with the integer decomposition property:

Proposition 5.19. *For $A \in \mathbb{Z}^{m\times n}$ totally unimodular and $b \in \mathbb{Z}^m$, the polyhedron $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ has the integer decomposition property.*

Proof. As suggested we prove this by induction over k. Nothing to show for $k = 1$, so we assume $k \geq 2$. Fix $y \in kP \cap \mathbb{Z}^n$. In order to find the first vector x_1 that we can split off, we consider

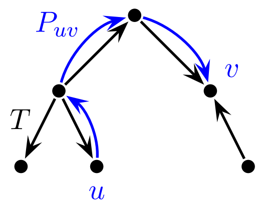
$$\begin{aligned} Q &:= \{x \in \mathbb{R}^n \mid x \in P, y - x \in (k-1)P\} \\ &= \{x \in \mathbb{R}^n \mid Ax \leq b, A(y-x) \leq (k-1)b\} \\ &= \{x \in \mathbb{R}^n \mid Ax \leq b, -Ax \leq -Ay + (k-1)b\} \\ &= \left\{x \in \mathbb{R}^n \mid \underbrace{\begin{pmatrix} A \\ -A \end{pmatrix}}_{=:B} x \leq \underbrace{\begin{pmatrix} b \\ -Ay + (k-1)b \end{pmatrix}}_{=:f} \right\} \end{aligned}$$

Note that because A is TU, also the constraint matrix B of Q is TU. Moreover because b, A, y, k are integer, also the right hand side vector f describing Q is integer. Note that $Q \neq \emptyset$ because $x := \frac{y}{k}$ is in Q (easy to see that $\frac{y}{k} \in P$ and $y - \frac{y}{k} = (k-1)\frac{y}{k} \in (k-1)P$). Hence the polyhedron Q contains an integer point that we denote by $x^1 \in P \cap \mathbb{Z}^n$ — recall that $y - x^1 \in (k-1)P \cap \mathbb{Z}^n$. We apply induction and write $y - x^1 = x^2 + \ldots + x^k$ with $x^2, \ldots, x^k \in P \cap \mathbb{Z}^n$ and are done! $\square$

5.7 Network matrices*

Finally we discuss an important class of TU matrices.

Definition 5.20. Let $T = (V, F)$ be a directed tree and let $D = (V, E)$ be a directed graph on the same set of vertices. For $e = (u, v) \in E$, let P_e denote the unique directed path from u to v using edges (or reverse edges) in T.

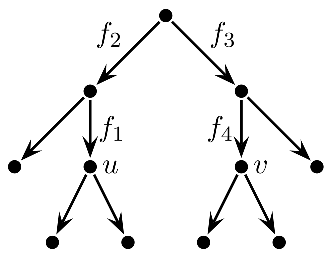


Define a matrix $A \in \mathbb{R}^{F \times E}$ with entries

$$A_{f,e} := \begin{cases} +1 & \text{if } f \text{ appears in forward direction in } P_e \\ -1 & \text{if } f \text{ appears in backward direction in } P_e \\ 0 & \text{if } f \text{ does not appear on } P_e \end{cases} \quad \forall f \in F, e \in E$$

Then A is called a *network matrix.*

We observe that flipping the direction of an edge in F corresponds to flipping the signs in the row A_f. This is an operation that does not change whether A is TU and so if desired we may assume that all arcs in T are directed away from the root.



$$\begin{array}{ccc} & e=(u,v) & \\ A = & \left[\begin{array}{ccc} & \vdots & \\ & -1 & \\ & -1 & \\ & +1 & \\ & +1 & \\ & \vdots & \end{array}\right] & \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ \\ \end{array} \end{array}$$

Here is a useful observation:

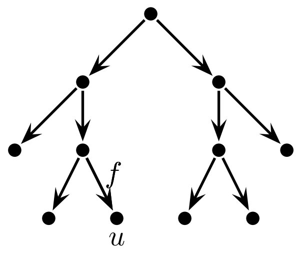
Lemma 5.21. *Any submatrix of a network matrix A is again a network matrix.*

Proof. Deleting a column corresponds to deleting one edge $e \in E$ which trivially results again in a network matrix. On the other hand, delete a row belonging to $f = (u, v)$ corresponds to contracting the vertices u, v in the tree T and treating u and v as the same node in D. $\square$

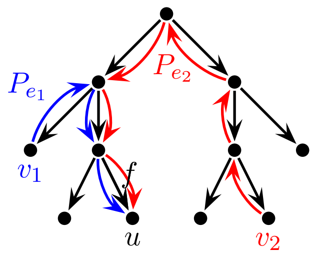
Now we can prove:

Theorem 5.22. *Every network matrix is TU.*

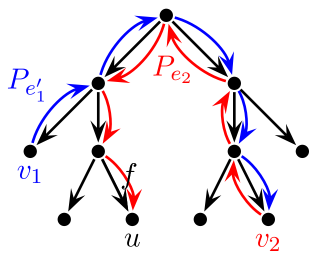
Proof. Let $G = (V, E)$ and $T = (V, F)$. By Lemma 5.21 it suffices to show that each square network matrix has determinant in $\{-1, 0, 1\}$. We prove this by induction over the size of that square matrix where the 1×1 case is trivial. Fix a leaf $u \in V$ in the tree and let $f \in F$ be the incident edge. After swapping directions of edges in D (which affects only the sign of the determinant) we may assume that all paths that contain f, are containing f in forward direction. That means the row A_f contains only entries from $\{0, 1\}$. Now consider two edges $e_1 = (v_1, u), e_2 = (v_2, u) \in E$ so that P_{e_1}, P_{e_2} both contain f. Replacing e_1 with $e_1' := (v_1, v_2)$ corresponds to replacing column A^{e_1} by $A^{e_1} - A^{e_2}$. This operation does not affect the determinant and results in another network matrix where one path less contains the edge f. We repeat this until only one path contains f (if there was no such path to begin with then $\det(A) = 0$ anyway). After this, row A_f contains a single 1; we develop $\det(A)$ using that row and apply induction.



Vertex u and edge $f \in F$



Paths P_{e_1} and P_{e_2}



Paths $P_{e_1'}$ and P_{e_2} $\square$

5.8 Exercises

Exercise 5.1.
Give a min-max relation for the maximum weight of a stable set in a bipartite graph $G = (V, E)$ without isolated vertices.
Comment. What is meant is that you are given a bipartite graph $G = (V, E)$ and a non-negative integer weight function $w : V \to \mathbb{Z}_{\geq 0}$ and you are asked to find an expression of the form $\min\{...\}$ that equals the maximum of $\sum_{i \in S} w_i$ over all stable sets $S \subseteq V$. *Source:* This exercise is modified from Schrijver [Sch17].

Exercise 5.2.
Let $D = (V, A)$ be a directed graph. Prove that the node-edge incidence matrix $A \in \{-1, 0, 1\}^{V \times A}$ of D is totally unimodular.

Exercise 5.3.
A set family $\mathcal{F} \subseteq 2^X$ is called *laminar* if for all $S, T \in \mathcal{F}$ one either has $S \subseteq T$ or $T \subseteq S$ or $S \cap T = \emptyset$. Let $A \in \{0, 1\}^{\mathcal{F} \times X}$ be a matrix whose rows are the characteristic vectors of $\mathcal{F}$. Prove that A is totally unimodular.
Example. The following is one such a matrix:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Chapter 6

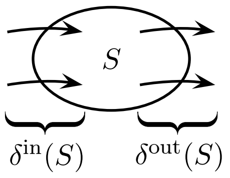
Flows and circulations

Most of the material from this chapter is adapted from Schrijver [Sch17].

6.1 Flows

In this section, we introduce one of the most natural and useful optimization problems. For a directed graph $D = (V, A)$ and a subset $S \subseteq V$ of vertices, we define

$$\delta^{\text{in}}(S) := \{(u, v) \in A \mid u \notin S, v \in S\}$$
$$\delta^{\text{out}}(S) := \{(u, v) \in A \mid u \in S, v \notin S\}$$



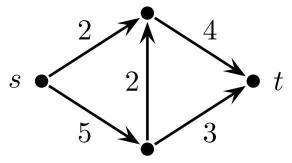
Definition 6.1. Let $D = (V, A)$ be a directed graph with $s, t \in V$. A function $f : A \to \mathbb{R}$ is an *s-t flow* if

- $f(a) \geq 0 \;\; \forall a \in A$
- $\sum_{a \in \delta^{\text{in}}(v)} f(a) = \sum_{a \in \delta^{\text{out}}(v)} f(a) \;\; \forall v \in V \setminus \{s, t\}$

The *value* of a flow is the net amount of flow leaving the source s:

$$\text{value}(f) := \sum_{a \in \delta^{\text{out}}(s)} f(a) - \sum_{a \in \delta^{\text{in}}(s)} f(a)$$

One can verify that the value of a flow is also equal to the net amount of flow entering the sink t.



s-t flow of value 7

Definition 6.2. Given a function $u : A \to \mathbb{R}_{\geq 0}$, a flow f is a *flow under* u if $f(a) \leq u(a)\ \forall a \in A$.

The main problem for us will be:

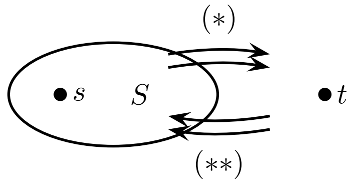
MAX FLOW PROBLEM
Input: $D = (V, A)$, $s, t \in V$ and $u : A \to \mathbb{R}_{\geq 0}$
Goal: Find an s-t flow f under u of maximum value

For $S \subseteq V$ with $s \in S$, $t \notin S$, we call the edge set $\delta^{\text{out}}(S)$ an s-t *cut*. Given capacities $u : A \to \mathbb{R}_{\geq 0}$, we call $u(\delta^{\text{out}}(S)) := \sum_{a \in \delta^{\text{out}}(S)} u(a)$ the *capacity* of a cut. It seems that s-t cuts are a natural "dual" concept to flows. And in fact, any s-t cut gives a simple upper bound on how large any s-t flow can be as any unit of flow needs to use an edge of the s-t cut.

Lemma 6.3. *Let f be an s-t flow under u and let $\delta^{out}(S)$ be an s-t cut. Then value$(f) \leq u(\delta^{out}(S))$.*

Proof. We have

$$\begin{aligned}
\text{value}(f) &= \sum_{a \in \delta^{\text{out}}(s)} f(a) - \sum_{a \in \delta^{\text{in}}(s)} f(a) \\
&= \sum_{v \in S} \Big(\sum_{a \in \delta^{\text{out}}(v)} f(a) - \sum_{a \in \delta^{\text{in}}(v)} f(a) \Big) \\
&= \sum_{a \in \delta^{\text{out}}(S)} \underbrace{f(a)}_{\leq u(a)\ (*)} - \underbrace{\sum_{a \in \delta^{\text{in}}(S)} f(a)}_{\geq 0\ (**)} \leq u(\delta^{\text{out}}(S))
\end{aligned}$$



□

From the proof we obtain the important observation we have equality in Lemma 6.3 iff both $(*)$ and $(**)$ are equalities for all arcs!

Corollary 6.4. *Let f be an s-t flow under u and let $\delta^{out}(S)$ be an s-t cut. Then*

$$\Big(\text{value}(f) = u(\delta^{out}(S))\Big) \iff \begin{pmatrix} f(a) = u(a) & \forall a \in \delta^{out}(S) \\ f(a) = 0 & \forall a \in \delta^{in}(S) \end{pmatrix}$$

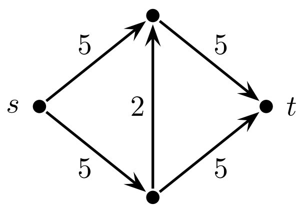
6.2 Finding a maximum flow

Next, we will describe a method to compute a maximum s-t flow in a graph. For an edge $a = (v, w)$ we denote the *inverse edge* as $a^{-1} := (w, v)$.

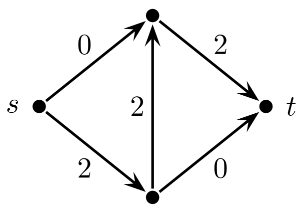
Definition 6.5. Let $D = (V, A)$ be a directed graph with capacities $u : A \to \mathbb{R}_{\geq 0}$ and let $f : A \to \mathbb{R}$ be an s-t flow under u. The *residual graph* $D_f = (V, A_f)$ with *residual capacities* $u_f : A_f \to \mathbb{R}_{>0}$ is defined by

$$\begin{array}{rcll} f(a) < u(a) & \Rightarrow & a \in A_f \text{ and } u_f(a) = u(a) - f(a) & \text{(forward edge)} \\ f(a) > 0 & \Rightarrow & a^{-1} \in A_f \text{ and } u_f(a^{-1}) = f(a) & \text{(backward edge)} \end{array}$$

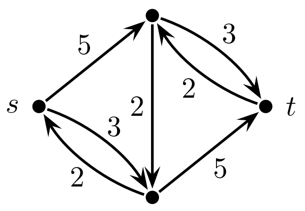
Intuitively, the residual graph denotes how the flow f can be changed without violating capacities or non-negativity.



capacities u



flow f



graph D_f with cap. u_f

There is however an annoying technicality: a directed graph D might potentially contain edges in both directions (v, w) and (w, v). Then the residual graph might contain (v, w) both as forward edge and as backward edge. This is neither a mathematical nor an algorithmic issue, but notationally one would need to either allow multi-edges or keep track which fraction of the residual capacities $u_f(a)$ comes from which of the two cases. In order to avoid this, whenever we discuss a residual graph we will make the simplifying assumption that the graph D does not contain both edges (v, w) and (w, v). If needed one could always introduce a dummy node z and replace (w, v) by $(w, z), (z, v)$.

Next, we discuss the usefulness of the residual graph. Fix a graph $D = (V, A)$ with $s, t \in V$, capacities $u : A \to \mathbb{R}_{\geq 0}$ and an s-t flow f under u. Suppose there is an s-t path P in D_f that consists of edges $a_1, \dots, a_k$. Let $\alpha := \min\{u_f(a_1), \dots, u_f(a_k)\}$ be the minimal residual capacity on that path. Note that $\alpha > 0$ by construction. We define a function $f' : A \to \mathbb{R}_{\geq 0}$ with

$$f'(a) := \begin{cases} f(a) + \alpha & \text{if } a = a_i \text{ for some } i \\ f(a) - \alpha & \text{if } a = a_i^{-1} \text{ for some } i \\ f(a) & \text{otherwise} \end{cases}$$

In other words, we increase the flow on all forward arcs in P by α and we decrease the flow on backward arcs by α. By a simple case split we can see the following (we leave the proof to the reader).

Lemma 6.6. *f' is an s-t flow under u with value(f') = value$(f) + \alpha$.*

An important early result on flows is that the value of a maximum s-t flow equals the value of a minimum s-t cut.

Theorem 6.7 (MaxFlow=MinCut; Ford Fulkerson [FF56]). *For any directed graph $D = (V, A)$ and $u : A \to \mathbb{R}_{\geq 0}$, one has*

$$\max\{\textit{value}(f) \mid f \text{ is } s\text{-}t \text{ flow under } u\} = \min\{u(\delta^{out}(S)) \mid \{s\} \subseteq S \subseteq V \setminus \{t\}\}$$

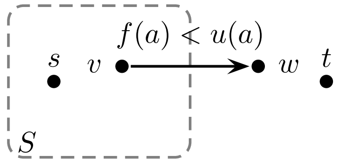
Proof. We fix an s-t flow f of maximum value (which has to exist by compactness). Consider the residual graph D_f. If there is any s-t path P in D_f, then we can augument f along P to some other flow f' with value(f') > value(f) which would be a contradiction to optimality of f. Hence we may assume that there is no s-t path in D_f.

Define $S := \{v \in V \mid \exists \text{path in } D_f \text{ from } s \text{ to } v\}$. By construction $s \in S$ and $t \notin S$. Hence $\delta^{\text{out}}(S)$ is an s-t cut. It remains to prove that this particular s-t cut S satisfies the claim.

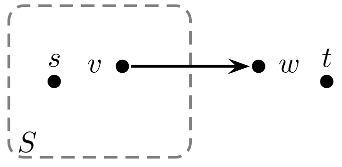
Claim. *One has value$(f) = u(\delta^{out}(S))$.*

Proof of Claim. Due to the observation from Cor 6.4 it suffices to verify that $(*)$ and $(**)$ are tight:

- *An outgoing edge $a = (v, w) \in \delta^{out}(S)$ has $f(a) = u(a)$.* Otherwise $f(a) < u(a)$, $a \in A_f$ and w would be reachable via v.

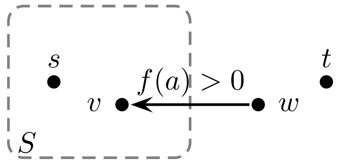


graph D

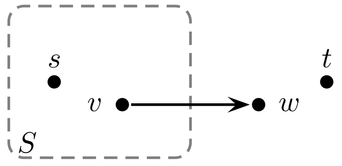


graph D_f

- *An incoming edge $a = (w, v) \in \delta^{in}(S)$ has $f(a) = 0$.* If $f(a) > 0$, then $a^{-1} \in A_f$ and again w was reachable via v.



graph D



graph D_f

□

The MaxFlow=MinCut Theorem is a form of a duality theorem similar to König's Theorem (Theorem 4.9) or Theorem 3.26. In fact, it can be proven alternatively using LP duality. We only want to sketch the arguments here. Consider a directed graph

$D = (V, A)$ and let $M \in \{-1, 0, 1\}^{V \times A}$ be the node edge incidence matrix. For a subset $U \subseteq V$, we write M_U as the submatrix containing only the rows indexed by U. Then one can indeed rewrite the max flow problem as a linear program and then take the dual:

$$\begin{aligned}
& \max\{\text{value}(f) \mid f \text{ flow under } u\} & (6.1)\\
= \quad & \max\{M_s x \mid \mathbf{0} \leq x \leq u, M_{V\setminus\{s,t\}} x = \mathbf{0}, x \in \mathbb{R}^A\} & (6.2)\\
\overset{\text{LP duality}}{=} \quad & \min\{y^T u \mid y^T + z^T M_{V\setminus\{s,t\}} \geq M_s^T,\ y \in \mathbb{R}^A_{\geq 0},\ z \in \mathbb{R}^{V\setminus\{s,t\}}\} & (6.3)\\
= \quad & \text{...(some work)...} & (6.4)\\
= \quad & \min\{u(\delta^{\text{out}}(S)) \mid \{s\} \subseteq S \subseteq V \setminus \{t\}\} & (6.5)
\end{aligned}$$

The constraint matrix of the LP in (6.2) is indeed totally unimodular. Not very surprisingly, this implies that if all capacities $u(a)$ are integral, then there is always an integral maximum s-t flow. But at this point it is more crucial that the constraint matrix of the dual is also TU and hence the LP in (6.3) will have an integral optimum solution (y, z). Then from that one can reason that the solution indeed corresponds to an s-t cut[1].

The proof of the MaxFlow=MinCut theorem suggests a natural algorithm in order to actually find a maximum s-t flow.

Ford and Fulkerson's algorithm for Max Flows

Input: $D = (V, A)$, $s, t \in V$, $u : A \to \mathbb{R}_{\geq 0}$.
Output: A maximum s-t-flow under u

```
(1) Set $f(a) = 0$ for all $a \in A$.
(2) REPEAT
      (3) Find any $s$-$t$ path $P = (a_1, \dots, a_k)$ in $D_f$. If none exists then stop.
      (4) Set $\alpha := \min\{u_f(a_i) : i = 1, \dots, k\}$ as the minimum residual capacity on that path.
      (5) Augment $f$ along $P$ by $\alpha$.
```

We prove correctness of the algorithm.

Theorem 6.8. *Let $D = (V, A)$ and $u : A \to \mathbb{Q}_{\geq 0}$. Then Ford Fulkerson finds a maximum flow in finite time.*

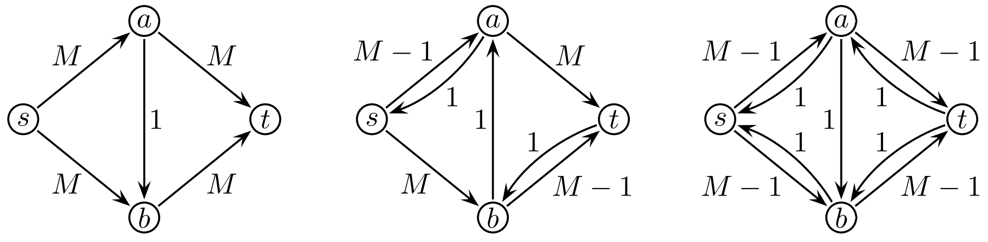
Proof. By assumption, there is some $K \in \mathbb{N}$ so that $K \cdot u(a) \in \mathbb{Z}$ for all $a \in A$. Then in each iteration, value(f) increases by at least $\frac{1}{K}$. Hence the algorithm will terminate in finite time and from the proof of Theorem 6.7 we know that f must then be optimal. □

[1]Note that one also needs to use that $u(a) \geq 0$ since otherwise the argument would apply for the **NP**-hard Max Cut problem.

Curiously, if the capacities $u(a) \in \mathbb{R}_{\geq 0}$ are arbitrary real numbers, then the Ford-Fulkerson algorithm might never terminate — in fact, it might not even converge to an optimum solution.

Instead of discussing that instance, we show a simple example where progress in each iteration might be very minuscule:

Example 6.9. Consider Ford-Fulkerson on this instance



D with cap. c residual graph D_f after 1 iter. residual graph D_f after iter. 2

Then if we alternatingly choose augmenting paths $s \to a \to b \to t$ and $s \to b \to a \to t$, then the flow value will only increase by 1 in each iteration and hence the Ford Fulkerson algorithm takes $2M$ iterations to terminate.

Note that if we modify the Ford Fulkerson algorithm and always select the currently *shortest* s-t path in the residual graph (in terms of number of edges), then $|V| \cdot |A|$ iterations suffice. This is a result independently by Dinitz [Din70] and Edmonds and Karp [EK72].

6.3 Application: Elimination of sports teams

Maximum flows have a huge number of practical applications as many problems can be formulated as a flow problem. Of course that implies that the problem can be solved with any maximum flow algorithm. But additionally it means that stuctural results carry over from flows to the specific problems. We want to demonstrate this with a non-obvious application to the elimination of sports teams.

We consider a *sport league* where a team gets 1 point for a win, 0 points for loosing (no ties). We are in the middle of the season and wonder whether our beloved team can still become *champion* (which equals to having the maximum number of points at the end of the season, possibly in a tie) or whether they are "mathematically eliminated".

Example 6.10 (Example 1)**.** Suppose the current state of the league is as follows

Team	Wins	To play
A	33	8
B	28	4
⋮		

Can team B still become champion? The answer is clearly *no!* as A will always dominate B.

Let us consider a more complex example.

Example 6.11. Suppose instead the current state of the league is:

Team	wins	To play	A	B	C	D
A	33	8	-	1	6	1
B	29	4	1	-	0	3
C	28	7	6	0	-	1
D	27	5	1	3	1	-

Can team B still become champion? The answer is again *no!* First of all, B would need to win all remaining games. Then A would have to loose all games. But then C wins at least another 6 games, obtaining $34 > 33$ points.

In the following, we will prove that in case our team B cannot become champion, there is always a simple criterion that certifies it.

First, we can make the assumption that (as we are discussion the most optimistic case), our team B wins all remaining games! We define

$$\begin{aligned} M &:= \#\text{wins for } B \text{ at end of season} \\ & \text{ assuming they win all remaining games} \\ T &:= \{\text{teams other than } B\} \\ w_i &:= \#\text{wins of team } i \text{ currently} \quad \forall i \in T \\ r_{ij} &:= \#\text{games remaining between } i \text{ and } j \quad \forall i,j \in T, i \neq j \\ P &:= \{\{i,j\} \subseteq T \mid i \neq j, r_{ij} > 0\} \end{aligned}$$

Then the goal is to find an outcome of the games so that all teams $i \in T$ get $\leq M - w_i$ additional wins! We prove the following equivalence:

Theorem 6.12. *The following is equivalent*

(I) Team B cannot become champion.

(II) There is a subset $T' \subseteq T$ with

$$\sum_{i \in T'} w_i + \sum_{\{i,j\} \subseteq T', \{i,j\} \in P} r_{ij} > M \cdot |T'|$$

Before we prove this theorem, we want to demonstrate how it applies to the two previously considered examples. In case of

Team	wins	To play
A	33	8
B	28	4
$\vdots$		

the elimination criterion applies with $T' := \{A\}$! In the second example of

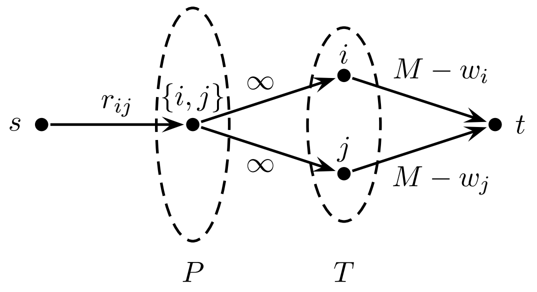
Team	wins	To play	A	B	C	D
A	33	8	-	1	6	1
B	29	4	1	-	0	3
C	28	7	6	0	-	1
D	27	5	1	3	1	-

the elimination criterion applies with $T' := \{A, C\}$! Now we come to the proof.

Proof of Theorem 6.12. ***(II)* ⇒ *(I)*.** We can rearrange the criterion to

$$\begin{array}{c}\text{average \#points}\\ \text{of teams in } T'\\ \text{at end of season}\end{array} \geq \frac{1}{|T'|}\Big(\sum_{i\in T'} w_i + \sum_{\{i,j\}\subseteq T', \{i,j\}\in P} r_{ij}\Big) > M$$

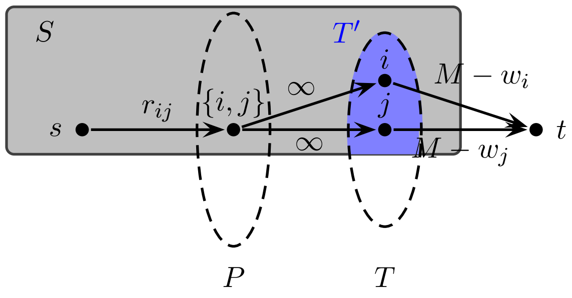
Hence, no matter the outcome, some team in T' will get more than M points and hence beat B.
***(I)* ⇒ *(II)*.** We create a graph $D = (V, A)$ with vertices $V := T \cup P \cup \{s, t\}$, edges $(s, \{i, j\})$ for $\{i, j\} \in P$ with capacity r_{ij}; edges $(\{i, j\}, i)$ for all $\{i, j\} \in P$ with capacity ∞ and edges (i, t) with capacity $M - w_i$ for all $i \in T$.



It is not hard to see that

$$\begin{aligned} B \text{ can become champion} \quad &\Leftrightarrow \quad \exists \text{ integral flow of value } \textstyle\sum_{\{i,j\}\in P} r_{ij} \\ &\Leftrightarrow \quad \exists \text{ flow of value } \textstyle\sum_{\{i,j\}\in P} r_{ij} \\ &\Leftrightarrow \quad \text{every } s\text{-}t \text{ cut } S \text{ has value } \geq \textstyle\sum_{\{i,j\}\in P} r_{ij} \end{aligned}$$

using the MaxFlow=MinCut Theorem. As we are assuming that B cannot become champion, we can fix an s-t cut S of value $u(\delta^{\text{out}}(S)) < \sum_{\{i,j\}\in P} r_{ij}$. We set $T' := T \cap S$ as all the teams in that cut.



We prove the structural claim that the cut S contains exactly the pairs $\{i,j\} \in P$ where $i,j \in T'$:

Claim. *One has* $S = \{s\} \cup \{\{i,j\} \in P : |\{i,j\} \cap T'| = 2\} \cup T'$.

Proof of Claim. If $\{i,j\} \in S$ with $|\{i,j\} \cap T'| < 2$, then an edge with capacity ∞ is cut. On the other hand, if $\{i,j\} \notin S$ while $|\{i,j\} \cap T'| = 2$, then moving $\{i,j\}$ into S decreases the cut value by $r_{ij} > 0$. □

Now we can write the value of the cut as

$$u(\delta^{\text{out}}(S)) = \sum_{\{i,j\} \in P : |\{i,j\} \cap T'| \leq 1} r_{ij} + \sum_{i \in T'} (M - w_i) < \sum_{\{i,j\} \in P} r_{ij}$$

which can be rearranged to

$$M \cdot |T'| < \sum_{i \in T'} w_i + \sum_{\{i,j\} \in P : |\{i,j\} \cap T'| = 2} r_{ij}$$

□

6.4 Circulations

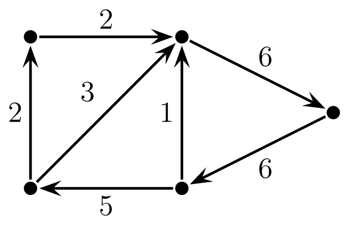
We make the following definition:

Definition 6.13. Let $D = (V, A)$ be a directed graph. A function[2] $f : A \to \mathbb{R}_{\geq 0}$ is a *circulation* if

$$\sum_{a \in \delta^{\text{out}}(v)} f(a) = \sum_{a \in \delta^{\text{in}}(v)} f(a) \quad \forall v \in V$$

Intuitively, a circulation is a flow without distinguished source and sink which makes some arguments cleaner.

[2]Note that we do ask for non-negativity but one can find definitions for circulations elsewhere in the literature that allow negative values $f(a)$.



graph D with circulation

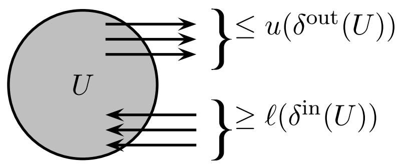
Suppose we are given a graph D with *lower bounds* $\ell(a)$ and *upper bound* $u(a)$ on each edge $a \in A$ and the question is whether there is a circulation f so that $\ell(a) \leq f(a) \leq u(a)$ for each edge $a \in A$. One can prove that if there is no such circulation, then there has to be a set $U \subseteq V$ of vertices where the lower bound on the incoming flow is higher than the upper bound on the outgoing flow.

Theorem 6.14 (Hoffman's Circulation Theorem)**.** *Let $D = (V, A)$ be a directed graph and let $\ell, u : A \to \mathbb{R}$ with $0 \leq \ell(a) \leq u(a)$ for all $a \in A$. The following is equivalent:*

(A) There exists a circulation $f : A \to \mathbb{R}$ with $\ell(a) \leq f(a) \leq u(a)$ for all $a \in A$.

(B) One has $\ell(\delta^{in}(U)) \leq u(\delta^{out}(U))$ for every $U \subseteq V$.

One may visualize (B) as


U
$\leq u(\delta^{\text{out}}(U))$
$\geq \ell(\delta^{\text{in}}(U))$

Proof. ***(A)* ⇒ *(B)***. Let f be a circulation satisfying (A). Then for any set $U \subseteq V$ we have

$$\ell(\delta^{\text{in}}(U)) \leq f(\delta^{\text{in}}(U)) \overset{\text{circulation}}{=} f(\delta^{\text{out}}(U)) \leq u(\delta^{\text{out}}(U))$$

which shows (B).

(B)* ⇒ *(A). We may assume (B) is satisfied and we need to show that a circulation f exists that satisfies (A). First, for any function $f : A \to \mathbb{R}$ (not necessarily a circulation) we define a vector $\text{loss}_f \in \mathbb{R}^V$ with

$$\text{loss}_f(v) := f(\delta^{\text{out}}(v)) - f(\delta^{\text{in}}(v)).$$

We choose a function $f : A \to \mathbb{R}$ with $\ell \leq f \leq u$ that minimizes $\|\text{loss}_f\|_1$. Such a choice much exist by compactness. Intuitively, this means we pick a function that is closest to being a circulation satisfying the lower and upper bounds. If $\text{loss}_f = \mathbf{0}$, then f is indeed a circulation and we are done. So suppose $\text{loss}_f \neq \mathbf{0}$. Set

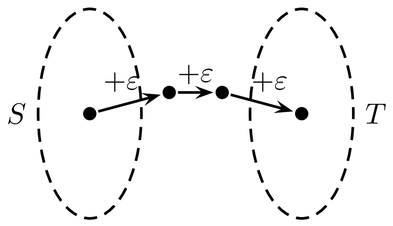
$$S := \{v \in V \mid \text{loss}_f(v) < 0\} \quad \text{and} \quad T := \{v \in V \mid \text{loss}_f(v) > 0\}$$

Slightly extending an earlier definition, we define the *residual graph* as the graph $D_f = (V, A_f)$ with edges

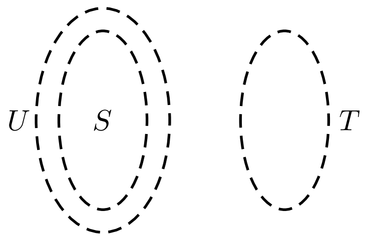
$$\begin{aligned} f(a) < u(a) &\quad\Rightarrow\quad a \in A_f \\ f(a) > \ell(a) &\quad\Rightarrow\quad a^{-1} \in A_f \end{aligned}$$

We distinguish two cases:

- *Case 1: There is a S-T path in D_f.* Fix any S-T path P in D_f We augment the function f along P by some by $\varepsilon > 0$. Then $\|\text{loss}_f\|_1$ decreases and we have a contradiction.



- *Case 2: There is no S-T path in D_f.* Let $U := \{v \in V \mid v \text{ reachable from } S \text{ in } D_f\}$. We verify that for each $a \in \delta^{\text{out}}(U)$, one has $f(a) = u(a)$. Moreover, for each $a \in \delta^{\text{in}}(U)$, one has $f(a) = \ell(a)$ (same argument as in the MaxFlow=MinCut proof).



  Then

  $$0 \;\; > \;\; \text{loss}_f(S) = \text{loss}_f(U) = f(\delta^{\text{out}}(U)) - f(\delta^{\text{in}}(U)) = u(\delta^{\text{out}}(U)) - \ell(\delta^{\text{in}}(U))$$

  Rearranging gives $u(\delta^{\text{out}}(U)) < \ell(\delta^{\text{in}}(U))$.

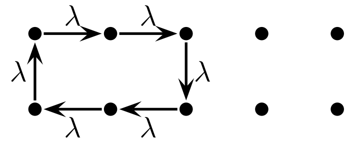
$\square$

We make another definition:

Definition 6.15. We call a circulation f *atomic* if there is a single directed circuit $C \subseteq A$ so that

$$f(a) = \begin{cases} \lambda & \text{if } a \in C \\ 0 & \text{otherwise} \end{cases}$$

for some $\lambda \geq 0$.



It seems intuitive that any circulation is somehow build from atomic circulations. In the literature this fact is usually called *flow decomposition*.

Lemma 6.16. *Let $f : A \to \mathbb{R}_{\geq 0}$ be a circulation in $D = (V, A)$. Then there are atomic circulations $f_1, \dots, f_k : A \to \mathbb{R}_{\geq 0}$ with $f = f_1 + \ldots + f_k$ and $k \leq |A|$.*

Proof. First, it is not hard to prove that for any non-zero circulation f, there is a circuit $C \subseteq \text{supp}(f)$ (just do a walk following any arc a with $f(a) > 0$ until some node is revisited; this closes a cycle; then decompose the cycle into circuits). We let $\lambda_1 := \min\{f(a) : a \in C\}$. Then $f_1 := \lambda_1 \mathbf{1}_C$ is the first atomic circulation and we iterate the argument with $f - f_1$. Note that in each iteration the flow one at least one arc becomes zero, hence this argument will terminate after at most $|A|$ steps. $\square$

There is also a polyhedral argument for Lemma 6.16. Fix a directed graph D and consider the polyhedral convex cone

$$K := \{x \in \mathbb{R}^A \mid A_D x = \mathbf{0}, x \geq \mathbf{0}\}$$

of all circulations in D. By a conic variant of Caratheodory's Theorem (Theorem 3.17), for any $f \in K$, we can write $f = \sum_{i=1}^{k} f_i$ where each $f_i \in K$ is an *extreme ray* of the cone K and $k \leq |A|$. In fact, the extreme rays of K are exactly the atomic circulations.

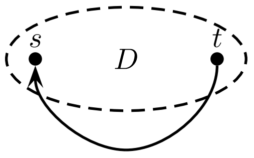
6.5 Circulations of minimum cost

Next, we want to generalize the problem and include edge cost.

Min Cost Circulation Problem
Input: $D = (V, A)$, $\ell, u : A \to \mathbb{R}_{\geq 0}$ and $c : A \to \mathbb{R}$
Goal: Find a circulation $f : A \to \mathbb{R}_{\geq 0}$ with $\ell(a) \leq f(a) \leq u(a)$ for all $a \in A$ that minimizes the cost $c(f) := \sum_{a \in A} c(a) \cdot f(a)$.

We want to make a few comments on this problem:

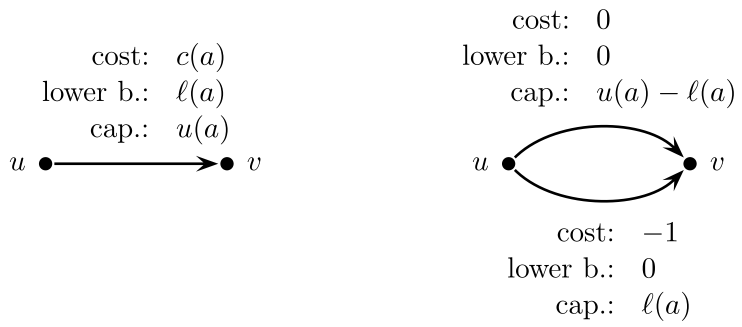
- Min Cost Circulation is indeed a generalization of the maximum s-t flow problem. To see this, take an instance $D = (V, A)$, $s, t \in V$, $u : A \to \mathbb{R}_{\geq 0}$. Add an arc (t, s) with capacity ∞ and cost -1 and call the new graph D'. Then a minimum cost circulation in D' corresponds to a maximum s-t flow in the original graph D.



$u(\cdot) := \infty, \ \ c(\cdot) := -1$

Other problems that can be modelled as Min Cost Circulation are for example shortest s-t path and minimum cost max s-t flow.

- Even finding a *feasible* solution (rather than an optimum one) for a min cost circulation instance (D, ℓ, u, c) does not seem to be trivial. But consider the auxiliary instance (D', ℓ', u', c') where we replace an arc a by two parallel arcs with the following cost / lower bound / capacity:



instance D instance D'

Then the 0-flow is feasible for the auxiliary instance while a minimum cost circulation in D' corresponds to a feasible circulation in D. Hence, algorithmically it suffices to solve min cost circulation with the assumption to have a feasible starting solution[3].

- If A_D is the node edge incidence matrix of the graph D, then the min cost circulation corresponds to an optimum solution to the LP

$$\min\{c^T f \mid A_D f = \mathbf{0}, \ell \leq f \leq u\}$$

The constraint matrix of this LP is totally unimodular by Theorem 5.15. In particular, this implies that if ℓ and u are integer, then there is always an integral optimum min cost circulation.

We extend the definition of residual graphs to include costs.

Definition 6.17. Let $D = (V, A)$ be a graph with bounds $\ell, u : A \to \mathbb{R}_{\geq 0}$ and cost $c : A \to \mathbb{R}$. Let $f : A \to \mathbb{R}$ be a circulation with $\ell \leq f \leq u$. Then the *residual graph*

[3]This argument is similar to "phase 1" for the simplex algorithm.

is the graph $D_f = (V, A_f)$ with residual capacities $u_f : A_f \to \mathbb{R}_{>0}$ and residual cost $c_f : A_f \to \mathbb{R}$ defined by

$$\begin{array}{lcl} f(a) < u(a) & \Rightarrow & a \in A_f, u_f(a) := u(a) - f(a), c_f(a) := c(a) \\ f(a) > \ell(a) & \Rightarrow & a^{-1} \in A_f, u_f(a^{-1}) := f(a) - \ell(a), c_f(a^{-1}) := -c(a) \end{array}$$

Again we can augment along a cycle of the residual graph:

Fact 6.18. *Let f be a circulation in D with $\ell \leq f \leq u$ and let $C \subseteq A_f$ be any directed circuit in the residual graph D_f. For $0 < \alpha \leq \min\{c_f(a) : c \in C\}$, the function $f' : A \to \mathbb{R}$ with*

$$f'(a) := \begin{cases} f(a) + \alpha & \text{if } a \in C \\ f(a) - \alpha & \text{if } a^{-1} \in C \\ f(a) & \text{otherwise} \end{cases}$$

is again a circulation in D with $\ell \leq f' \leq u$ and $c(f') = c(f) + \alpha \cdot c(C)$.

This fact implies that if there is a negative cost cycle in the residual graph, we can augment along the cycle and decrease the cost. This suggests a rather natural algorithm that iteratively finds negative cost cycles in the residual graph and aguments along them. In order to prove correctness, we make the following useful observation:

Fact 6.19. *Let f and f^* be two circulations in D with $\ell \leq f \leq u$ and $\ell \leq f^* \leq u$. Then the "difference circulation" $g : A_f \to \mathbb{R}_{\geq 0}$ with*

$$\begin{array}{lcl} f(a) < f^*(a) & \Rightarrow & g(a) := f^*(a) - f(a) \\ f(a) > f^*(a) & \Rightarrow & g(a^{-1}) := f(a) - f^*(a) \end{array}$$

(and 0 elsewhere) is a circulation in D_f with $0 \leq g(a) \leq u_f(a)$ for all $a \in A_f$ and cost $c_f(g) = c(f^*) - c(f)$.

Proposition 6.20. *Let $D = (V, A)$ be a directed graph and let $\ell, u, c : A \to \mathbb{R}$. A circulation f with $\ell \leq f \leq u$ is optimal $\Leftrightarrow$ there is no negative cost cycle in D_f.*

Proof. The direction ($\Rightarrow$) follows from Fact 6.18. For ($\Leftarrow$), let f be an arbitrary circulation in D and assume that D_f does not contain negative cost cycles. Let f^* be a minimum cost circulation. Let $g : A_f \to \mathbb{R}_{\geq 0}$ be the difference between f^* and f, i.e. $c_f(g) = c(f^*) - c(f)$ (see Fact 6.19). Then by Lemma 6.16 we can write $g = \sum_{i=1}^k g_i$ where $g_1, \ldots, g_k$ are atomic circulations in the residual graph D_f. Then

$$c(f^*) - c(f) = c_f(g) = \sum_{i=1}^k \underbrace{c_f(g_i)}_{\geq 0} \geq 0$$

Then $c(f) \leq c(f^*)$ and hence f is optimal. □

```
Min cost circulation algorithm
Input: $D = (V, A)$, $\ell, u : A \to \mathbb{R}_{\geq 0}$, $c : A \to \mathbb{R}$.
Output: A min cost circulation $f$ with $\ell \leq f \leq u$
  (1) Find a feasible circulation $f_0$ with $\ell(a) \leq f_0(a) \leq u(a)$ for all $a \in A$
      using remark on page 69.
  (2) FOR $t = 0$ TO $\infty$ DO
      (3) Construct the residual graph $D_{f_t}$.
      (4) Find a cycle $C \subseteq D_{f_t}$ with $c_{f_t}(C) < 0$. If there is non STOP.
      (5) Set $\alpha := \min\{u_{f_t}(a) : a \in C\}$
      (6) Augment $f_t$ along $C$ by $\alpha$ and call the outcome $f_{t+1}$
```

We can conclude the following:

Corollary 6.21. *The sequence of circulations satisfies $c(f_0) < c(f_1) < \ldots$. Moreover, if the algorithm terminates, the penultimate circulation is optimal.*

Proof. Follows from Prop 6.20. □

It is indeed possible to select a cycle in (4) so that the algorithm runs in polynomial time. This is the goal of the next section.

6.6 The minimum mean cycle canceling algorithm*

In this section, we discuss a result due to Tardos [Tar85] where in the exposition we will follow Chapter 12 in Schrijver [Sch03]. In our min cost circulation algorithm, we want to make a more careful choice of which cycle in D_f to use for augmentation. It will turn out that the right cycle is the one that minimizes the *average cost*. Let us define

$$\mu(f) := \min_{C \text{ cycle in } D_f} \left\{ \frac{c_f(C)}{|C|} \right\}$$

as the cost of the *minimum mean cycle*. Note that we allow the empty cycle so that always $\mu(f) \leq 0$.

Lemma 6.22. *Given a directed graph $D = (V, A)$ with edge cost $c : A \to \mathbb{R}$. Then a minimum mean length cycle can be found in time $O(|V|^2 \cdot |A|)$.*

Proof. Consider the following slight variation of the Bellman-Ford algorithm: set

$$d_0(u, v) := \begin{cases} 0 & \text{if } u = v \\ \infty & \text{if } u \neq v \end{cases} \qquad \text{and} \qquad d_k(u, v) := \min_{(w,v) \in A} \{d_{k-1}(u, w) + c(w, v)\}$$

for $k = 1, \dots, |V|$ and $u, v \in V$. Then $d_k(u, v) = c(P)$ where P is the u-v walk with exactly k arcs that minimizes $c(P)$. Computing all entries takes time $O(|V|^2 \cdot |A|)$. Then the cost of the minimum mean cycle is

$$\min_{k=0,\dots,|V|;\ u \in V} \left\{ \frac{d_k(u,u)}{k} \right\} \qquad (*)$$

(counting the entries for $k = 0$ as having value 0) and also the cycle itself can be recovered from the entries. The crucial argument why $(*)$ is attained by a cycle and not (only) by a walk is the following: suppose that C is a closed walk attaining $(*)$. Then that walk can be decomposed into cycles $C_1, \dots, C_\ell$ and in particular $c(C) = \sum_{i=1}^{\ell} c(C_\ell)$. Then one can verify that $\frac{c(C)}{|C|} \leq \min_{i=1,\dots,\ell} \frac{c(C_i)}{|C_i|}$, hence one of the cycles C_i was at least as good as C. ☐

Note that the minimum mean cost cycle can be found with a more sophisticated argument in time $O(|V| \cdot |A|)$, see for example Chapter 7.3 in the textbook of Korte and Vygen [KV12].

We restate the algorithm that we will analyze for convinience:

Minimum mean cycle canceling algorithm

Input: $D = (V, A)$, $\ell, u : A \to \mathbb{R}_{\geq 0}$, $c : A \to \mathbb{R}$.
Output: A min cost circulation f with $\ell \leq f \leq u$

```
(1) Compute any feasible circulation $f$ with $\ell(a) \leq f(a) \leq u(a)$ for all $a \in A$.
(2) WHILE $\exists$ cycle in $D_f$ with negative cost DO
        (3) Compute minimum mean cycle $C \subseteq A_f$ w.r.t. cost $c_f$
        (4) Augment $f$ along $C$ by $\min\{u_f(a) : a \in C\}$
```

6.6.1 Node potentials

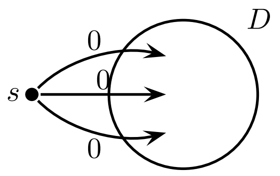
For a moment we will discuss general directed graphs and cost functions, not in the context of circulations. For a directed graph $D = (V, A)$ with edge cost $c : A \to \mathbb{R}$, a function $\pi : V \to \mathbb{R}$ are called *node potentials*. These induce *reduced costs* $c^{\pi}_{i,j} := c_{ij} + \pi_i - \pi_j$.

Lemma 6.23. *Let $D = (V, A)$ be a directed graph with cost $c : A \to \mathbb{R}$ and node potentials π. Then any cycle $C \subseteq A$ has $c^{\pi}(C) = c(C)$.*

Proof. Clear as the node potentials cancel out. ☐

Lemma 6.24. *Let $D = (V, A)$ be a directed graph with arc cost $c : A \to \mathbb{R}$. Then D has no negative cost cycle $\Leftrightarrow$ there are node potentials with $c^{\pi}_{i,j} := c_{ij} + \pi_i - \pi_j \geq 0 \quad \forall (i,j) \in A$.*

Proof. "$\Leftarrow$". Any cycle C has $c(C) = c^\pi(C) \geq 0$ by Lemma 6.23.
"$\Rightarrow$" We add an artificial node s and arcs (s,v) with $c(s,v) := 0$ for all $v \in V$.



Let $d(u,v)$ be the length of the shortest path u-v in the resulting graph. Note that $d(u,v)$ maybe negative, but since there are no negative cost cycle, d is well defined. Define $\pi(i) := d(s,i)$ as the node potentials. Then $c^\pi_{i,j} = c_{ij} + d(s,i) - d(s,j) \geq 0 \Leftrightarrow d(s,j) \leq d(s,i) + c_{ij}$ which is the triangle inequality. $\square$

Now, let us go back to the case that we have a directed graph $D = (V,A)$ with bounds $\ell \leq u$ and a circulation $f : A \to \mathbb{R}_{\geq 0}$. For a node potential $\pi : V \to \mathbb{R}$ we consider the reduced costs $c^\pi_f(i,j) := c_f(i,j) + \pi_i - \pi_j$. However the considered circulations will keep changing which changes the arcs A_f in the residual graph, but it will not change the cost $c_f(i,j)$. Hence in order to keep the notation clean, we will drop the index f in the cost and only use $c^\pi(i,j)$ — but we will always consider the reduced costs with respect to the residual graph!

A circulation $f : A \to \mathbb{R}_{\geq 0}$ is *ε-optimal* for $\varepsilon \geq 0$ if there are node potentials $\pi : V \to \mathbb{R}$ so that $c^\pi_{ij} \geq -\varepsilon$ for all $(i,j) \in A_f$. Let

$$\varepsilon(f) := \min\{\varepsilon \geq 0 : f \text{ is } \varepsilon\text{-optimal}\}$$

The interpretation of this quantity is that $\varepsilon(f)$ is the smallest amount that has to be added to the cost of the arcs in the residual graph to eliminate all negative cost cycles. In some sense $\varepsilon(f)$ gives a measure of distance to optimality.

Lemma 6.25. *For any circulation f, $\mu(f) = -\varepsilon(f)$.*

Proof. "$\geq$" Let π be the node potentials valid for $\varepsilon(f)$. The minimum mean cycle C in D_f has $|C| \cdot \mu(f) = c(C) = c^\pi(C) \geq -\varepsilon(f) \cdot |C|$.
"$\leq$". Let use define a cost function $\tilde{c}(u,v) := c(u,v) - \mu(f)$. Now there is no negative cost cycle w.r.t. $\tilde{c}$ and there are node potentials π with $\tilde{c}(i,j) + \pi_i - \pi_j \geq 0$, which is the same as $c(i,j) + \pi_i - \pi_j \geq \mu(f)$. $\square$

6.6.2 The weakly polynomial bound

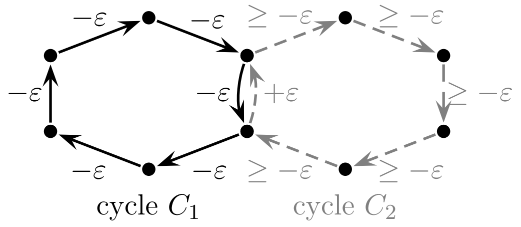
The analysis of the cycle cancelling method will consist of showing that the value of $\varepsilon(f)$ is decreasing in the course of the algorithm.

Lemma 6.26. *Update f to f' by augmenting along a minimum mean cost cycle. Then $\varepsilon(f') \leq \varepsilon(f)$.*

Proof. Abbreviate $\varepsilon := \varepsilon(f)$. Let $\pi : V \to \mathbb{R}$ be the node potentials with $c^\pi(a) \geq -\varepsilon$ for every arc $a \in A_f$. We will show that the *same* node potentials are still feasible for the updated graph $D_{f'}$.

Let $C \subseteq A_f$ be the minimum cost mean cycle. Note that its cost are $c^\pi(C) = -|C| \cdot \varepsilon$ by Lemma 6.25. That means that *every* arc $a \in C$ must have $c^\pi(a) = -\varepsilon$. The only new arcs $(i,j) \in A_{f'} \setminus A_f$ have $(j,i) \in C$. Hence the reduced cost are $c^\pi_{ij} = -c_{ji} + \pi_i - \pi_j = -c^\pi_{ji} = \varepsilon \geq 0$. $\square$

We should remark that the claim of Lemma 6.26 is false if an arbitrary cycle is chosen for updates. To understand the final part of the analysis better, fix a current circulation f with $\varepsilon := -\mu(f)$ and let π be the corresponding potential. Suppose for the sake of argument that the next two mean cycles are C_1 and C_2 contains the reverse of the arc that was the bottleneck for C_1. Then the picture for the reduced cost in the residual graph will be as follows:



But then the mean cost of C_2 has to be higher as it contains an arc with reduced cost $+\varepsilon$. This idea can be generalized as follows:

Lemma 6.27. *Consider a sequence $\{f_i\}_{i\geq 0}$ of circulations where f_{i+1} emerges from f_i by augmenting along the minimum mean cycle in D_{f_i}. Then $\varepsilon(f_{|A|+1}) \leq (1 - \frac{1}{|V|}) \cdot \varepsilon(f_0)$.*

Proof. Let $\varepsilon := \varepsilon(f_0)$ and fix the potentials $\pi : V \to \mathbb{R}$ with $c^\pi(a) \geq -\varepsilon$ for all $a \in A_{f_0}$.
Claim I. *There is a $k \in \{0, \ldots, |A| + 1\}$ so that the minimum mean cost cycle $C \subseteq D_{f_k}$ in that iteration contains an arc $a \in C$ with $c^\pi(a) \geq 0$.*
Proof of Claim I. As long as the current iteration k uses a minimum mean cost cycle $C \subseteq D_{f_k}$ with $c^\pi(a) < 0$ for all $a \in C$, every arc that appears new in the residual graph will have non-negative reduced cost. Moreover, in every iteration at least one arc is bottleneck arc and will not appear in the residual graph of the next iteration. This can only go on for at most $|A|$ iterations.
Claim II. *Consider the minimal such k from Claim I. Then $\mu(f_k) \geq -(1 - \frac{1}{|V|}) \cdot \varepsilon$.*
Proof of Claim II. Let $C \subseteq A_{f_k}$ be the minimum mean cycle. We know that $c^\pi(a) \geq -\varepsilon \ \forall a \in A_{f_k}$. Then $c(C) = c^\pi(C) \geq (|C| - 1) \cdot (-\varepsilon)$ and hence $\mu(f_k) = \frac{c(C)}{|C|} \geq (1 - \frac{1}{|C|}) \cdot (-\varepsilon)$. The claim follows from $|C| \leq |V|$. $\square$

Theorem 6.28. *Suppose the cost function is $c : A \to \{-c_{\max}, \dots, +c_{\max}\}$. Then the minimum mean cycle cancelling algorithm terminates after $|V| \cdot (|A|+1) \cdot \ln(2|V| \cdot c_{\max})$ iterations.*

Proof. Let f be the flow before the 1st iteration and f' is the circulation after $|V| \cdot (|A| + 1) \cdot \ln(2|V| \cdot c_{\max})$ iterations. Then f is $c_{\max}$-optimal. Moreover,

$$\varepsilon(f') \leq c_{\max} \cdot \Big(1 - \frac{1}{|V|}\Big)^{|V| \cdot \ln(2|V| \cdot c_{\max})} \leq c_{\max} \cdot \exp(-\ln(2|V|c_{\max})) = \frac{1}{2|V|}.$$

Since the cost are integral, that implies that actually $\varepsilon(f') = 0$. □

6.6.3 The strongly polynomial bound

Surprisingly one can prove that the number of iterations is bounded by a polynomial only in $|V|$ and $|A|$, independent from cost and capacity values.

Theorem 6.29 (Tardos [Tar85])**.** *The minimum mean cost cycle algorithm terminates after $O(|A|^2|V|\ln(|V|))$ iterations.*

Proof. We set $n := |V|$ and $m := |A|$. Let $f_0, f_1, \dots$ be the *circulations* appearing in the minimum mean cycle algorithm and let C_t be the *minimum mean cycle* in D_{f_t} used to augment f_t to f_{t+1}. We abbreviate $\tau := 2nm\lceil\ln(n)\rceil$. The crucial claim is the following:

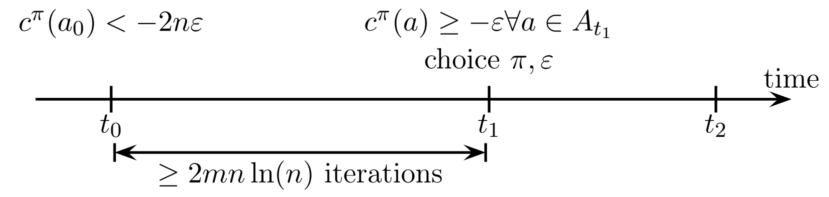
Claim I. *For each t_0, there exists at least one arc $a \in C_{t_0}$ so that $f_{t_1}(a_0) = f_{t_2}(a_0)$ for all $t_2 \geq t_1 := t_0 + \tau$.*

Intuitively, the claim says that every τ iterations, the flow on some arc will be fixed and never change again. If we manage to prove Claim I, then indeed the algorithm will terminate after at most $2m\tau = O(m^2 n \ln(n))$ iterations.

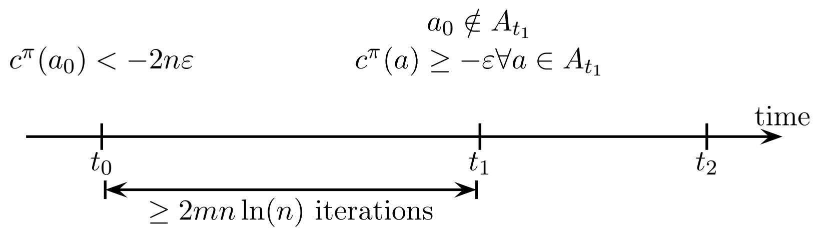
Proof of Claim I. We set $\varepsilon := \varepsilon(t_1)$. Let π be *potential* for iteration t_1, i.e. $c^\pi(a) := c(a) + \pi(u) - \pi(v)$ and $c^\pi(a) \geq -\varepsilon$ for every $a = (u, v) \in A_{t_1}$. Then using Lemma 6.27 we obtain

$$\varepsilon \leq \varepsilon(t_0) \cdot \Big(1 - \frac{1}{n}\Big)^{\tau/m} \leq \varepsilon(t_0)\exp(-2\lceil\ln(n)\rceil) \leq \frac{\varepsilon(t_0)}{2n}$$

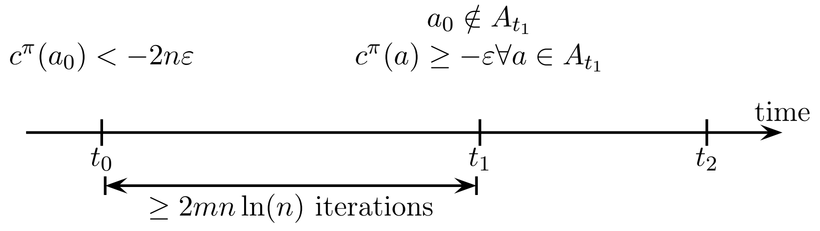
Fix any arc $a_0 \in C_{t_0}$ with $c^\pi(a_0) \leq -\varepsilon(f_{t_0}) < -2n\varepsilon$. It may be helpful to consider the time axis:


$c^\pi(a_0) < -2n\varepsilon$
$c^\pi(a) \geq -\varepsilon \forall a \in A_{t_1}$
choice π, ε
time
t_0
t_1
t_2
$\geq 2mn\ln(n)$ iterations

Assume by symmetry that $a_0 \in A$ *(=forward arc)* and assume for sake of contradition that $f_{t_2}(a_0) \neq f_{t_1}(a_0)$. We will analyze what happens to this particular edge.
Subclaim I.A. *One has $a_0 \notin A_{t_1}$*
Proof of Subclaim I.A. If $a_0 \in A_{t_1}$, then $c^\pi(a_0) \geq -\varepsilon$ while at the same time $c^\pi(a_0) < -2n\varepsilon$. That is a contradiction! □


$a_0 \notin A_{t_1}$
$c^\pi(a_0) < -2n\varepsilon$
$c^\pi(a) \geq -\varepsilon \forall a \in A_{t_1}$
time
t_0
t_1
t_2
$\geq 2mn\ln(n)$ iterations

Subclaim I.B. *There exists a directed circuit C so that (i) $C \subseteq A_{t_2}$, (ii) $C^{-1} \subseteq A_{t_1}$, (iii) $a_0 \in C$.*
Proof of Subclaim I.B. By the previous claim we have $u(a_0) = f_{t_1}(a_0) > f_{t_2}(a_0)$. Then the difference $h := f_{t_1} - f_{t_2}$ (see Fact 6.19) is a circulation in D_{t_2} with $h(a_0) > 0$. Choose a circuit $C \subseteq \{a \mid h(a) > 0\}$ containing a_0. Then $C \subseteq A_{t_2}$ and $C^{-1} \subseteq A_{t_1}$. □


$a_0 \notin A_{t_1}$
$c^\pi(a_0) < -2n\varepsilon$
$c^\pi(a) \geq -\varepsilon \forall a \in A_{t_1}$
time
t_0
t_1
t_2
$\geq 2mn\ln(n)$ iterations

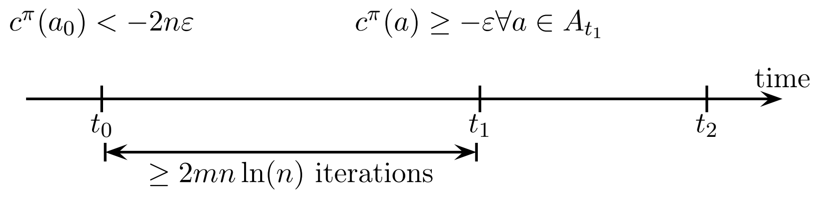
Now we continue the proof of Claim I. We have

$$\left(C^{-1} \subseteq A_{t_1} \text{ and } c^\pi(a^{-1}) \geq -\varepsilon\right) \quad \Longrightarrow \quad c^\pi(a) \leq \varepsilon \ \forall a \in C$$

Then

$$c(C) = c^\pi(C) = \underbrace{c^\pi(a_0)}_{<-2n\varepsilon} + \underbrace{c^\pi(C \setminus \{a_0\})}_{\leq(|C|-1)\cdot\varepsilon} < -n\varepsilon \leq -|C| \cdot \varepsilon(t_2)$$

So there is a circuit C in D_{t_2} with mean cost $< -\varepsilon(t_2)$. That is a contradiction.

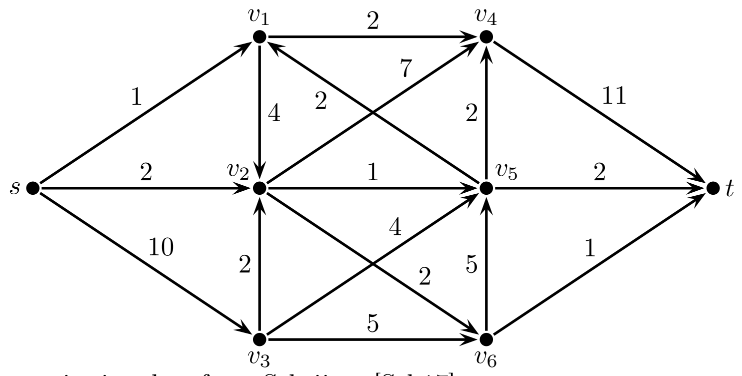

$c^\pi(a_0) < -2n\varepsilon$
$c^\pi(a) \geq -\varepsilon \forall a \in A_{t_1}$
time
t_0
t_1
t_2
$\geq 2mn\ln(n)$ iterations

□

6.7 Exercises

Exercise 6.1.
Let $D = (V, A)$ be a directed graph and let $s, t \in V$. Let $f : A \to \mathbb{R}_{\geq 0}$ be an s-t flow of value β. Show that there exists an s-t flow $f' : A \to \mathbb{Z}_{\geq 0}$ of value $\lceil \beta \rceil$ so that $\lfloor f(a) \rfloor \leq f'(a) \leq \lceil f(a) \rceil$ for every $a \in A$.
Source: This exercise is taken from Schrijver [Sch17].

Exercise 6.2.
In the following graph $D = (V, A)$ (edges labelled with capacities $u(a)$), compute a maximum s-t flow under u and a minimum s-t cut $\delta^{out}(U)$. What are their values? It suffices to state the final outcomes.



Source: This exercise is taken from Schrijver [Sch17].

Exercise 6.3.
Let $D = (V, A)$ be a directed graph with two distinguished nodes $s, t \in V$. A set U is called an *s-t vertex cut* if $U \subseteq V \setminus \{s, t\}$ and every s-t path intersects U. A collection of s-t paths $P_1, \ldots, P_N$ is called *internally vertex disjoint* if they have no nodes in common other than s and t. Prove the following using the MaxFlow=MinCut Theorem: *Let $D = (V, A)$ be a directed graph with $s, t \in V$ so that $(s, t) \notin A$. Then the maximum number of internally vertex-disjoint s-t paths equals the minimum $|U|$ where U is an s-t vertex cut.*
Hint: Create an auxiliary graph and apply the MaxFlow=MinCut Theorem there!

Exercise 6.4.
First answer the following:

(i) Let $D' = (V, A')$ be a directed graph with capacities u' and $s, t \in V$. We call an s-t flow $g : A' \to \mathbb{R}_{\geq 0}$ *elementary* if there is a single s-t path P in D' so that

$$g(a) = \begin{cases} \text{value}(g) & \text{if } a \in P \\ 0 & \text{if } a \notin P \end{cases}$$

Now let f^* be a maximum s-t flow in D' under u'. Prove that there is an elementary s-t flow g under u' with $\text{value}(g) \geq \frac{1}{|A'|}\text{value}(f^*)$.

Now let $D = (V, A)$ be a directed graph with integral capacities $u : A \to \mathbb{Z}_{\geq 0}$, distinguished vertices $s, t \in V$ and for the sake of simplicity suppose that for each arc $a \in A$ one has $a^{-1} \notin A$.

(ii) Let f be any flow under u and let f^* be a maximum value s-t flow. Prove that the residual graph D_f contains an s-t path P where $u_f(a) \geq \frac{1}{2|A|}(\text{value}(f^*) - \text{value}(f))$ for all $a \in P$.

(iii) Consider the modification of the Ford-Fulkerson algorithm where in each iteration we pick an s-t path P that maximizes $\min\{u_f(a) : a \in P\}$ where f is the current flow. Prove that this algorithm takes at most $O(|A| \ln(2\text{value}(f^*)))$ many iterations.

Exercise 6.5.
Let $D = (V, A)$ be a directed graph with capacities $u(a) := 1$ for all $a \in A$. Let $s, t \in V$ and assume that $\delta^{\text{out}}(t) = \emptyset$. Let $\delta^{\text{in}}(t) = \{a_1, \dots, a_m\}$ be the arcs incoming to t. Define $M = (X, \mathcal{I})$ with $X := \{a_1, \dots, a_m\}$ and $\mathcal{I} := \{\{a_i \in X : f(a_i) = 1\} \mid f \text{ is an } s\text{-}t \text{ flow under } u \text{ in } D\}$. Prove that M is a matroid!

Chapter 7

Non-bipartite matching

In Chapter 4, we discussed the matching problem in bipartite graphs. As it turns out matchings still have a nice structural properties in general graphs, but the extensions of theorems from bipartite graphs can be highly non-trivial. We follow Chapter 5 in [Sch17].

7.1 The Tutte-Berge Formula

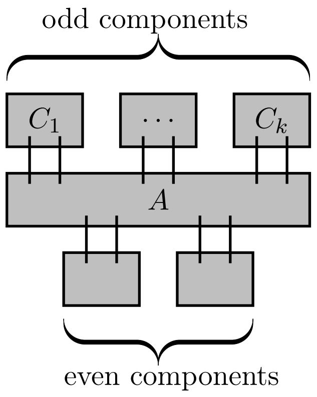
We want to begin with an extension of König's Theorem (Theorem 4.9). Suppose we have an arbitrary undirected graph $G = (V, E)$. How could we certify that there is no perfect matching? A trivial reason would be if $|V|$ is odd. Surprisingly this idea can be extended nicely.

Recall that $\nu(G)$ denotes the maximum cardinality of a matching in G. Let us define

$$\text{ex}(G) := \min\{\#M\text{-exposed nodes} \mid M \text{ matching in } G\} = |V| - 2\nu(G)$$

as the minimum number of exposed vertices. Let us call a connected component in a graph *odd* if it has an odd number of vertices. We define $\text{odd}(G)$ as the number of odd components in graph G.

Consider a matching M in a graph G and fix some subset $A \subseteq V$. Suppose that $C_1, \dots, C_k$ with $k := \text{odd}(G \setminus A)$ are the odd components in $G \setminus A$. Then for each $i = 1, \dots, k$, the matching M either leaves a node in C_i exposed, or it contains an edge between a node in C_i and A.



Either way, M must have at least $k - |A|$ many exposed nodes, that means

$$\text{ex}(G) \geq \text{odd}(G \setminus A) - |A| \quad \Leftrightarrow \quad \nu(G) \leq \min\left\{\frac{1}{2}\Big(|V| + |A| - \text{odd}(G \setminus A)\Big) \mid A \subseteq V\right\}$$

Quite surprisingly it turns out that there is always a set A that provides a tight bound.

Theorem 7.1 (Tutte-Berge Formula)**.** *For every graph $G = (V, E)$ one has*

$$\nu(G) = \min_{A \subseteq V} \left\{\frac{1}{2}\big(|V| + |A| - odd(G \setminus A)\big)\right\}$$

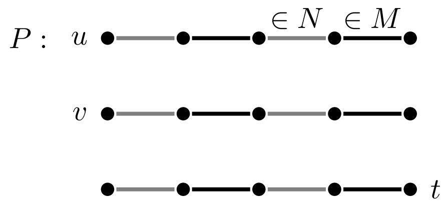
Before we can prove the Tutte-Berge formula, we need a crucial structural lemma. Let us call a node v *critical* if it is covered by every maximum matching. In the proof of Theorem 4.9 we have shown that every non-empty bipartite graph contains a critical node. This is not true for non-bipartite graphs, but instead we can prove the following for graphs without critical nodes:

Lemma 7.2. *Let $G = (V, E)$ be a connected graph with $E \neq \emptyset$ and no critical node. Then $\text{ex}(G) = 1$.*

Proof. If $\text{ex}(G) = 0$, then every node is critical. Hence suppose for the sake of contradiction that $\text{ex}(G) \geq 2$, which means that every maximal matching M leaves at least two nodes exposed. For two nodes $u, v \in V$, let $d(u, v)$ be the distance in G (in terms of the number of edges; note that here we use connectedness). Fix a maximal matching that minimizes $d(u, v)$ for a pair of M-exposed nodes. If $d(u, v) = 1$, then $M \cup \{u, v\}$ is a bigger matching and we have a contradiction. Hence suppose that $d(u, v) \geq 2$. Select any node $t \in V$ with $d(u, t), d(v, t) < d(u, v)$ (for example by taking a node t on the shortest path between u and v).

Consider a maximal matching N that leaves t exposed. If there are several, choose the matching that maximizes the number $|M \cap N|$ of joint edges.

Next, consider the symmetric difference $M \Delta N$. Each of the nodes u, v, t is exposed in either M or N, so they are all endpoints of some paths in $M \Delta N$. Since M and N are maximal and we maximized $|M \cap N|$, we know that $M \Delta N$ consists only of even length paths. Consider the even length path P containing u as an endpoint. If the other endpoint is t, then $M \Delta P$ has the exposed nodes v and t which contradicts the choice of M. That means P has u as one endpoint and the other one is neither v nor t. In fact, the situation looks like this:



Then the matching $N \Delta P$ still has t exposed but has more edges in common with M, which is a contradiction. □

Proof of the Tutte-Berge Formula. We prove the equivalent statement:

$$\text{ex}(G) = \max\{\text{odd}(G \setminus A) - |A| \mid A \subseteq V\}$$

We already argued the direction "$\geq$". We prove "$\leq$" by induction.

- *Case G is disconnected:* Let $G_1, \ldots, G_k$ are the connected components of G (doesn't matter whether these are even or odd) with $k \geq 2$. We apply induction to each of the connected components to find $A_i \subseteq V(G_i)$ with $\text{ex}(G_i) = \text{odd}(G_i \setminus A_i) - |A_i|$. Then $A := \bigcup_{i=1}^k A_i$ gives

$$\text{ex}(G) = \sum_{i=1}^k \text{ex}(G_i) = \underbrace{\sum_{i=1}^k \text{odd}(G_i \setminus A_i)}_{=\text{odd}(G \setminus A)} - \underbrace{\sum_{i=1}^k |A_i|}_{=|A|}$$

- *Case: G connected and there is no critical node.* Using the last Lemma we know that G is odd and $\text{ex}(G) = 1$. Hence for $A := \emptyset$ we have $\text{ex}(G) = 1 = \text{odd}(G \setminus A) - |A|$.

- *Case: G connected and there is a critical node $u \in V$.* We apply induction to $G \setminus \{u\}$ and obtain a set $A \subseteq V \setminus \{u\}$ with

$$\begin{aligned} \text{ex}(G) &\overset{u \text{ critical}}{=} \text{ex}(G \setminus \{u\}) - 1 \\ &\overset{\text{induction}}{=} \text{odd}((G \setminus \{u\}) \setminus A) - |A| - 1 \\ &= \text{odd}(G \setminus (A \cup \{u\})) - |A \cup \{u\}| \end{aligned}$$

Here we use that deleting a critical node increases the minimum number of exposed nodes by 1. That means $A \cup \{u\}$ satisfies the claim.

□

A consequence is the following:

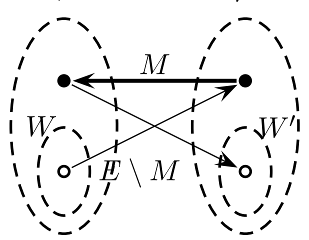
Corollary 7.3 (Tutte's 1-factor theorem [Tut50])**.** *A graph $G = (V, E)$ has a perfect matching if and only if $odd(G \setminus A) \leq |A|$ for all $A \subseteq V$.*

7.2 Cardinality matching algorithm

In this section, we want to design a polynomial time algorithm for the *cardinality matching problem*: given a graph $G = (V, E)$, find a matching $M \subseteq E$ maximizing $|M|$. We know from Theorem 4.8 that it suffices to find augmenting paths for any matching M. In the special case of bipartite graphs, this is reasonably straightforward. Unfortunately, M-augmenting paths are harder to find in general graphs. Recall that a *walk* in a graph $G = (V, E)$ is a sequence $v_0, v_1, \ldots, v_t$ so that $\{v_i, v_{i+1}\} \in E$ for all $i = 0, \ldots, t-1$. In particular in a walk one is allowed to revisit nodes and edges (which is in contrast to a path). We say that a walk $(v_0, v_1, \ldots, v_t)$ is *M-alternating* if for each node v_i with $i \in \{1, \ldots, t-1\}$ exactly one of the edges $\{v_{i-1}, v_i\}, \{v_i, v_{i+1}\}$ lies in M and the other one does not. As the name suggests, edges on the walk are alternatingly in M and not in M. In particular an M-augmenting path is also an M-alternating walk. For a subset $W \subseteq V$, we call a walk a *W-W walk* if start and endpoint are in W.

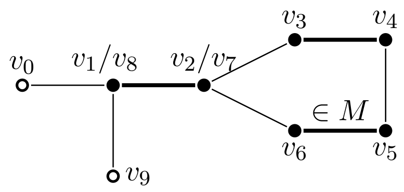
Lemma 7.4. *Let $G = (V, E)$ be a graph and $M \subseteq E$ be a matching leaving $W \subseteq V$ exposed. Then a shortest M-alternating W-W walk P between two exposed vertices can be found in time $O(|V| + |E|)$.*

Proof. We define a directed graph $D = (V \cup V', A)$ that has nodes v and v' for every original node in G. We insert an edge $(u, v') \in A$ if $\{u, v\} \in E \setminus M$. We insert $(u', v) \in A$ if $\{u, v\} \in M$. For every node $u \in W$, we can find the shortest path to some node in W' in polynomial time using breadth first search.

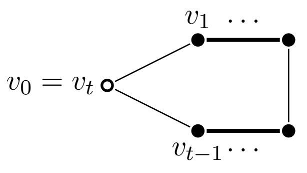


□

Note that every M-alternating W-W walk will have odd length. Unfortunately, an M-alternating path between M-exposed nodes is not necessarily an M-augmenting path:



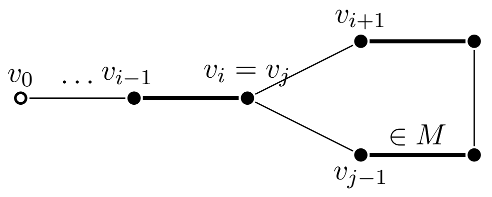
While such a walk does not yield an M-augmenting path, it does provide us with a different useful structure: An *M-blossom* is an M-alternating walk $P = (v_0, v_1, \ldots, v_t)$ where $v_0 = v_t$ is M-exposed and $v_0, \ldots, v_{t-1}$ are distinct.



If the algorithm behind Lemma 7.4 does not provide us with a path, then we can extract a blossom as follows: flip the edges on the walk until the first revisited node to get another matching M'; the first closed cycle will then be an M'-blossom.

Lemma 7.5. *Let $G = (V, E)$ and let M be a matching leaving $W \subseteq V$ exposed. Suppose $P = (v_0, \ldots, v_t)$ is a shortest M-alternating W-W walk and P is not a path. Pick j minimal with $v_i = v_j$ for some $0 \leq i < j \leq t$. Let $M' := M \Delta \{\{v_0, v_1\}, \ldots, \{v_{i-1}, v_i\}\}$, $C := (v_i, \ldots, v_j)$. Then $|M'| = |M|$ is a matching and C is an M'-blossom.*

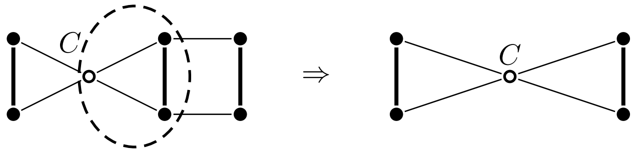
Proof. If $j - i$ was even, we could shorten P. So $j - i$ is odd. As v_j is the first revisited node, all edges $\{v_0, v_1\}, \ldots, \{v_{j-1}, v_j\}$ are distinct. As $j - i$ is odd, either both or none of $\{v_i, v_{i+1}\}, \{v_{j-1}, v_j\}$ are in M. Then the latter has to be the case. Hence $\{v_{i-1}, v_i\} \in M$ has to be the matching edge incident to $v_i = v_j$. Then i is even and the claim follows.



$\square$

Let $G = (V, E)$ and $C \subseteq V$. Then $G/C := (V/C, E/C)$ is the graph that is obtained by *shrinking* the nodes in C into one supernode:

$$\begin{aligned} V/C &:= V \setminus C \cup \{C\} \\ E/C &:= \{e \in E \mid e \cap C = \emptyset\} \cup \{\{u, C\} \mid \{u, v\} \in E \text{ with } |\{u, v\} \cap C| = 1\} \end{aligned}$$



contraction of an M-blossom C

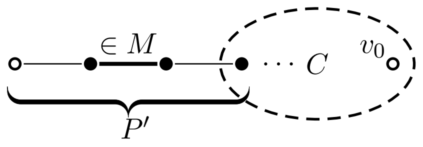
We have already seen how to identify an M-blossom. Now we can show why an M-blossom is anyhow useful:

Lemma 7.6. *Let C be an M-blossom in G. Then M has maximum size in G iff M/C has maximum size in G/C.*

Proof. Let $C = (v_0, \ldots, v_t)$ be the M-flower. Note that v_0 is M-exposed in G and C is M/C-exposed in G/C. We prove the equivalent statement

$$\exists M\text{-augmenting path in } G \quad \Leftrightarrow \quad \exists M/C\text{-augmenting path in } G/C$$

"$\Rightarrow$" Let P be an M-augmenting path. We may assume that P does not start at v_0 (otherwise reverse it) and does touch C (otherwise P is itself M/C-augmenting). Let P' be the beginning of P ending with the first node in C. The edge of P' entering C is not in M (by def. of M-blossom). Then P' is M/C-augmenting.



"$\Leftarrow$" Let P be an M/C-augmenting path in G/C. If node C is not contained in P, then P is M-augmenting. Otherwise, C will be one of the endpoints of P. In G, P is a path ending in some node v_i, $i \in \{0, \ldots, t\}$. We can extend P by going either clockwise or counterclockwise through the blossom and ending in v_0. This gives an M-augmenting path ending in v_0. □

Now we have all the pieces together for an algorithm:

Edmonds' algorithm

Input: $G = (V, E)$, matching $M \subseteq E$.
Output: A matching N with $|N| = |M| + 1$ or conclusion that M is maximal.

```
(1) Set $W := M$-exposed vertices.
(2) Compute shortest $W$-$W$ $M$-alternating walk $P = (v_0, \dots, v_t)$
(3) Case 1. There is no such walk
    (4) Return "$M$ is maximal"
(5) Case 2. There is such a walk
    (6) Case 2a. $P$ is a path
        (7) Then $P$ is an $M$-augmenting path.
        (8) Return $N := M \Delta E(P)$
    (9) Case 2b. $P$ is not a path
        (10) Let $v_j$ be first revisited node with $v_i = v_j$ for $i < j$, $i$ even, $j$ odd
        (11) Set $M' := M \Delta \{\{v_0, v_1\}, \dots, \{v_{i-1}, v_i\}\}$ which is a matching with $|M'| = |M|$
        (12) Set $C := \{v_i, v_{i+1}, \dots, v_j\}$ which is an $M'$-blossom
        (13) Call algorithm recursively for $G/C$ and $M'/C$
        (14) IF $M'/C$ is maximal, then
            (15) $M'$ is maximal in $G \rightarrow$ return $M$ is maximal
        (16) IF $\exists M'/C$-augmenting path $P$, then
            (17) obtain $M'$-augmenting path $P'$
            (18) return $M' \Delta E(P')$
```

Correctness follows from the preceeding lemmas. We analyze the running time:

Theorem 7.7. *A maximum cardinality matching in $G = (V, E)$ can be found in time $O(|V|^2|E|)$.*

Proof. Assume $|E| \geq |V|/2$, otherwise delete vertices without edges. We compute a maximum matching starting with $M := \emptyset$, augmenting it always by 1 using the above algorithm. A shortest M-alternating W-W walk in (2) can be found in time $O(|E|)$. The recursion has depth at most $|V|$, hence augmenting M by 1 using the algorithm above takes time at most $O(|V| \cdot |E|)$. At most $|V|/2$ augmentations are needed, hence the total time to compute a maximum matching is $O(|V|^2 \cdot |E|)$. □

With (a lot!) more work one can get the running time down to $O(\sqrt{|V|} \cdot |E|)$, see the work of Micali and Vazirani.

7.3 The Perfect Matching Polytope*

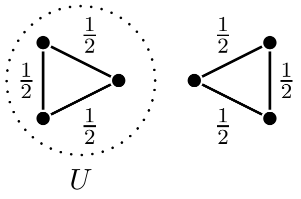
Similar to Sec 4.4, for an undirected graph G, the *perfect matching polytope* is

$$P_{\text{perfectmatching}}(G) := \text{conv}\{\chi^M \in \mathbb{R}^E \mid M \subseteq E \text{ is perfect matching}\}$$

We have learned that for a bipartite graph, $P_{\text{perfectmatching}}(G)$ is equal to the vectors $x \in \mathbb{R}^E$ satisfying

$$x(\delta(v)) = 1 \ \ \forall v \in V; \quad x_e \geq 0 \ \ \forall e \in E \tag{7.1}$$

We also know that in general graphs the inequalities in (7.1) do not suffice to describe $P_{\text{perfectmatching}}(G)$. For example, suppose G is a complete graph on an even number of vertices and we define a vector $x^* \in \mathbb{R}^E$ with $x^*_e = \frac{1}{2}$ for all edges on two disjoint odd cycles.



Then x^* satisfies (7.1) but $x^* \notin P_{\text{perfectmatching}}(G)$. One way to certify this is to let U be the vertices of one of the odd cycles. Then $|U|$ is odd and any perfect matching M must have $|\delta(U) \cap M| \geq 1$. In other words, $x(\delta(U)) \geq 1$ is a feasible linear inequality for $P_{\text{perfectmatching}}(G)$ that is violated by x^*. It is a seminal result due to Edmonds, that these additional inequalities suffice to define the perfect matching polytope of any graph. So, we abbreviate

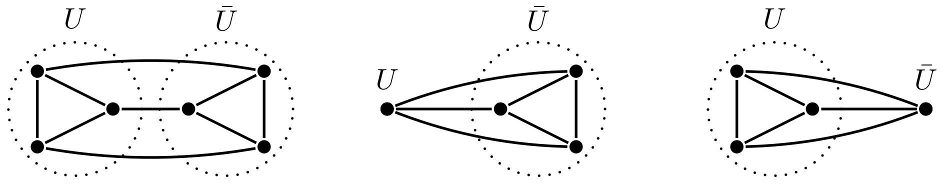
$$Q_{\text{perfectmatching}}(G) := \left\{ x \in \mathbb{R}^E \mid \begin{array}{rcl} x(\delta(v)) & = & 1 \ \ \forall v \in V \\ x_e & \geq & 0 \ \ \forall e \in E \\ x(\delta(U)) & \geq & 1 \ \ \forall U \subseteq V \text{ with } |U| \text{ odd} \end{array} \right\}$$

Theorem 7.8 (Edmonds [Edm65]). *For any graph G one has $P_{\text{perfectmatching}}(G) = Q_{\text{perfectmatching}}(G)$.*

We follow the short proof from [Sch03].

Proof. We already argued that $P_{\text{perfectmatching}}(G) \subseteq Q_{\text{perfectmatching}}(G)$. Suppose the reverse direction $Q_{\text{perfectmatching}}(G) \subseteq P_{\text{perfectmatching}}(G)$ is false and consider the counterexample $G = (V, E)$ that minimizes $|V| + |E|$. Let x^* be a vertex of $Q_{\text{perfectmatching}}(G)$ with $x^* \notin P_{\text{perfectmatching}}(G)$. If $x^*_e = 0$ then we can delete the edge e, and if $x^*_e = 1$ then we can delete the edge plus the two endpoints of e in order to obtain a smaller counter example. So we may assume that $0 < x^*_e < 1$ for all $e \in E$. In particular $d_G(v) \geq 2$ for each vertex $v \in V$ and $|E| \geq |V|$. We know that $|V|$ is even since otherwise the odd set constraint for $U := V$ was violated. If $|E| = |V|$, then all vertices have degree exactly 2 and G consists of a collection of circuits that must all be even as overwise $x(\delta(U)) \geq 1$ would again be violated where U are the vertices of one of the odd circuits. Then we are in the bipartite case and indeed $x^* \in P_{\text{perfectmatching}}(G)$. So, suppose that $|E| > |V|$ instead. Since x^* is a vertex of $Q_{\text{perfectmatching}}(G)$ we know that there is some odd set U with $3 \leq |U| < |V|$ and $x^*(\delta(U)) = 1$.

Consider the contracted graphs $G' := G/U$ and $G'' := G/\bar{U}$ where $\bar{U} = V \setminus U$. Let x' be the weight function x in G' after contraction and let x'' be the weight function x in G'' after contraction. Note that $x' \in Q_{\text{perfectmatching}}(G')$ and $x'' \in Q_{\text{perfectmatching}}(G'')$. For this we use in particular that $|U|$, $|\bar{U}|$ and a singleton vertex are all odd sets.



graph G graph G' graph G''

As G is assumed to be a minimal counterexample, we know that $x' \in P_{\text{perfectmatching}}(G')$ and $x'' \in P_{\text{perfectmatching}}(G'')$. Since each extreme point must be rational, there must be a $k \in \mathbb{N}$ so that x' and x'' are unweighted averages of k perfect matchings in G' and G'', i.e. $x' = \frac{1}{k}\sum_{i=1}^k \chi^{M'_i}$ and $x'' = \frac{1}{k}\sum_{i=1}^k \chi^{M''_i}$. Each of the matchings M'_i and M''_i has exactly one edge going between U and $\bar{U}$ and moreover, each edge e that runs between U and $\bar{U}$ is in exactly kx^*_e many of the matchings $M'_1, \ldots, M'_k$ and also in the same number of matchings $M''_1, \ldots, M''_k$. Hence we can change the indices so that for each i, $M'_i \cap M''_i$ contains exactly one edge and that edge is in $\delta(U)$. That means $M_i := M'_i \cup M''_i$ is a perfect matching in G and one can verify that $x^* = \frac{1}{k}\sum_{i=1}^k \chi^{M_i}$. □

7.4 Vizing's Theorem*

For an undirected graph $G = (V, E)$, we write $d_G(v)$ as the *degree* of a node v and $\Delta(G) := \max_{v \in V} d_G(v)$ is the *maximum degree*. A *k-edge coloring* is a map $c : E \to \{1, \ldots, k\}$ so that for any pair $e, f \in E$ of incident edges (i.e. $|e \cap f| = 1$) one has $c(e) \neq c(f)$. In other words, it is a coloring of the edges of the graph with k colors so that incident edges receive different colors. We denote the *edge coloring number* $\chi'(G)$ as the minimum number of colors needed to color the edges in G. It is useful to note that given an edge coloring c, each *color class* $c^{-1}(i)$ forms a matching in G and so $\chi'(G)$ equals the minimum number of matchings needed to cover E.

It is not hard to see that $\chi'(G) \geq \Delta(G)$. Also, a greedy coloring strategy can easily give the bound of $\chi'(G) \leq 2\Delta(G) - 1$. Surprisingly, the edge coloring number is always almost exactly the degree. The original result is due to Vizing; here we give the proof from [Sch03].

Theorem 7.9 (Vizing). *In any undirected graph $G = (V, E)$ one has $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$.*

Before we come to the main proof, we show an auxiliary result that will allow us to extend valid colorings.

Proposition 7.10. *Let $G = (V, E)$ be a graph and let $v \in V$ and $k \in \mathbb{N}$ so that:*

(i) $G - v$ is k-edge colorable

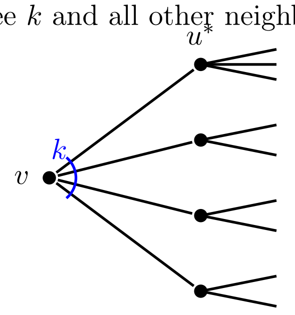
(ii) v and all neighbors of v have degree at most k

(iii) at most one neighbor of v has degree exactly k.

Then G is k-edge colorable.

Proof. We prove the statement by induction on k. If $k = 1$, then either $d_G(v) = 0$ and there is nothing to show, or there is a single edge $\{v, u^*\}$ in which case we are done as well.

Now assume $k > 1$. We may insert dummy edges and vertices so that v and exactly one neighbor u^* have degree k and all other neighbors have degree $k - 1$.



By assumption there is a k-coloring $c : E \setminus \delta(v) \to \{1, \ldots, k\}$ for the graph $G - v$. For color $i \in \{1, \ldots, k\}$, let $X_i := \{u \in N(v) \mid u$ misses color i in $c\}$. There maybe more than one possible coloring and so we select one that minimizes the quantity

$$\sum_{i=1}^{k} |X_i|^2$$

Intuitively, this means we want a coloring that distributes colors evenly among the neighbors of v. This has the following consequence:
Claim I. *There is an index $i \in [k]$ with $|X_i| = 1$.*
Proof of Claim I. Suppose for the sake of contradiction that $|X_i| \neq 1$ for all

$i = 1, \dots, k$. Note that in $G - v$, all nodes in $N(v)$ have degree one less than in G. That means u^* misses one color in c and all the nodes in $N(v) \setminus \{u^*\}$ are missing exactly two colors. In particular, summing over all missing colors gives

$$\sum_{i=1}^{k} |X_i| = 1 + 2 \cdot (d_G(v) - 1) = 2d_G(v) - 1 \overset{d_G(v) \leq k}{<} 2k \quad (*)$$

Hence there is an index i with $|X_i| < 2$ which by assumption means that $|X_i| = 0$. Also the left hand side of $(*)$ is odd, hence there has to be an index j where $|X_j|$ is odd. Since $|X_j| \neq 1$, this means $|X_j| \geq 3$. Consider the subgraph $H := (V, c^{-1}(\{i, j\}))$ formed by the edges of color i and j. Each color class $c^{-1}(i)$ and $c^{-1}(j)$ is a matching and their union is a graph where components are paths (including singletons) or circuits. There has to be one path P with more vertices in X_j than in X_i, meaning that one endpoint is in X_j and the other one is not in X_i. Then exchanging colors i and j on this path gives another k-coloring where $|X_i|^2 + |X_j|^2$ is reduced. This is a contradiction. □

After changing indices, using Claim II, we may assume that $|X_k| = 1$, say $X_k = \{w\}$. In other words, w is the only node in $N(v)$ missing color k. Let $G' := (V, E \setminus (\{v, w\} \cup c^{-1}(k)))$ be the graph obtained by deleting $\{v, w\}$ and the edges colored with k from G. Then $G' - v$ is $(k-1)$-edge colorable (simply use c without color k) and $d_{G'}(u) = d_G(u) - 1$ for all $u \in N(v) \cup \{v\}$. By induction there exists a $(k-1)$-edge coloring $\tilde{c}$ for G'. We restore the edges with color k and set $\tilde{c}(v, w) := k$. This gives a valid k-edge coloring for G. □

Now we can finish Vizing's Theorem:

Proof 7.9. We prove Vizing's theorem by induction over $|V|$ where the claim is true for $|V| = 1$. Let G be any graph and fix any vertex $v \in V$. By induction, $\chi'(G-v) \leq \Delta(G - v) + 1 \leq \Delta(G) + 1 =: k$. Then v and all its neighbors have degree strictly less than k, hence Prop 7.10 applies and also $\chi'(G) \leq k$. □

Note that the claim of Prop 7.10 is stronger than we needed in the application.

7.5 Exercises

Exercise 7.1.
Let $G = (V, E)$ be a 3-regular graph without any bridge. Show that G has a perfect matching. (A *bridge* is an edge e not contained in any circuit; equivalently, deleting e increases the number of components; equivalently $\{e\}$ is a cut. A graph G is *k-regular* if all vertices have degree k.)
Source: This exercise is taken from Schrijver [Sch17].

Exercise 7.2.

Let $G = (V, E)$ be a graph and let $T \subseteq V$. Prove that G has a matching covering T if and only if the number of odd components of $G \setminus W$ contained in T is at most $|W|$, for each $W \subseteq V$.
Source: This exercise is taken from Schrijver [Sch17].

Exercise 7.3.
Recall that for a graph $G = (V, E)$ we have defined $P_{\text{matching}}(G) = \text{conv}\{\chi^M \in \mathbb{R}^E \mid M \subseteq E \text{ is a matching}\}$. Prove that for any $k \in \mathbb{N}$,

$$P_{\text{matching}}(G) \cap \{x \in \mathbb{R}^E \mid \mathbf{1}^T x = k\} = \text{conv}\{\chi^M \mid M \subseteq E \text{ is a matching with } |M| = k\}$$

Hint: Let $\mathcal{F} := \{M \subseteq E \mid M \text{ is a matching}\}$. For the non-trivial direction, take a vector $x^* \in P_{\text{matching}}(G) \cap \{x \in \mathbb{R}^E \mid \mathbf{1}^T x = k\}$ and consider the vector $\lambda \in \mathbb{R}^{\mathcal{F}}_{\geq 0}$ that minimizes the function $G(\lambda) := \sum_{M \in \mathcal{F}} \lambda_M \cdot \big||M| - k\big|$ subject to $\sum_{M \in \mathcal{F}} \lambda_M = 1$ and $x^* = \sum_{M \in \mathcal{F}} \lambda_M \chi^M$.
Source: This exercise is taken from Schrijver [Sch17].

Exercise 7.4.
Proof that in any bipartite graph $G = (V, E)$ one has $\chi'(G) = \Delta(G)$.

Chapter 8

Interior Point Methods*

We extensively covered the theory of linear programming in Chapter 3 and its applications to discrete optimization in Chapter 5. However, we did not show how to actually solve LPs efficiently. That is the goal of this chapter. There are three classes of algorithms for solving linear programs:

- *The Simplex method.* The first known method to solve LPs, developed by Dantzig. Geometrically speaking, one optimizes a linear function $x \mapsto c^T x$ over a polyhedron by moving from an extreme point to a neighboring extreme point while improving the objective. While efficient in practice, there is no proof known that any variant of the simplex method would require only polynomially many iterations.

- *The Ellipsoid method.* The first method that was proven to take (weakly) polynomial time to solve LPs. The method constructs a sequence of shrinking ellipsoids $\mathcal{E}_0 \supseteq \mathcal{E}_1 \supseteq \ldots$ that all contain a target convex set K. While useful in theory, in practice this method is not competitive with the others.

- *Interior Point Methods.* A class of algorithms that is based on continuous optimization techniques. Currently giving the fastest theoretical running times to solve LPs while also being competetive in practical applications.

In this chapter, we describe one particular algorithm that is a *path-following interior point method.* Our presentation is loosely based on Chapter 5.3 in the monograph by Bubeck [Bub15] but we keep our exposition less abstract and include essentially all proofs.

8.1 Preliminaries

We will describe a polynomial time algorithm that solves the optimization problem $\min\{c^T x \mid x \in P\}$ where $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ is a polytope with a matrix

$A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$ are vectors. The main result in a simplified form is as follows:

Theorem 8.1. *For $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$ with $m \leq O(n)$ one can solve the LP $\min\{c^T x \mid Ax \leq b\}$ in time $O(n^{3.5}L)$, where L is the number of bits needed to represent A, b, c.*

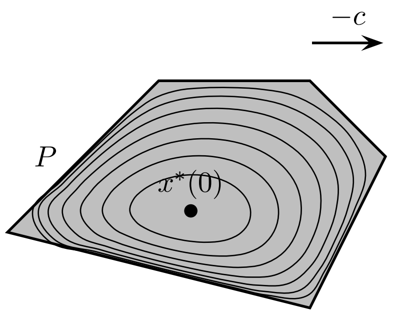
We assume that P is a full-dimensional polytope and hence the *interior* $\text{int}(P) := \{x \in \mathbb{R}^n \mid Ax < b\}$ is non-empty. We also assume that we know at least one point in $\text{int}(P)$. Let $s_i(x) := b_i - A_i^T x$ be the *slack* that x has with respect to the ith constraint. It is not further important, but if one likes, one can normalize the rows so that $\|A_i\|_2 = 1$ for $i = 1, \dots, m$ and then $s_i(x)$ gives the geometric distance of x to the ith hyperplane. For some parameter $t \geq 0$ consider the convex *log-barrier function*

$$F_t(x) := t \cdot c^T x + \sum_{i=1}^m \ln\Big(\frac{1}{s_i(x)}\Big).$$

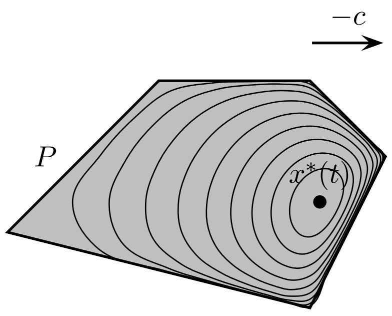
Later we will occasionally use $F_0(x)$ which is only the log-barrier term. We define

$$x^*(t) := \text{argmin}\{F_t(x) \mid x \in \mathbb{R}^n\}$$

as the unique minimizer, where we interpret $F_t(x) = \infty$ if $x \notin \text{int}(P)$. It is not hard to imagine that if $t \to \infty$, more weight is put on the linear term and $x^*(t)$ will converge to the optimum solution of $\min\{c^T x \mid x \in P\}$.

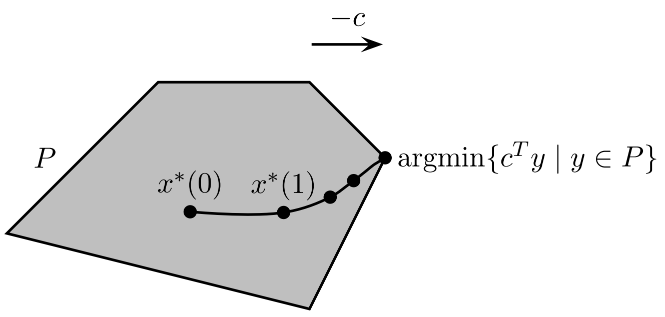


level curves for F_t for $t = 0$



level curves for F_t for $t \gg 0$

The points $\{x^*(t)\}_{t\geq 0}$ define a curve inside P that is called the *central path*.



In fact, the interior point method will approximately follow the central path to converge to the optimum. A simple calculation yields the *gradient* of the barrier function as

$$\nabla F_t(x) = t \cdot c - \sum_{i=1}^{m} \frac{A_i}{s_i(x)}$$

Moreover, the *Hessian* of the barrier function as

$$\nabla^2 F_t(x) = \sum_{i=1}^{m} \frac{A_i A_i^T}{s_i(x)^2}$$

Observe that the Hessian is an $n \times n$ symmetric, positive definite matrix; in our case it is independent of t and c. We define the *Dikin ellipsoid* of radius R around x as

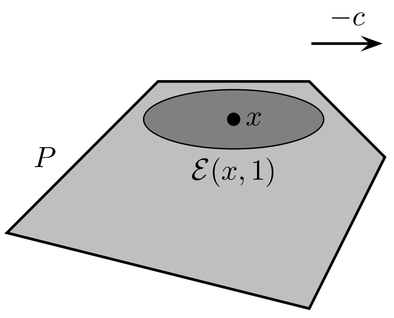
$$\begin{aligned} \mathcal{E}(x,R) &:= \left\{ y \in \mathbb{R}^n \mid \sum_{i=1}^{m} \frac{(s_i(y) - s_i(x))^2}{s_i(x)^2} \leq R^2 \right\} \\ &= \left\{ y \in \mathbb{R}^n \mid (y-x)^T [\nabla^2 F_t(x)](y-x) \leq R^2 \right\}. \end{aligned}$$

As we can see, the Hessian of F_t is also the matrix that defines the ellipsoid.

Lemma 8.2. *For any $x \in int(P)$ one has $\mathcal{E}(x,1) \subseteq P$.*

Proof. Let $y \in \mathcal{E}(x,1)$. Then in particular $(s_i(y) - s_i(x))^2 \leq s_i(x)^2$ which implies $s_i(y) \geq 0$ for all i. □

This means that starting at a point x, it is safe to move to another point $x' \in \mathcal{E}(x,1)$ without the danger of leaving P.



Geometrically one can see that if x is very close to the ith boundary constraint, then the ellipsoid $\mathcal{E}(x,1)$ is getting very thin in direction A_i.

8.2 The algorithm

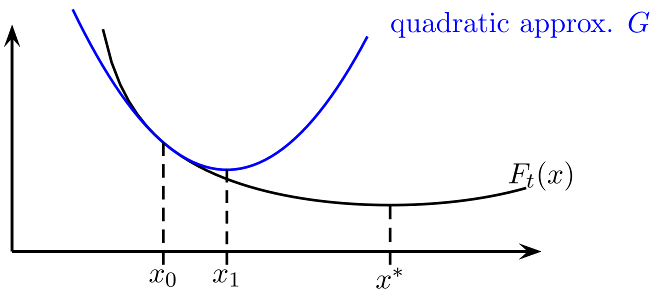
Suppose we have a starting point $x_0 \in \mathrm{int}(P)$ and some fixed t and we try to move closer to the current optimum $x^* := x^*(t)$. Consider the *quadratic approximation*

$$G(x) = F_t(x_0) + (\nabla F_t(x_0))^T(x - x_0) + \frac{1}{2}(x - x_0)[\nabla^2 F_t(x_0)](x - x_0)$$

at this point. Of course, we can obtain an explicit optimum solution for any quadratic function. The *first order optimality condition* tells us that the minimizer x_1 of G satisfies

$$\nabla G(x_1) = \nabla F_t(x_0) + [\nabla^2 F_t(x_0)](x_1 - x_0) \overset{!}{=} \mathbf{0} \quad \Rightarrow \quad x_1 = x_0 - [\nabla^2 F_t(x_0)]^{-1}(\nabla F_t(x_0))$$

A 1-dimensional visualization would like like this:



Replacing x_0 by the point x_1 that minimizes the quadratic approximation is also called a *Newton step*. It turns out that the distance to the optimum point x^* decreases quadratically if the starting point x_0 is close enough to x^*. Back to our interior point method, this means that applying a Newton iteration to a point $x \in \mathcal{E}(x^*(t), R)$ moves it closer to $x^*(t)$ in terms of the Dikin radius, assuming that R was small enough. Once the current point is close enough to $x^*(t)$ we can then increase the value of t by a factor $1 + \Theta(\frac{1}{\sqrt{m}})$. The full algorithm is as follows:

Path Following Interior Point Method

- **Input:** LP $\min\{c^T x \mid x \in P\}$ and $x_0 \in \mathcal{E}(x^*(t_0), \frac{1}{24})$ for some $t_0 > 0$.
- **Output:** Sequence $\{x_k\}_{k \geq 0}$ converging to $y^* = \mathrm{argmin}\{c^T y \mid y \in P\}$

```
(1) FOR $k = 0$ TO $\infty$ DO
    (2) Perform Newton Step $x_{k+1} := x_k - [\nabla^2 F_{t_k}(x_k)]^{-1}(\nabla F_{t_k}(x_k))$
    (3) Update $t_{k+1} := t_k \cdot (1 + \frac{1}{100\sqrt{m}})$
```

The analysis breaks down in the following main steps:

(1) First we will analyze the Newton step and prove that $x_k \in \mathcal{E}(x^*(t_k), R) \Rightarrow x_{k+1} \in \mathcal{E}(x^*(t_k), 9R^2)$. Setting $R = \frac{1}{24}$ means that $x_{k+1} \in \mathcal{E}(x^*(t_k), \frac{1}{64})$. We could in fact iterate the Newton step to get arbitrarily close to $x^*(t_k)$ — but that would not give an asymptotic advantage.

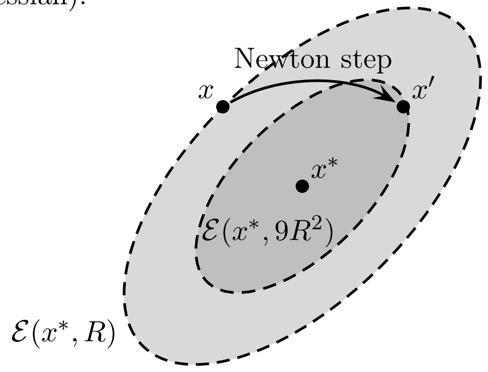
(2) For step (3), we need to show that the point on the central curve does not move too quickly when increasing the parameter t. In fact we show that $x^*(t \cdot (1+\varepsilon)) \in \mathcal{E}(x^*(t), \varepsilon\sqrt{m})$. Setting $\varepsilon := \frac{1}{100\sqrt{m}}$ is enough to guarantee that $x^*(t_{k+1}) \in \mathcal{E}(x^*(t_k), \frac{1}{100})$. As the ellipsoids change only slowly, one has $\big(x_{k+1} \in \mathcal{E}(x^*(t_k), \frac{1}{64})$ & $x^*(t_{k+1}) \in \mathcal{E}(x^*(t_k), \frac{1}{100})\big) \Rightarrow x_{k+1} \in \mathcal{E}(x^*(t_{k+1}), \frac{1}{24})$.

(3) Finally we can bound the distance from any intermediate point to the optimum $y^* := \text{argmin}\{c^T y \mid y \in P\}$ by proving that $x \in \mathcal{E}(x^*(t), 1) \Rightarrow c^T x \leq c^T y^* + O(\frac{m}{t})$.

(4) We did not describe yet how to obtain a starting point x_0 that is close enough to $x^*(t_0)$ for some $t_0 > 0$. It turns out that one can run a "reverse path following algorithm" to move from any point $x \in \text{int}(P)$ to a point close to the analytic center.

8.3 Analysis of a Newton Step

For a symmetric matrix $H \in \mathbb{R}^{n \times n}$ with Eigen decomposition $H = \sum_{i=1}^n \lambda_i u_i u_i^T$, let $|H| := \sum_{i=1}^n |\lambda_i| u_i u_i^T$ be the matrix where the absolute value function has been applied to all Eigenvalues. In particular $|H| \succeq 0$. For a $H \in \mathbb{R}^{n \times n}$ that is not necessarily symmetric, let $\|H\|_{\text{op}}$ be the largest singular value. In particular we will use that $\|Hx\|_2 \leq \|H\|_{\text{op}} \cdot \|x\|_2$ for any vector x. We will now show that a Newton step shrinks the distance from the optimum (if the distance is measured in the norm induced by the Hessian).



Lemma 8.3. *Fix a value of $t \geq 0$ and let $x^* := x^*(t)$. For $x \in \mathcal{E}(x^*, R)$ with $R \leq \frac{1}{8}$, set*

$$x' := x - [\nabla^2 F_t(x)]^{-1}(\nabla F_t(x))$$

Then $x' \in \mathcal{E}(x^*, 9R^2)$.

Proof. We begin with proving two useful claims. First we show that the Hessian of F_t only changes slowly:
Claim I. *For any vector y be on the line segment between x and x^* one has $(1-3R)\cdot\nabla^2 F_t(x) \preceq \nabla^2 F_t(y) \preceq (1+3R)\cdot\nabla^2 F_t(x)$.*
Proof of claim. We have

$$\frac{|s_i(y)-s_i(x)|}{s_i(x)} \leq \frac{|s_i(x^*)-s_i(x)|}{s_i(x)} = \underbrace{\frac{|s_i(x^*)-s_i(x)|}{s_i(x^*)}}_{\leq R}\cdot\underbrace{\frac{s_i(x^*)}{s_i(x)}}_{\leq 1+R\leq\frac{9}{8}} \leq \frac{9}{8}R$$

Inverting and squaring the inequality $(1-\frac{9}{8}R)s_i(x) \leq s_i(y) \leq (1+\frac{9}{8}R)s_i(x)$ gives the claim since $\frac{1}{(1+\frac{9}{8}R)^2} \geq 1-3R$ and $\frac{1}{(1-\frac{9}{8}R)^2} \leq 1+3R$ for $R \leq \frac{1}{8}$. □
Claim II. *One can write $\nabla F_t(x) = ([\nabla^2 F_t(x)]+E)(x-x^*)$ where $|E| \preceq 3R\cdot\nabla^2 F_t(x)$.*
Proof of claim. We apply the fundamental theorem of calculus to obtain

$$\nabla F_t(x) = \nabla F_t(x) - \underbrace{\nabla F_t(x^*)}_{=\mathbf{0}} = \underbrace{\Big[\int_0^1 \nabla^2 F_t(\lambda x + (1-\lambda)x^*)]d\lambda\Big]}_{=:\nabla^2 F_t(x)+E}(x-x^*)$$

with $|E| \preceq 3R\cdot\nabla^2 F_t(x)$ as we can derive from Claim I. □

Now we can write

$$\begin{aligned} x'-x^* &\overset{\text{Def. } x'}{=} (x-x^*) - [\nabla^2 F_t(x)]^{-1}(\nabla F_t(x)) \qquad (*)\\ &\overset{\text{Claim II}}{=} (x-x^*) - [\nabla^2 F_t(x)]^{-1}([\nabla^2 F_t(x)]+E)(x-x^*)\\ &= -[\nabla^2 F_t(x)]^{-1}E(x-x^*) \end{aligned}$$

Claim III. *One has $x' \in \mathcal{E}(x^*, 9R^2)$.*
Proof of claim. Instead of a careful (and annoying) calculation with matrices that have different Eigenvectors, we can use another trick. We apply a linear transformation to P so that $\nabla^2 F_t(x^*) = I_n$. Then $x \in \mathcal{E}(x^*, R) \Leftrightarrow \|x-x^*\|_2 \leq R$. Moreover, $(1-3R)I_n \preceq \nabla^2 F_t(x) \preceq (1+3R)I_n$ and $-6R\cdot I_n \preceq E \preceq 6R\cdot I_n$. Then

$$\|x'-x^*\|_2 = \|\nabla^2 F_t(x)^{-1}E(x-x^*)\|_2 \leq \underbrace{\|[\nabla^2 F_t(x)]^{-1}\|_{\text{op}}}_{\leq\frac{1}{1-3R}\leq\frac{3}{2}}\cdot\underbrace{\|E\|_{\text{op}}}_{\leq 6R}\cdot\underbrace{\|x-x^*\|_2}_{\leq R} \leq 9R^2$$

and hence $x' \in \mathcal{E}(x^*, 9R^2)$. □

8.4 Bounding the movement of $x^*(t)$

One of the main arguments is that the parameter t can be increased by a $(1+\Theta(\frac{1}{\sqrt{m}}))$-factor while the new optimum $x^*(t')$ still lies in the Dikin ellipsoid around $x^*(t)$.

First, we observe that around an optimum point $x^*(t)$ the function F_t is well approximated by the square of the Dikin radius. We can give a general approximation result (here the term $\pm R^3$ means that the equation holds up to an error that lies in $[-R^3, R^3]$):

Lemma 8.4. *Let $x \in int(P)$ and $x + h$ on the boundary of $\mathcal{E}(x, R)$ for $0 \leq R \leq \frac{1}{2}$. Then*

$$F_t(x+h) = F_t(x) + \langle \nabla F_t(x), h\rangle + \frac{R^2}{2} \quad \pm R^3$$

Proof. We write

$$\begin{aligned} F_t(x+h) - F_t(x) &= t \cdot c^T h - \sum_{i=1}^m \ln\Big(\frac{s_i(x+h)}{s_i(x)}\Big) \\ &= t \cdot c^T h - \sum_{i=1}^m \ln\Big(1 + \frac{\langle A_i, h\rangle}{s_i(x)}\Big) \\ &= \underbrace{t \cdot c^T h - \sum_{i=1}^m \frac{\langle A_i, h\rangle}{s_i(x)}}_{=\langle \nabla F_t(x), h\rangle} + \frac{1}{2} \underbrace{\sum_{i=1}^m \frac{\langle A_i, h\rangle^2}{s_i(x)^2}}_{=h^T \nabla^2 F_t(x) h = R^2} \pm \sum_{i=1}^m \frac{|\langle A_i, h\rangle|^3}{s_i(x)^3} \\ &= \langle \nabla F_t(x), h\rangle + \frac{R^2}{2} \pm \max_{i=1,\dots,m} \Big\{ \underbrace{\frac{|\langle A_i, h\rangle|}{s_i(x)}}_{\leq R} \Big\} \cdot \underbrace{\sum_{i=1}^m \frac{\langle A_i, h\rangle^2}{s_i(x)^2}}_{=R^2} \end{aligned}$$

using that $\ln(1+z) = z - \frac{1}{2}z^2 \pm |z|^3$ for $|z| \leq \frac{1}{2}$. □

For an arbitrary point $x \in \text{int}(P)$, the function $t \cdot c^T x$ might vary arbitrarily over the Dikin ellipsoid $\mathcal{E}(x, R)$. Interestingly the function can only vary by $R\sqrt{m}$ if x is an optimum point to F_t.

Lemma 8.5. *One has* $\max\{t \cdot c^T(x - x^*(t)) \mid x \in \mathcal{E}(x^*(t), R)\} \leq R\sqrt{m}$.

Proof. Consider the ratios $r_i := \frac{s_i(x) - s_i(x^*(t))}{s_i(x^*(t))}$. By first order optimality

$$\nabla F_t(x^*(t)) = t \cdot c - \sum_{i=1}^m \frac{A_i}{s_i(x^*(t))} = \mathbf{0}$$

Multiplying this vector equation with $x - x^*(t)$ reveals that

$$t \cdot c^T(x - x^*(t)) = \sum_{i=1}^m \frac{A_i^T(x - x^*(t))}{s_i(x^*(t))} = \sum_{i=1}^m r_i \leq \|r\|_1 \overset{r \in \mathbb{R}^m}{\leq} \sqrt{m} \cdot \underbrace{\|r\|_2}_{\leq R} \leq R \cdot \sqrt{m}$$

□

Finally we can prove an important fact:

Lemma 8.6. *Fix values of $t, t' > 0$ so that $x^*(t')$ lies on the boundary of $\mathcal{E}(x^*(t), R)$ for $0 \leq R \leq \frac{1}{4}$. Then $\frac{t'}{t} \geq 1 + \frac{R}{4\sqrt{m}}$.*

Proof. As $x^*(t')$ lies on the boundary of $\mathcal{E}(x^*(t), R)$ we can apply Lemma 8.4 to get $F_t(x^*(t')) = F_t(x^*(t)) + \frac{R^2}{2} \pm R^3$ as $\nabla F_t(x^*(t)) = \mathbf{0}$. Abbreviate $t' = t \cdot (1+\varepsilon)$. Then

$$\begin{aligned} 0 &\overset{\text{optimality}}{\leq} F_{t'}(x^*(t)) - F_{t'}(x^*(t')) \\ &= \varepsilon \cdot \underbrace{t \cdot c^T(x^*(t) - x^*(t'))}_{\leq R\sqrt{m}} + \underbrace{(F_t(x^*(t)) - F_t(x^*(t')))}_{\leq -\frac{R^2}{2}+R^3 \leq -\frac{R^2}{4}} \leq \varepsilon R\sqrt{m} - \frac{R^2}{4} \end{aligned}$$

Rearranging gives $\varepsilon \geq \frac{R}{4\sqrt{m}}$. □

8.5 Distance from the optimum

Finally we prove an upper bound on the optimality gap as t grows:

Lemma 8.7. *For any $t > 0$ and $x \in \mathcal{E}(x^*(t), 1)$ one has $c^T x - \min\{c^T y \mid y \in P\} \leq \frac{3m}{t}$.*

Proof. Let y^* be the point minimizing $c^T y$ over P. The function value $c^T x$ differs only by an additive $\frac{\sqrt{m}}{t} \leq \frac{m}{t}$ term over points in $\mathcal{E}(x^*(t), 1)$ as we have seen in Lemma 8.5. Hence it suffices to show that $c^T x^*(t) \leq c^T y^* + \frac{2m}{t}$.

Consider the midpoint $x' := \frac{1}{2}y^* + \frac{1}{2}x^*(t)$. We know that $s_i(x') \geq \frac{1}{2}s_i(x^*(t))$ for each $i \in [m]$. Hence

$$\begin{aligned} 0 &\overset{\text{optimality}}{\leq} F_t(x') - F_t(x^*(t)) = t \cdot \underbrace{(c^T x' - c^T x^*(t))}_{=\frac{1}{2}(c^T y^* - c^T x^*(t))} + \sum_{i=1}^m \underbrace{\Big(\ln\Big(\frac{1}{s_i(x')}\Big) - \ln\Big(\frac{1}{s_i(x^*(t))}\Big)\Big)}_{\leq 1} \\ &\leq \frac{t}{2}(c^T y^* - c^T x^*(t)) + m \end{aligned}$$

Rearranging gives the claim. □

8.6 Finding the analytical center

In the Interior Point Method that we stated above, we do assume that we know the *analytical center* $x^*(0)$ (or at least a very close point). We want to quickly argue how that center can be found — assuming that we know an arbitrary point $y \in \text{int}(P)$.

First, we define an auxiliary function

$$\tilde{F}_t(x) := t \cdot (-\nabla F_0(y))^T x + \sum_{i=1}^{m} \ln\left(\frac{1}{s_i(x)}\right)$$

that differs from F_t only in the linear part. Let $\tilde{x}^*(t) := \text{argmin}\{\tilde{F}_t(x) \mid x \in \mathbb{R}^n\}$ be the central path with respect to $\tilde{F}_t$. In particular we have defined the linear part in $\tilde{F}_t(x)$ so that $\nabla \tilde{F}_1(y) = \mathbf{0}$, which implies that $\tilde{x}^*(1) = y$. The 2nd trick is that the analysis (in particular Lemma 8.6) also applies if we *decrease* t by a factor $1 - \frac{1}{100\sqrt{m}}$ and the interior point will stay close the the central path of $\tilde{F}_t$. Then for $t \to 0$ we obtain a sequence of points that converge to the analytical center $\tilde{x}^*(0)$ which is also $x^*(0)$. We can summarize the algorithm as follows:

Interior Point Method for finding Analytical Center

- **Input:** Polytope $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ and point $y \in \text{int}(P)$
- **Output:** Sequence $\{x_k\}_{k \geq 0}$ converging towards analytical center

```
(1) Set $F(x) := \sum_{i=1}^m \ln\left(\frac{1}{s_i(x)}\right)$
(2) Set $\tilde{F}_t(x) := t \cdot (-\nabla F(y))^T x + F(x)$.
(3) Set $x_0 := y$ and $t_0 := 1$
(4) FOR $k = 0$ TO $\infty$ DO
      (5) Perform Newton Step $x_{k+1} := x_k - [\nabla^2 \tilde{F}_{t_k}(x_k)]^{-1}(\nabla \tilde{F}_{t_k}(x_k))$
      (6) Update $t_{k+1} := t_k \cdot (1 - \frac{1}{100\sqrt{m}})$
```

Obviously this raises the question how to find a feasible point in P after all. But one can simply solve $\min\{\lambda \mid Ax \leq b + \lambda\mathbf{1}\}$ which has a strictly feasible solution of $(x, \lambda) = (\mathbf{0}, \|b\|_\infty + 1)$ and an optimum solution is contained in $\text{int}(P)$, given that $\text{int}(P) \neq \emptyset$.

8.7 Running time

Finally we want to discuss the running time of the method in terms of number of *arithmetic operations*. Let L be the *number of bits* needed to encode the linear program. Suppose that $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$ and $c \in \mathbb{Z}^n$, then a safe definition is $L := \sum_{i,j}(1 + \log(|A_{ij}| + 1)) + \sum_{i=1}^m (1 + \log(|b_i| + 1)) + \sum_{j=1}^n (1 + \log(|c_j| + 1))$. To keep the calculations simple, we assume that $m = \Theta(n)$ and we will express the running time in terms of n and L.

A somewhat technical calculation shows that it suffices to find a point $x \in \text{int}(P)$ that is within a $2^{-\Theta(L)}$ term of the optimum y^*. Then one can select indices $I := \{i \in [m] \mid s_i(x) \leq 2^{-\Theta(L)}\}$ and the projection of x onto the subspace $\{y \in \mathbb{R}^n \mid A_i^T y = b_i \ \forall i \in I\}$ will recover an optimum solution to $\min\{c^T y \mid y \in P\}$. Similarly, the starting value of t can be of the form $t_0 \geq 2^{-\Theta(L)}$. Hence, the number iterations of the interior point method can be bounded by $O(\sqrt{n} \cdot L)$. Each iteration is dominated by the time that it takes to solve the *linear system* $[\nabla^2 F_{t_k}(x_k)]y = \nabla F_{t_k}(x_k)$ for y. Using Gaussian elimination, solving a linear system takes time $O(n^3)$ which results in a total running time of $O(n^{3.5}L)$ for solving a linear program.

On the other hand, matrices can be multiplied/inverted in time $O(n^\omega)$ where the best known value for the exponent is currently $\omega < 2.3729$. That means using *fast matrix multiplication*, linear programs can be solved in a total running time of $O(n^{\omega+1/2}L) \leq O(n^{2.8729}L)$.

However, it seems that fast matrix multiplication is not used in practice, so we want to describe a different speed-up that is based on *low rank updates*. The idea uses the basic fact that for a symmetric matrix $S \in \mathbb{R}^{n \times n}$, a vector $v \in \mathbb{R}^n$ and a scalar $\lambda \in \mathbb{R}$ one has the *Sherman-Morrison formula*

$$(S + \lambda vv^T)^{-1} = S^{-1} - \underbrace{\frac{\lambda}{1 + \lambda v^T S^{-1} v}}_{\in \mathbb{R}} \cdot \underbrace{(S^{-1}v)(S^{-1}v)^T}_{\text{rank-1 matrix}}$$

(assuming that both S and $S + \lambda vv^T$ are invertible). In particular if S^{-1} is known, then $(S + \lambda vv^T)^{-1}$ can be computed in time $O(n^2)$.

Now suppose that instead of performing a Newton step $x_{k+1} = x_k - [\nabla^2 F_t(x_k)]^{-1}(\nabla F_t(x_k))$ with the exact inverse of the Hessian, we maintain the inverse S_k^{-1} for a matrix $S_k \in \mathbb{R}^{n \times n}$ satisfying $\frac{1}{1+R} S_k \preceq \nabla^2 F_t(x_k) \preceq (1+R) S_k$. Then one can still prove the implication

$$x_k \in \mathcal{E}(x^*(t), R) \quad \Rightarrow \quad x_{k+1} \in \mathcal{E}(x^*(t), O(R^2))$$

by slightly modifying the calculations in Lemma 8.3. Then again, choosing $R > 0$ as a small enough constant suffices for our purpose. Recall that $\nabla^2 F_t(x) = \sum_{i=1}^m \frac{1}{s_i(x)^2} A_i A_i^T$. The natural idea is to choose $S_k = \sum_{i=1}^m \frac{1}{d_k(i)^2} A_i A_i^T$ but only update $d_{k+1}(i) := s_i(x_{k+1})$ when the distances have changed by more than a $1 \pm \frac{R}{4}$ factor from the last update. If S_k^{-1} is known and only q distances have been updated, then computing S_{k+1}^{-1} takes time $O(qn^2)$.

Now take consecutive points $x_{k+1} \in \mathcal{E}(x_k, R)$ and consider the ratio $r_i = \frac{s_i(x_{k+1}) - s_i(x_k)}{s_i(x_k)}$. Then the amortized number of rank-1 updates caused by this Newton step are $O(\|r\|_1) \leq O(\sqrt{n} \cdot \|r\|_2) \leq O(\sqrt{n})$. That means the amortized time per iteration is $O(n^{2.5})$ and hence solving the linear program takes time $O(n^3 L)$ even without fast matrix multiplication.

A further improvement can be made by combining low rank updates and fast matrix multiplication. In fact, Lee and Sidford show that the amortized running time per iteration can be brought down to $\tilde{O}(n^2)$ (where the $\tilde{O}$-notation hides some lower order terms), which results in a total running time of $O(n^{2.5}L)$ to solve linear programs $\min\{c^T x \mid Ax \leq b\}$ where $A \in \mathbb{Z}^{O(n) \times n}$.

8.8 Exercises

Exercise 8.1.
Consider the cube $P := [0,1]^n = \{x \in \mathbb{R}^n \mid 0 \leq x_i \leq 1 \text{ for } i = 1,\ldots,n\}$. Consider a sequence of points $\{x_k\}_{k\geq 0}$ with $x_0 = (\frac{1}{2},\ldots,\frac{1}{2})$ and with the only restriction that $x_{k+1} \in \mathcal{E}(x_k, \frac{1}{2})$. Prove that it takes at least $\Omega(\sqrt{n} \cdot \log(\frac{1}{\delta}))$ iterations until x_k can be within a $\|\cdot\|_\infty$-distance of δ from the vertex $\mathbf{0}$.

Exercise 8.2.
Recall that the presented interior point method takes $O(L\sqrt{m})$ iterations to get within an additive 2^{-L} distance to the optimum for a polytope $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ with $A \in \mathbb{R}^{m\times n}$. There is indeed a way of bringing the number of iterations down to $O(L\sqrt{n})$. A deep result of Nesterov and Nemirovsky says that there is a convex function $\phi : \mathbb{R}^n \to \mathbb{R}$ that is *self-concordant* which means it satisfies the following properties for some universal constant $C > 0$:

(A) For any $0 < R \leq \frac{1}{C}$ and $x \in \mathcal{E}(x^*, R)$ one has $(1-2R)\nabla^2\phi(x) \preceq \nabla^2\phi(x^*) \preceq (1+2R)\nabla^2\phi(x)$ where we redefine the ellipsoid $\mathcal{E}(x^*, R) := \{x \in \mathbb{R}^n \mid (x-x^*)^T[\nabla^2\phi(x^*)](x-x^*) \leq R^2\}$

(B) One has $\nabla\phi(x)\nabla\phi(x)^T \preceq Cn \cdot \nabla^2\phi(x)$ for all $x \in \text{int}(P)$.

(C) If $x \to \partial P$, then $\phi(x) \to \infty$.

For $t \geq 0$ we modify the barrier function to $F_t(x) := t \cdot c^T x + \phi(x)$. Prove the following (where the O-notation is allowed to hide dependence on C):

(1) Show that for $x \in \mathcal{E}(x^*, R)$ with $x^* := x^*(t)$ and $x' := x - [\nabla^2 F_t(x)]^{-1}\nabla F_t(x)$ one has $x' \in \mathcal{E}(x^*, O(R^2))$ assuming R is small enough.

(2) One has $\max\{t \cdot c^T(x - x^*(t)) \mid x \in \mathcal{E}(x^*(t), R)\} \leq O(R\sqrt{n})$ for all $t > 0$ and $R > 0$ small enough.

Chapter 9

Matroids — Advanced topics*

1

This chapter is a continuation of Chapter 2. Recall that a *matroid* is a pair $M = (X, \mathcal{I})$ satisfying the following:

(i) *Non-emptyness:* $\emptyset \in \mathcal{I}$

(ii) *Monotonicity:* If $Y \in \mathcal{I}$ and $Z \subseteq Y$, then $Z \in \mathcal{I}$

(iii) *Exchange property:* If $Y, Z \in \mathcal{I}$ with $|Y| < |Z|$, then there is an $x \in Z/Y$ so that $Y \cup \{x\} \in \mathcal{I}$

Here, X is called the *groundset* and elements of $\mathcal{I}$ are called *independent sets*. Despite the simple axioms, matroids have enough structure to fill whole textbooks. In this chapter we will present just a few of those results.

9.1 Matroid Intersection

We already learned that one can use the greedy algorithm to find a maximum weight independent set. In this chapter, we will see that a way more complex problem also can be solved in polynomial time:

MATROID INTERSECTION
Input: Matroid $M_1 = (X, \mathcal{I}_1)$, $M_2 = (X, \mathcal{I}_2)$ on the same groundset
Goal: Find $\max\{\, |I| : I \in \mathcal{I}_1 \cap \mathcal{I}_2\}$

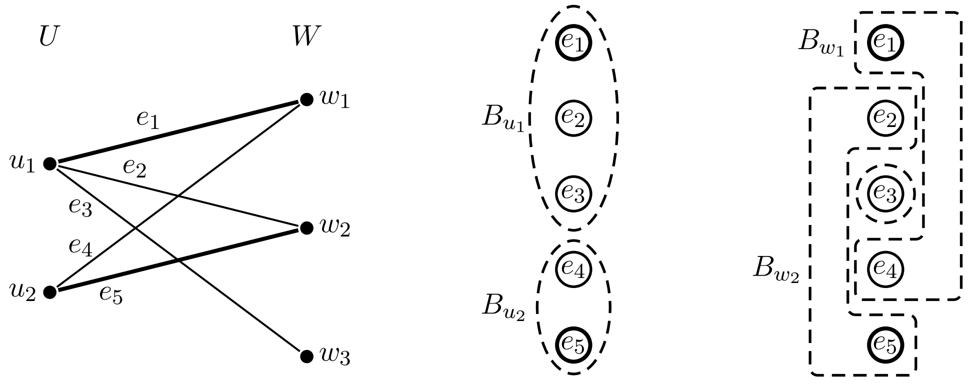
Our exposition here follows Chapter 10.4 and 10.5 in Schrijver [Sch17].

[1]July 15, 2023: Did one more iteration. ALL DONE.

9.1.1 Preliminaries

We want to begin by giving one more example of a simple class of matroids. A *partition matroid* with ground set X can be obtained as follows: take any partition $X = B_1 \dot\cup \ldots \dot\cup B_m$ and select numbers $d_i \in \{0, \ldots, |B_i|\}$. Then one can verify that $M = (X, \mathcal{I})$ with $\mathcal{I} := \{I \subseteq X : |I \cap B_i| \leq d_i \text{ for all } i = 1, \ldots, m\}$ is indeed a matroid.

To understand why matroid intersection is a non-trivial problem, we want to argue that it contains *maximum bipartite matching* as a special case. To see this, take any bipartite graph $G = (V, E)$. Suppose that $V = U \cup W$ with $U = \{u_1, \ldots, u_{|U|}\}$ and $W = \{w_1, \ldots, w_{|W|}\}$ are both sides. Then we can define two matroids that both have the edge set E as ground set as follows: take $M_1 = (E, \mathcal{I}_1)$ as the partition matroid with partitions $\delta(u_1), \ldots, \delta(u_{|U|})$, all with parameter $d_i := 1$. Similarly, we introduce $M_2 = (E, \mathcal{I}_2)$ as partition matroid with partitions $\delta(w_1), \ldots, \delta(w_{|W|})$. Now the matroid intersection problem asks to select as many edges as possible, where in each neighborhood $\delta(u_i)$ and $\delta(w_j)$ we select at most one edge. This is exactly maximum bipartite matching. See the figure below for an example:



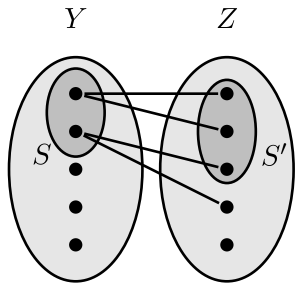
bipartite graph $G = (V, E)$ M_1 M_2

9.1.2 The exchange lemma

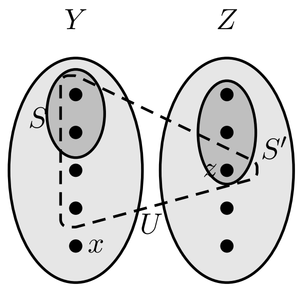
For example if we have two spanning trees T_1, T_2 in a graph, then the exchange property implies that for any $e \in T_1$, there exists *some* edge $f(e) \in T_2$ so that $(T_1 \setminus \{e\}) \cup f(e)$ is again a spanning tree. Now we will see that a stronger property is true: the map $f : T_1 \to T_2$ can be chosen to be bijective.

Lemma 9.1. *Let $M = (X, \mathcal{I})$ be a matroid and let $Y, Z \in \mathcal{I}$ be disjoint independent sets of the same size. Define a bipartite exchange graph $H = (Y \cup Z, E)$ with $E = \{(y, z) : (Y \setminus y) \cup z \in \mathcal{I}\}$. Then H contains a perfect matching.*

Proof. Suppose for the sake of contradiction that H has no perfect matching. From *Hall's condition* we know that there must be subsets $S \subseteq Y$ and $S' \subseteq Z$ so that all edges incident to S' must have their partner in S and $|S| < |S'|$.



Since $|S| < |S'|$ and S, S' are both independent sets, there is an element $z \in S'$ so that $S \cup \{z\} \in \mathcal{I}$. We can keep adding elements from Y to $S \cup \{z\}$ until we get a set $U \subseteq Y \cup \{z\}$ with $|U| = |Y|$.



There is exactly one element in $Y \setminus U$; we call it x. Then $(Y/x) \cup \{z\} = U \in \mathcal{I}$ and $(x, z) \in E$ would be an edge — a contradiction. $\square$

We will use that *exchange graph* more intensively later. Formally, for a matroid $M = (X, \mathcal{I})$ and an independent set $Y \in \mathcal{I}$, we can define $H(M, Y)$ as the bipartite graph with partitions Y and $X \setminus Y$ where we have an edge between $y \in Y$ and $x \in X \setminus Y$ if

$$(Y \setminus y) \cup \{x\} \in \mathcal{I}.$$

9.1.3 The rank function

Again, let $M = (X, \mathcal{I})$ be a matroid. Recall that an inclusionwise maximal independent set is called a *basis*. Moreover, all bases have the same size which is also called the *rank* of a matroid. One can generalize this to the *rank function* $r_M : 2^X \to \mathbb{Z}_{\geq 0}$ which is defined by

$$r_M(S) := \max\{|Y| : Y \subseteq S \text{ and } Y \in \mathcal{I}\}$$

which for a subset $S \subseteq X$ of the groundset, tells how many independent elements one can select from S.

Now suppose we have two matroids $M_1 = (X, \mathcal{I}_1)$ and $M_2 = (X, \mathcal{I}_2)$ over the same groundset. The rank function will be useful to decide at some point that we have found the largest joint independent set. Let us make the following observation:

Lemma 9.2. *Let $M_1 = (X, \mathcal{I}_1)$, $M_2 = (X, \mathcal{I}_2)$ with rank functions r_1 and r_2. Then for any independent set $Y \in \mathcal{I}_1 \cap \mathcal{I}_2$ and any set $U \subseteq X$ one has*

$$|Y| \leq r_1(U) + r_2(X/U).$$

Proof. We have

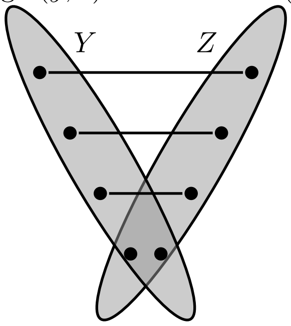
$$|Y| = \underbrace{|U \cap Y|}_{\leq r_1(U)} + \underbrace{|(X/U) \cap Y|}_{\leq r_2(X/U)} \leq r_1(U) + r_2(X/U).$$

using that Y is an independent set in both matroid. □

Later in the algorithm, we will see that this inequality is tight for some Y and U. As a side remark, for partition matroids in bipartite graphs, the lemma coincides with the fact that a vertex cover is always an upper bound to the size of any matching.

9.1.4 An reverse exchange lemma

We just saw that the exchange graph has a perfect matching between independent sets of the same size. We now show the converse, namely that a unique perfect matching between an independent set Y and any set Z implies that Z is also independent. In the following, we will consider perfect matchings in the graph $H(M, Y)$ between $Y \Delta Z$. What we mean is a perfect matching N, matching nodes in $Y \setminus Z$ to nodes in $Z \setminus Y$ and each edge $(y, z) \in N$ satisfies $(Y \setminus y) \cup \{z\} \in \mathcal{I}$.

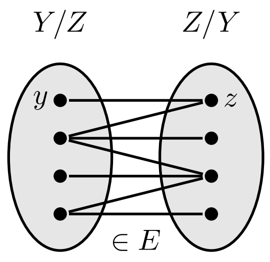


Lemma 9.3. *Let $M = (X, \mathcal{I})$ be a matroid and let $Y \in \mathcal{I}$ be an independent set and let $Z \subseteq X$ be any set with $|Z| = |Y|$. Suppose that there exists a unique perfect matching N in $H(M, Y)$ between $Y \Delta Z$. Then $Z \in \mathcal{I}$.*

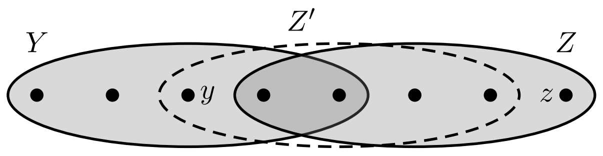
Proof. Let $E = \{(y, z) \in (Y \setminus Z) \times (Z \setminus Y) \mid (Y/y) \cup \{z\} \in \mathcal{I}\}$ be all the exchange edges between $Y \setminus Z$ and $Z \setminus Y$.
Claim. *E has a leaf*[2] *$y \in Y/Z$.*
Proof of claim. By assumption there is a perfect matching $N \subseteq E$. Start at any node $w \in Y \Delta Z$. If you are on the "right side" $Z \setminus Y$, then move along a matching

[2]Recall that a *leaf* is a degree-1 node.

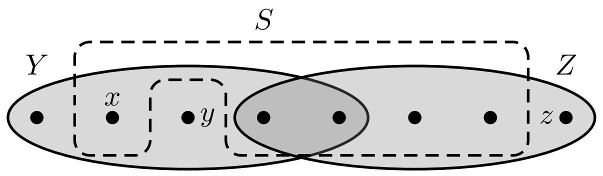
edge in N; if we are on the left hand side $Y \setminus Z$, take a non-matching edge. If we ever revisit a node, then we have found an even length path $C \subseteq E$ that alternates between matching edges and non-matching edges. Hence $N\Delta C$ is again a perfect matching, which contradicts the uniqueness. That implies that our path will not revisit a node, but that it will get stuck at some point. It cannot get stuck at a node in Z/Y because there is always a matching edge incident. Hence it can only get stuck at a node $y \in Y/Z$ that is only incident to one edge (y, z) and that edge must be in N. $\square$



Fix the element $y \in Y \setminus Z$ that is a leaf w.r.t. E and let z denote the element with $(y, z) \in N$. Note that $Z' := (Z \setminus z) \cup \{y\}$ satisfies $|Y\Delta Z'| = |Y\Delta Z| - 2$ and there is still exactly one perfect matching between $Y\Delta Z'$ (which is $N \setminus \{(y, z)\}$). Hence we can apply induction and assume that $Z' \in \mathcal{I}$.



We know that $r((Y \cup Z) \setminus y) \geq r((Y \setminus y) \cup \{z\}) = |Y|$. By the matroid exchange property, there is some element $x \in (Y \cup Z)/y$ so that $S := (Z'/y) \cup \{x\}$ is an independent set of size $|Y|$. If $x = z$ then $Z = S \in \mathcal{I}$ and we are done. Otherwise, $x \in Y/Z$.



As $|S| > |Y \setminus y|$, there must be an exchange edge between y and a node in S/Y. That contradicts the choice of y. $\square$

9.1.5 The algorithm

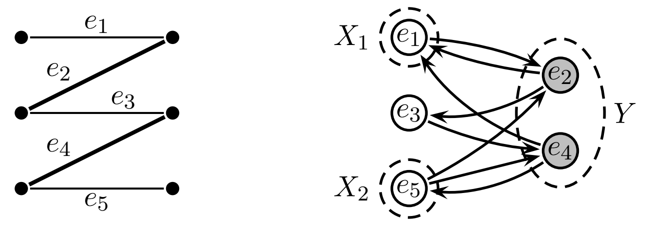
Now, suppose that we have two matroids $M_1 = (X, \mathcal{I}_1)$ and $M_2 = (X, \mathcal{I}_2)$ over the same ground set. Our algorithm starts with the independent set $Y := \emptyset$ and then augments it iteratively. Suppose we already have some joint independent set $Y \in \mathcal{I}_1 \cap \mathcal{I}_2$. We will show how to either find another set $Y' \in \mathcal{I}_1 \cap \mathcal{I}_2$ with $|Y'| = |Y| + 1$ or decide that Y is already optimal. Let us define sets

$$X_1 := \{y \in X \setminus Y \mid Y \cup \{y\} \in \mathcal{I}_1\} \quad \text{and} \quad X_2 := \{y \in X \setminus Y \mid Y \cup \{y\} \in \mathcal{I}_2\}$$

In other words, X_1 denotes the elements that could be added to the independent set Y so that we would still have an independent set in M_1. We define a directed graph $H = (X, E)$ as follows: for all $y \in Y$ and $x \in X/Y$

$$\begin{aligned} (y, x) \in E \quad &\Leftrightarrow \quad (Y/y) \cup \{x\} \in \mathcal{I}_1 \\ (x, y) \in E \quad &\Leftrightarrow \quad (Y/y) \cup \{x\} \in \mathcal{I}_2 \end{aligned}$$

Let us check what this graph does for bipartite graphs (and M_1, M_2 are the partition matroids modelling both sides). In this case Y corresponds to a matching, X_1 are edges whose left-side node is unmatched by Y and X_2 are edges whose right-side node is unmatched. We also observe that a Y-augmenting path corresponds to a directed path in H.



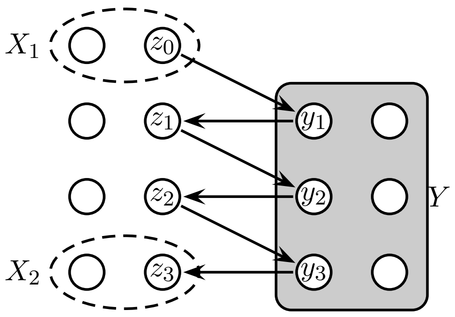
original graph exchange graph H

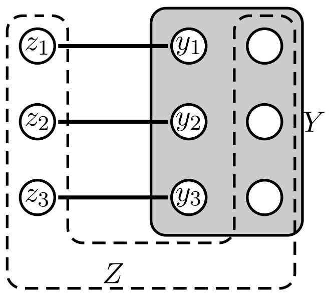
With a bit care, we can use the concept of augmenting paths also for general matroid.

Lemma 9.4. *Suppose there exists a directed path $z_0, y_1, z_1, \ldots, y_m, z_m$ starting at a vertex $z_0 \in X_1$ and ending at a node $z_m \in X_2$. If that is a shortest path, then*

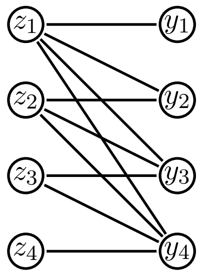
$$Y' := (Y \setminus \{y_1, \ldots, y_m\}) \cup \{z_0, \ldots, z_m\} \in \mathcal{I}_1 \cap \mathcal{I}_2$$

Proof. We will show that $Y' \in \mathcal{I}_1$, the other inclusion follows by symmetry. On the figure below, on the left hand side, we consider the directed path and on the right hand side, we consider only edges E of the exchange graph $H(M_1, Y)$ that run between $Y \setminus Z$ and $Z \setminus Y$ for $Z := (Y \setminus \{y_1, \ldots, y_m\}) \cup \{z_1, \ldots, z_m\} = Y' \setminus y_0$.





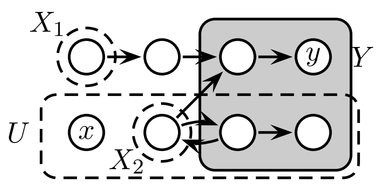
Note that the edges $\{(z_i, y_i) : i = 1, \ldots, m\}$ from the directed path form a perfect matching on $Y\Delta Z$. While E may contain more edges than that, it does not contain a *coord*, which is an edge (y_i, z_j) with $j > i$. The reason is that in this case our X_1-X_2 path would not have been the shortest possible, as we could have used the coord as shortcut. Now, consider the "complete" cordless graph $E^* := \{(y_i, z_j) : i \geq j\}$. Then this graph does have only one perfect matching. In particular, (y_1, z_1) has to be in a matching — then apply induction.



As the matching on $Y\Delta Z$ is unique, by Lemma 9.3 we have $Z = Y'/z_0 \in \mathcal{I}_1$. We know that $r_{M_1}(Y \cup Y') \geq r_{M_1}(Y \cup \{z_0\}) \geq |Y| + 1$ since $z_0 \in X_1$ is one of the "M_1-augmenting" elements. One the other hand $r_{M_1}(Y \cup Y'/\{z_0\}) \leq |Y|$ as none of the other elements of Y' is in X_1 (here we use again that we have a shortest path). Hence, the only element that could possibly augment Y'/z_0 to an independent set of size $|Y| + 1$ is z_0 itself. □

Lemma 9.5. *Suppose there is no path from a node in X_1 to a node in X_2. Then Y is optimal. In particular we can find a subset $U \subseteq X$ so that $|Y| = r_{M_1}(U)+r_{M_2}(X\backslash U)$.*

Proof. Let $U := \{i \in X : \nexists X_1 - i \text{ path in } H\}$ (or maybe more intuitively, $X \setminus U$ are the nodes that are reachable from X_1).



First, we claim that $r_{M_1}(U) = |Y \cap U|$. One direction is easy: $r_{M_1}(U) \geq r_{M_1}(U \cap Y) = |U \cap Y|$. For the other direction, suppose for the sake of contradiction that $r_{M_1}(U) > |Y \cap U|$ and hence there is some $x \in U$ so that $(Y \cap U) \cup \{x\}$ is an independent set of size $|Y \cap U| + 1$. There are two cases, depending on whether or not x also increases the rank of Y itself:

- *Case* $r_{M_1}(Y \cup \{x\}) = |Y| + 1$. Then $x \in X_1 \cap U$, which is a contradiction to the choice of U.
- *Case:* $r_{M_1}(Y \cup \{x\}) = |Y|$. Take a maximal independent set Z with $(Y \cap U) \cup \{x\} \subseteq Z \subseteq Y \cup \{x\}$. Then there is exactly one element $y \in Y/U$, so that $Z = (Y/y) \cup \{x\}$. This implies that we have would contain a directed edge (y, x). Then the node $x \in U$ is reachable from a element $y \notin U$, which contradicts the definition of U.

From the contradiction we obtain that indeed $r_{M_1}(U) = |Y \cap U|$. Similarly one can show that $r_{M_2}(X/U) = |Y \cap (X/U)|$ (which we skip for symmetry reasons). Overall, we have found a set U so that $|Y| = |Y \cap U| + |Y \cap (X \setminus U)| = r_{M_1}(U) + r_{M_2}(X \setminus U)$. □

It follows that:

Theorem 9.6. *Matroid intersection can be solved in polynomial time.*

Proof. Start from $Y := \emptyset$ and iteratively construct the directed exchange graph; compute shortest X_1-X_2 paths and augment Y as long as possible. □

The matroids that we have seen so far, all had some explicit representation. Note that the matroid intersection algorithm would work also in the *black box model*, where the only information that we have about the matroids is given by a so-called *independence oracle*. This is a procedure that receives a set $Y \subseteq X$ and simply answers whether or not this is an independent set.

Our algorithm provides a nice min-max formula for the size of joint independent sets:

Theorem 9.7 (Edmond's matroid intersection theorem). *For any matroids $M_1 = (X, \mathcal{I}_1)$ and $M_2 = (X, \mathcal{I}_2)$ one has*

$$\max\{|S| : S \in \mathcal{I}_1 \cap \mathcal{I}_2\} = \min_{U \subseteq X}\{r_{M_1}(U) + r_{M_2}(X \setminus U)\}$$

Proof. We saw the inequality "$\leq$" already in Lemma 9.2. When the matroid intersection algorithm terminates, then it has found a set U providing equality. □

9.2 Matroid partitioning

For the *matroid partitioning* problem, we are given matroids $M_1, \dots, M_k$ over the same groundset X and the goal is to cover as many elements as possible by selecting one independent set from each of the matroids. More formally:

MATROID PARTITIONING PROBLEM
Input: Matroids $M_1 = (X, \mathcal{I}_1), \dots, M_k = (X, \mathcal{I}_k)$ over ground set X.
Goal: Solve $\max\{|\dot{\bigcup}_{i=1}^k S_i| : S_i \in \mathcal{I}_i \; \forall i \in [k]\}$.

Although the problem depends on k matroids, it can be modelled by intersecting *two* matroids. Then we can derive a polynomial time algorithm by Theorem 9.6 and a min-max formula by Theorem 9.7.

9.2.1 Disjoint union

Consider matroids $M_i = (X_i, \mathcal{I}_i)$ for $i = 1, \dots, k$ where we assume that the ground sets $X_1, \dots, X_k$ are disjoint. Consider $M = (X, \mathcal{I})$ with $X := \bigcup_{i=1}^k X_i$ and $\mathcal{I} := \{S \subseteq X \mid S \cap X_i \in \mathcal{I}_i \; \forall i = 1, \dots, k\}$. Then M is called the *disjoint union* of $M_1, \dots, M_k$. It is easy to verify that M is again a matroid; it is common to write $M = M_1 \oplus \dots \oplus M_k$ to denote the disjoint union. We summarize:

Theorem 9.8 (Disjoint union)**.** *For any matroids $M_1, \dots, M_k$ with $M_i = (X_i, \mathcal{I}_i)$, the disjoint union $M = M_1 \oplus \dots \oplus M_k$ is again a matroid. Moreover its rank function is $r_M(Y) = \sum_{i=1}^k r_{M_i}(Y \cap X_i)$ for $Y \subseteq X$.*

The proof is very straightforward and we skip it here.

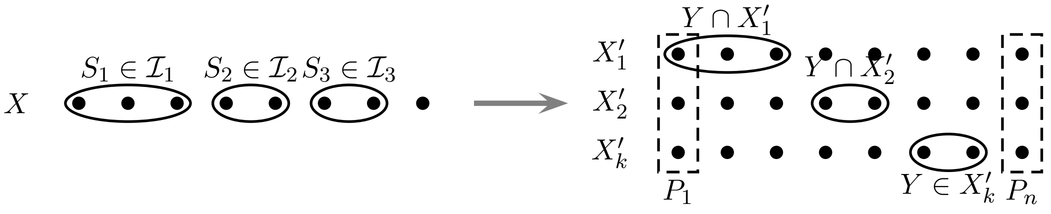
9.2.2 The reduction

Now we come to the main claim.

Theorem 9.9. *For any matroids $M_1, \dots, M_k$ with $M_i = (X, \mathcal{I}_i)$, the Matroid Partitioning problem can be solved in polynomial time. Moreover the problem satisfies the min-max formula*

$$\max\left\{\left|\dot{\bigcup}_{i=1}^k S_i\right| : S_i \in \mathcal{I}_i \; \forall i \in [k]\right\} = \min\left\{|X \setminus T| + \sum_{i=1}^k r_{M_i}(T) \mid T \subseteq X\right\}$$

Proof. Let $X := \{e_1, \dots, e_n\}$. We clone the elements k times and let $X_i := \{e_1^i, \dots, e_n^i\}$ denote the ith copy. Instead of using M_i we consider the identical matroid $M_i' = (X_i, \mathcal{I}_i')$ on the ith copy of elements, that means $\{e_j\}_{j \in J} \in \mathcal{I}_i \Leftrightarrow \{e_j^i\}_{j \in J} \in \mathcal{I}_i'$. Consider the matroid $M_U = (X', \mathcal{I}_U) := M_1' \oplus \dots \oplus M_k'$ which is the *disjoint union* of those k matroids. Then the matroid partitioning problem corresponds to selecting an independent set $Y \in \mathcal{I}_U$ of maximum cardinality so that Y contains at most one copy of each original element. To model this as matroid intersection, let $M_P = (X', \mathcal{I}_P)$ be the *partitioning matroid* with partitions $P_j := \{e_j^1, \dots, e_j^k\}$ (and parameter $b_j := 1$).



Then the optimum value of the matroid partitioning instance can be written as

$$\begin{aligned}\max\{|Y| : Y \in \mathcal{I}_U \cap \mathcal{I}_P\} &\overset{\text{Thm 9.7}}{=} \min\left\{r_{M_P}(X' \setminus T') + r_{M_U}(T') \mid T' \subseteq X'\right\} \\ &= \min\left\{|X \setminus T| + \sum_{i=1}^{k} r_{M_i}(T) \mid T \subseteq X\right\}\end{aligned}$$

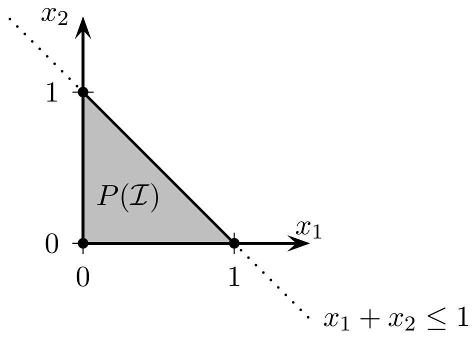
where we use the min-max formula from the Matroid Intersection Theorem (Theorem 9.7). Moreover, we have used that $r_{M_P}(T')$ counts every partition P_j that is intersected by T' and hence a minimizer T' can can always be chosen so that $|T' \cap P_j| \in \{0, |P_j|\}$. This gives the identity $T = \{e_j : P_j \subseteq T'\}$. Note that the underlying matroid intersection instance can be solved in polynomial time, see Theorem 9.6. □

9.3 The Independent Set Polytope of a Matroid

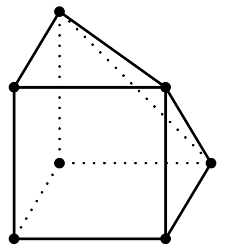
For this first section, we mostly follow the exposition in the very readable book of Cook, Cunningham, Pulleyblank and Schrijver [CCPS98]. Let $M = (X, \mathcal{I})$ be a matroid and consider the polytope

$$P(\mathcal{I}) := \text{conv}\{\mathbf{1}_S : S \in \mathcal{I}\}$$

which is the convex hull of the characteristic vectors of independent sets. This is also called the *matroid independence polytope* of M. By definition, this is a polytope whose vertices are in $\{0,1\}^X$. In fact, we have defined $P(\mathcal{I})$ in terms of its *vertices*.



$P(\mathcal{I})$ for uniform matroid with $n = 2, k = 1$



$P(\mathcal{I})$ for uniform matroid with $n = 3, k = 2$

Now we are wondering how one could characterize $P(\mathcal{I})$ with *linear inequalities*? We know that for each independent set $Y \in \mathcal{I}$ and each $S \subseteq X$ one has $|Y \cap S| \leq r_M(S)$. So it is reasonable to consider the candidate polytope

$$Q(M) := \Big\{x \in \mathbb{R}^X_{\geq 0} \mid \sum_{i \in S} x_i \leq r_M(S) \ \ \forall S \subseteq X\Big\}$$

From our reasoning we clearly know that by convexity $P(\mathcal{I}) \subseteq Q(M)$. But is it true that $P(\mathcal{I}) = Q(M)$ or did we miss some relevant inequalities in the description of $Q(M)$? It is not hard to argue that $P(\mathcal{I}) \cap \{0,1\}^X = Q(M) \cap \{0,1\}^X$. To see this, take a vector $x \in Q(M) \cap \{0,1\}^X$ and set $Y := \{i \in X \mid x_i = 1\}$. Then this point satisfies $|Y \cap S| \leq r_M(S)$ for all $S \subseteq Y$ and hence $Y \in \mathcal{I}$. Yet, these properties are still not enough to conclude that $P(\mathcal{I})$ and $Q(M)$ are equal. This will require additional arguments:

Theorem 9.10. *For any matroid $M = (X, \mathcal{I})$ one has $P(\mathcal{I}) = Q(M)$.*

Proof. To show the remaining inclusion $P(\mathcal{I}) \supseteq Q(M)$ it suffices to prove that for every vector $c \in \mathbb{R}^X$ one has

$$\max\big\{c^T x \mid x \in P(\mathcal{I})\big\} \geq \max\big\{c^T x \mid x \in Q(M)\big\} \quad (*)$$

Since $P(I)$ and $Q(M)$ are both monotone and contained in $\mathbb{R}^X_{\geq 0}$, it suffices to consider $c \in \mathbb{R}^X_{\geq 0}$. We know that the left hand side of $(*)$ is also attained by the Matroid Greedy algorithm. The right hand side of $(*)$ is the value of a linear program that we denote by (Primal(M, c)). So it suffices to prove that the solution of the Matroid Greedy algorithm is an optimum solution for the LP (Primal(M, c)). Mechanically we can develop the dual of the LP as follows, see Theorem 3.27:

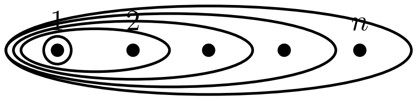
$$\max\Big\{\sum_{i \in X} c_e x_e \mid \sum_{i \in S} x_i \leq r_M(S) \ \ \forall S \subseteq X; \ \ x_i \geq 0 \ \ \forall i \in X\Big\} \qquad (\textsc{Primal}(M, c))$$

$$\min\Big\{\sum_{S \subseteq X} r_M(S) y_S \mid \sum_{S \subseteq X : i \in S} y_S \geq c_i \ \ \forall i \in X; \ y_S \geq 0 \ \ \forall S \subseteq X\Big\} \qquad (\textsc{Dual}(M, c))$$

Then it remains to show:
Claim. *For any matroid $M = (X, \mathcal{I})$ and $c \in \mathbb{R}^X$, if $Y^* \in \mathcal{I}$ is the solution found by the Matroid Greedy algorithm, then $x^* := \mathbf{1}_{Y^*}$ is an optimum solution to* $(\textsc{Primal}(M, c))$.
Proof of Claim. We will construct a vector y^* that is feasible for $(\textsc{Dual}(M, c))$ and so that the pair (x^*, y^*) satisfies the *complementary slackness conditions.* Let us write $X = \{1, \dots, n\}$ and sort the elements in the same order $c_1 \geq c_2 \geq \dots \geq c_n$ that the Matroid Greedy algorithm uses. Then set

$$y^*_S := \begin{cases} c_i - c_{i+1} & \text{if } S = \{1, \dots, i\} \text{ with } i \in \{1, \dots, n-1\} \\ c_n & \text{if } S = \{1, \dots, n\} \end{cases}$$

In fact, the support of y^* can be visualized as follows:



Interestingly, the solution y^* itself does *not* dependent on the matroid M. However, the objective function of $(\textsc{Dual}(M, c))$ depends on the matroid. First, we can see that $y^* \geq \mathbf{0}$ and for each $j \in \{1, \dots, n\}$ one has

$$\sum_{S \subseteq X : j \in S} y^*_S = \sum_{i=j}^{n} y^*_{\{e_1, \dots, e_i\}} = \sum_{i=j}^{n-1} (c_i - c_{i+1}) + c_n = c_j \quad (*)$$

Hence y^* is feasible for $(\textsc{Dual}(M, c))$. It remains to verify that the complementary slackness conditions from Theorem 3.29 are satisfied. Recall that the conditions are:

- (CS1) $x^*_i > 0 \Rightarrow \sum_{S \subseteq X : i \in S} y^*_S = c_i$
- (CS2) $y^*_S > 0 \Rightarrow \sum_{i \in S} x^*_i = r_M(S)$

(CS1) is clearly satisfied due to $(*)$. Now consider a set S with $y^*_S > 0$. By construction this means that $S = \{1, \dots, i\}$. Recall that the matroid greedy algorithm always maintains a basis of the considered elements, i.e. $|Y^* \cap \{1, \dots, i\}| = r_M(\{1, \dots, i\})$ for all $i = 1, \dots, n$. That means also (CS2) is satisfied. Then the pair (x^*, y^*) are optimal primal and dual solutions and the claim follows. □

9.4 Submodularity and matroids

The following concept is extremely useful in optimization:

Definition 9.11. Let X be a finite set. A function $f : 2^X \to \mathbb{R}$ is called *submodular* if

$$f(A \cup \{j\}) - f(A) \geq f(B \cup \{j\}) - f(B) \quad \forall A \subseteq B \subseteq X \quad \forall j \in X \setminus B$$

Intuitively, a submodular function satisfies a *diminishing returns* property. Often one may find the definition of submodular functions with a different, equivalent condition:

Lemma 9.12. *A function $f : 2^X \to \mathbb{R}$ is submodular if and only if*

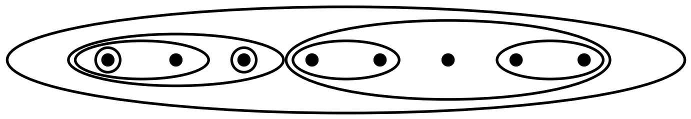
$$f(A) + f(B) \geq f(A \cap B) + f(A \cup B) \quad \forall A, B \subseteq X$$

Submodular functions naturally appear in the context of matroids:

Lemma 9.13. *Let $M = (X, \mathcal{I})$ be a matroid. Then the rank function r_M is submodular.*

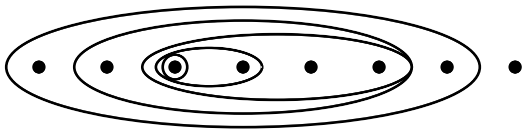
Proof. It suffices to prove the following:
Claim. *For $A \subseteq B \subseteq X$ and $j \in X \setminus B$ one has $(r_M(B \cup \{j\}) - r_M(B) = 1) \Rightarrow (r_M(A \cup \{j\}) - r_M(A) = 1)$.*
Proof of Claim. Construct an independent set S as follows: Take a basis of A, extend it to a basis of B, then extend it to a basis of $B \cup \{j\}$. Then $S \in \mathcal{I}$ with $|S \cap A| = r_M(A)$, $|S \cap B| = r_M(B)$ and $|S \cap (B \cup \{j\})| = r_M(B) + 1$. Hence $j \in S$ and $S \cap (A \cup \{j\}) \in \mathcal{I}$. That means $r_M(A \cup \{j\}) \geq |S \cap (A \cup \{j\})| = r_M(A) + 1$. $\square$

Definition 9.14. Given a finite set X, a set family $\mathcal{F} \subseteq 2^X$ is called *laminar* if for all $S, T \in \mathcal{F}$ one has either $S \subseteq T$ or $T \subseteq S$ or $S \cap T = \emptyset$.


laminar family

Definition 9.15. Given a finite set X, a set family $\mathcal{F} \subseteq 2^X$ is called a *chain* if one can write $\mathcal{F} = \{S_1, \ldots, S_m\}$ so that $S_1 \subseteq S_2 \subseteq \ldots \subseteq S_m$.

A chain is a special type of laminar family.


chain

We will illustrate a technique called *uncrossing* which can often be found in proofs dealing with submodular functions.

Lemma 9.16. *Let $f : 2^X \to \mathbb{R}$ be a submodular function and let $Q \subseteq \mathbb{R}^X$ be a polyhedron. Then the linear program*

$$\min\Big\{\sum_{S\subseteq X} z_S f(S) \mid \sum_{S\subseteq X} z_s \mathbf{1}_S \in Q,\ \ z_S \geq 0\ S \subseteq X\Big\} \quad \text{(LP)}$$

has an optimum solution z^* so that the support $\mathcal{F} := \{S \subseteq X \mid z^*_S > 0\}$ is a chain. In fact, any optimum solution minimizing $\sum_{S\subseteq X} z_S \cdot |S| \cdot |X \setminus S|$ will have that property.

Proof. Let $K \subseteq \mathbb{R}^X$ be the set of optimum solutions to (LP). Choose a vector $z^* \in K$ that minimizes the quantity $\sum_{S\subseteq X} z_S \cdot |S| \cdot |X \setminus S|$. Note that this itself is a linear program as K is a polyhedron and hence that minimum is attained. Suppose for the sake of contradiction that the support $\mathcal{F} := \{S \subseteq X \mid z^*_S > 0\}$ is not a chain. Choose two sets $S, T \in \mathcal{F}$ with $S \setminus T \neq \emptyset$ and $T \setminus S \neq \emptyset$ and set $\varepsilon := \min\{z^*_S, z^*_T\} > 0$. We can modify z^* by increasing the value of $z^*_{S\cap T}$ and $z^*_{S\cup T}$ by ε and decreasing it on z^*_S and z^*_T by ε. Let us call the modified solution z^{**}. Note that $\mathbf{1}_{S\cap T} + \mathbf{1}_{S\cup T} = \mathbf{1}_S + \mathbf{1}_T$ and so z^{**} is still feasible for (LP). The change in the objective function is

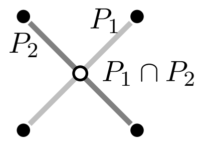
$$\varepsilon \cdot (f(S\cap T) + f(S\cup T) - f(S) - f(T)) \leq 0 \quad (**)$$

by submodularity, so $z^{**} \in K$ is also an optimum solution for (LP). This is not yet a contradiction as the inequality in $(**)$ might not have been strict. But $\sum_{S\subseteq X} z^{**}_S \cdot |S| \cdot |X \setminus S| > \sum_{S\subseteq X} z^*_S \cdot |S| \cdot |X \setminus S|$ as one can verify. This is then a contradiction. □

Note that we have reproven the insight from Theorem 9.10 that (Dual(M, c)) always has solution whose support is a chain — just that the statement in Lemma 9.16 is much more general.

9.5 The Matroid Intersection Polytope

For this section, we follow the exposition in Schrijver [Sch03], Chapter 41. In general, if we have two polytopes P_1 and P_2 with vertices in $\{0,1\}^n$, then it is *not* true that the vertices of $P_1 \cap P_2$ are in $\{0,1\}^n$ as well, see an easy example in $n = 2$ below:



But as we have seen in Section 9.1, the intersection of matroids is well behaved combinatorially, so there is some hope that the *intersection of two matroid independence polytopes* is well behaved too. And in fact, we will see the result of Edmonds that indeed $P(\mathcal{I}_1 \cap \mathcal{I}_2) = P(\mathcal{I}_1) \cap P(\mathcal{I}_2)$. In particular this implies that one can solve the weighted matroid intersection problem in polynomial time.

Again, it is immediate that $P(\mathcal{I}_1 \cap \mathcal{I}_2) \subseteq P(\mathcal{I}_1) \cap P(\mathcal{I}_2)$ and it remains to prove the reverse inclusion of $P(\mathcal{I}_1) \cap P(\mathcal{I}_2) \subseteq P(\mathcal{I}_1 \cap \mathcal{I}_2)$. As $P(\mathcal{I}_1) = Q(M_1)$ and $P(\mathcal{I}_2) = Q(M_2)$ we know that it suffices to prove that $\max\{c^T x \mid x \in Q(M_1) \cap Q(M_2)\}$ has an integral optimum solution $x \in \{0,1\}^X$ for every objective function $c \in \mathbb{R}^X$. Let us write out that linear program and develop its dual:

$$(\textsc{Primal}(M_1, M_2, c)):$$
$$\max \left\{ \sum_{i \in X} c_i x_i \mid \begin{array}{rcll} \sum_{i \in S} x_i & \leq & r_{M_1}(S) & \forall S \subseteq X \\ \sum_{i \in S} x_i & \leq & r_{M_2}(S) & \forall S \subseteq X \\ x_i & \geq & 0 & \forall i \in X \end{array} \right\}$$

$$(\textsc{Dual}(M_1, M_2, c)):$$
$$\min \left\{ \sum_{S \subseteq X} \left(r_{M_1}(S) y_S^1 + r_{M_2}(S) y_S^2 \right) \mid \begin{array}{rcll} \sum\limits_{S \subseteq X : i \in S} (y_S^1 + y_S^2) & \geq & c_i & \forall i \in X \\ y_S^1, y_S^2 & \geq & 0 & \forall S \subseteq X \end{array} \right\}$$

Note that the constraints of $(\textsc{Dual}(M_1, M_2, c))$ can also be written in the more compact vector notation $\sum_{S \subseteq X} (y_S^1 + y_S^2) \mathbf{1}_S \geq c$.

We will an auxiliary result that we state in more generality than required:[3]:

Lemma 9.17. *Let $\mathcal{C} \subseteq 2^X$ be the union of two laminar families of subsets of X and let $A \in \{0,1\}^{\mathcal{C} \times X}$ be the incidence matrix of $\mathcal{C}$. Then A is totally unimodular.*

Proof. Any submatrix of A is again the incidence matrix of the union of two laminar families, so it suffices to consider the case that A is a square matrix and prove that $\det(A) \in \{-1, 0, 1\}$. Let $\mathcal{C} = \mathcal{F}_1 \cup \mathcal{F}_2$ be the partition into the two laminar families $\mathcal{F}_1, \mathcal{F}_2 \subseteq 2^X$. Each of the families $\mathcal{F}_k$ with $k \in \{1, 2\}$ induces a *rooted forest* structure (that means it is a collection of trees that have a distinguished root) where S being a child of T implies that $S \subset T$. Now, we define a new matrix $A' \in \{0,1\}^{\mathcal{C} \times X}$ that emerges from A as follows: For each $T \in \mathcal{F}_k$, while the matrix A contains the row $\mathbf{1}_T$; the matrix A' will contain the row $\mathbf{1}_T - \sum_{S \in \mathcal{F}_k \text{ is child of } T} \mathbf{1}_S$. In particular the rows of the leafs will simply be copied over.

$$\begin{array}{c} \\ \mathcal{F}_1 \\ \\ \\ \mathcal{F}_2 \\ \\ \end{array} \left[\begin{array}{cccccc} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right] \quad \Rightarrow \quad \left[\begin{array}{cccccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

matrix A matrix A'

[3] In our application $\mathcal{C}$ will simply be the union of two chains.

Subtracting rows from other rows does not change the determinant of a matrix, hence $\det(A') = \det(A)$. Then in each column, the matrix A' at most one $+1$ entry for the first family and at most one $+1$ entry for the 2nd family. Hence A' is the node-edge incidence matrix of a bipartite undirected graph possibly plus some columns containing at most one $+1$ entry. Such a matrix is TU by Theorem 5.13. $\square$

Now we can prove the crucial structural result that we need:

Theorem 9.18. *Let $M_1 = (X, \mathcal{I}_1)$ and $M_2 = (X, \mathcal{I}_2)$ be matroids. For any $c \in \mathbb{Z}^X$, there is an optimum solution (y^1, y^2) to $(\textsc{Dual}(M_1, M_2, c))$ that is integral. Moreover that solution can be chosen so that supp(y^1) and supp(y^2) are both chains.*

Proof. Among all the optimum solutions to $(\textsc{Dual}(M_1, M_2, c))$, let $(\bar{y}^1, \bar{y}^2)$ be one minimizing the quantity

$$\sum_{S \subseteq X} (y_S^1 + y_S^2) \cdot |S| \cdot |X \setminus S| \quad (*)$$

Then let $\mathcal{F}_1 := \{S \subseteq X \mid \bar{y}_S^1 > 0\}$ and $\mathcal{F}_2 := \{S \subseteq X \mid \bar{y}_S^2 > 0\}$ be the support. We want to use Lemma 9.16 to argue that both $\mathcal{F}_1$ and $\mathcal{F}_2$ must be chains. To see this, imagine to fix $\bar{y}^2$ and reoptimize over the y^1 variables, i.e. solve

$$\min \left\{ \sum_{S \subseteq X} r_{M_1}(S) \cdot y_S^1 \mid \begin{array}{rcl} \sum_{S \subseteq X} y_S^1 \mathbf{1}_S & \geq & b \\ y_S^1 & \geq & 0 \quad \forall S \subseteq X \end{array} \right\}$$

where $b := c - \sum_{S \subseteq X} \bar{y}_S^2 \mathbf{1}_S$ gives the part of the objective function that needs to be covered by y^1. Then $\bar{y}^1$ is still an optimum solution minimizing $\sum_{S \subseteq X} y_S^1 \cdot |S| \cdot |X \setminus S|$. Crucially, the rank function r_{M_1} is submodular by Lemma 9.13, hence Lemma 9.16 applies and $\mathcal{F}_1$ is a chain. Analogously we can obtain that $\mathcal{F}_2$ is a chain.

Now consider the LP that we obtain by deleting all variables not in $\mathcal{F}_1$ or $\mathcal{F}_2$ from $(\textsc{Dual}(M_1, M_2, c))$:

$$\min \left\{ \sum_{S \in \mathcal{F}_1} r_{M_1}(S) y_S^1 + \sum_{S \in \mathcal{F}_2} r_{M_2}(S) y_S^2 \mid \begin{array}{rcl} \sum_{S \in \mathcal{F}_1} \mathbf{1}_S y_S^1 + \sum_{S \in \mathcal{F}_2} \mathbf{1}_S y_S^2 & \geq & c \\ y_S^1 & \geq & 0 \quad \forall S \in \mathcal{F}_1 \\ y_S^2 & \geq & 0 \quad \forall S \in \mathcal{F}_2 \end{array} \right\}$$

Then this LP is of the form $\min\{r^T y \mid Ay \geq c, y \geq \mathbf{0}\}$ where $A \in \{0,1\}^{X \times (\mathcal{F}_1 \cup \mathcal{F}_2)}$ is the transpose of the incidence matrix of the union of two laminar families and that matrix A is TU by Lemma 9.17. Then by Theorem 5.7 there is an integral optimum solution $(\tilde{y}^1, \tilde{y}^2)$. The objective function value of $(\tilde{y}^1, \tilde{y}^2)$ has to be at least as good as the one of $(\bar{y}^1, \bar{y}^2)$ and hence $(\tilde{y}^1, \tilde{y}^2)$ is also optimal for the full LP $(\textsc{Dual}(M_1, M_2, c))$. $\square$

Now the main result of this section easily follows:

Theorem 9.19 (Matroid Intersection Polytope Theorem (Edmonds))**.** *Let $M_1 = (X, \mathcal{I}_1)$ and $M_2 = (X, \mathcal{I}_2)$ be matroids. Then all vertices of $P(\mathcal{I}_1) \cap P(\mathcal{I}_2)$ are in $\{0,1\}^X$ and $P(\mathcal{I}_1 \cap \mathcal{I}_2) = P(\mathcal{I}_1) \cap P(\mathcal{I}_2)$.*

Proof. To show that all vertices in $P(\mathcal{I}_1) \cap P(\mathcal{I}_2) = Q(M_1) \cap Q(M_2)$ are integral, it suffices to show that for all $c \in \mathbb{Z}^X$, ($\textsc{Primal}(M_1, M_2, c)$) has an integral optimum value. This is the same as showing that ($\textsc{Dual}(M_1, M_2, c)$) has an integral optimum value for all $c \in \mathbb{Z}^X$. And that is implied by Theorem 9.18. $\square$

WEIGHTED MATROID INTERSECTION
Input: Two matroid $M_1 = (X, \mathcal{I}_1)$, $M_2 = (X, \mathcal{I}_2)$ on the same groundset and $c \in \mathbb{R}^X$.
Goal: Find $\max\{\sum_{i \in S} c_i \mid S \in \mathcal{I}_1 \cap \mathcal{I}_2\}$.

Theorem 9.20. *The Weighted Matroid Intersection problem can be solved in strongly polynomial time assuming the oracles for M_1 and M_2 admit algorithms with running time polynomial in $|X|$.*

The argument uses two ingredients that we only cover in later chapters. However, we give the proof for the sake of completeness.

Proof. By the Theorem of Frank and Tardos [FT87] (Theorem 12.1), we can replace the original objective function c with an approximation $\tilde{c}$ that has the same set of optimum solutions while the encoding length of $\tilde{c}$ is polynomial in $|X|$.

Using the Matroid Greedy algorithm we can optimize over $P(M_1)$ and $P(M_2)$ in polynomial time in $|X|$. Hence by the *equivalence of separation and optimization* (Theorem 11.9) we can also separate in polynomial time in $|X|$ over $P(M_1)$ and $P(M_2)$. Using the equivalence of separation and optimization a 2nd time, we can also optimize any linear function over $P(M_1) \cap P(M_2)$ in time polynomial in $|X|$ and the encoding length of the objective function $\tilde{c}$ which is also bounded by a polynomial in $|X|$. $\square$

As a sanity check, let us conside the consequence of Theorem 9.18 for the objective function $c := \mathbf{1}$. Then ($\textsc{Dual}(M_1, M_2, \mathbf{1})$) has an optimum integral solution (y^1, y^2) so that the support of y^1 and the support of y^2 are both chains. Recall that the only constraint is $\sum_{S \subseteq X} \mathbf{1}_S y^1_S + \sum_{S \subseteq X} \mathbf{1}_S y^2_S \geq \mathbf{1}$. Then in every chain one only needs to select the unique maximal set to an extend of 1. With that observation we can write the matroid intersection problem as

$$\begin{aligned}
\max\left\{|Y| : Y \in \mathcal{I}_1 \cap \mathcal{I}_2\right\} &\overset{\text{Thm 9.19}}{=} (\textsc{Primal}(M_1, M_2, \mathbf{1})) \\
&\overset{\text{LP-duality}}{=} (\textsc{Dual}(M_1, M_2, \mathbf{1})) \\
&= \min\left\{r_{M_1}(S) + r_{M_2}(T) \mid S \cup T = X\right\}
\end{aligned}$$

Finally by monotonicity of the rank function the sets S and T an be chosen to be disjoint. Hence we recover an alternative proof of the Matroid Intersection Theorem (Theorem 9.7).

Finally, note that given 3 matroids $M_i = (X, \mathcal{I}_i)$, $i = 1, 2, 3$, it is **NP**-hard to determine $\max\{|S| : S \in \mathcal{I}_1 \cap \mathcal{I}_2 \cap \mathcal{I}_3\}$. In particular, $P(\mathcal{I}_1 \cap \mathcal{I}_2 \cap \mathcal{I}_3) \neq P(\mathcal{I}_1) \cap P(\mathcal{I}_2) \cap P(\mathcal{I}_3)$ in general.

9.6 Exercises

Exercise 9.1.
Derive König's theorem[4] from Edmonds' matroid intersection theorem.
Source: This exercise is taken from Schrijver [Sch17].

Exercise 9.2.
Prove the following Theorem of Rado from 1942: Let $M = (X, \mathcal{I})$ be a matroid and let $X_1, \ldots, X_m \subseteq X$. Then there exists a transversal $S \in \mathcal{I}$ for $X_1, \ldots, X_m$, if and only if for any $J \subseteq [m]$ one has $r_M(\bigcup_{i \in J} X_i) \geq |J|$.

Hint: Recall that a *transversal* for sets $X_1, \ldots, X_m \subseteq X$ is a set $S \subseteq X$ such that there is an bijective map $f : S \to [m]$ with $x \in X_{f(x)}$ for $x \in S$. For any sets $X_1, \ldots, X_m \subseteq X$, the pair $M' = (X, \mathcal{I}')$ with $\mathcal{I}' := \{S \subseteq X \mid S \text{ is contained in a transversal of } X_1, \ldots, X_m\}$ is a matroid, called the *transversal matroid*. Moreover the rank function of that matroid is

$$r_{M'}(T) = \min_{J \subseteq [m]} \left\{ \left| T \cap \bigcup_{i \in J} X_i \right| + m - |J| \right\}$$

You may use these properties without proof.
Source: This exercise is taken from Schrijver [Sch17].

[4]In any bipartite graph $G = (V, E)$ one has $\nu(G) = \tau(G)$.

Chapter 10

Semidefinite programming*

In this chapter we will introduce *semidefinite programming* which is a generalization of linear programming. We will focus on two striking applications in the area of approximation algorithms. Here, an *approximation algorithm* is an algorithm that finds a solution that is provably close an optimum one in term of objective function value. Typically one employs approximation algorithms for **NP**-hard problems where exact efficient algorithms do not exist, assuming that $\mathbf{NP} \neq \mathbf{P}$.

10.1 Positive semi-definite matrices

We begin by reviewing some linear algebra facts. A matrix $X \in \mathbb{R}^{n\times n}$ is *symmetric* if $X_{ij} = X_{ji}$ for all $i, j \in [n]$. We know the following:

Fact 10.1. *For a symmetric matrix $X \in \mathbb{R}^{n\times n}$, all Eigenvalues are real.*

That means the following definition makes sense:

Definition 10.2. A symmetric matrix $X \in \mathbb{R}^{n\times n}$ is *positive semidefinite* if all its Eigenvalues are non-negative. In that case we write $X \succeq 0$.

For matrices $A, B \in \mathbb{R}^{n\times n}$ we write

$$\langle A, B\rangle := \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij} \cdot B_{ij}$$

as the *Frobenius inner product.*

Lemma 10.3. *For a symmetric matrix $X \in \mathbb{R}^{n\times n}$, the following is equivalent*

a) $a^T X a \geq 0 \; \forall a \in \mathbb{R}^n$.

b) *X is positive semidefinite.*

c) There exists a matrix U so that $X = UU^T$.

d) There are $u_1, \dots, u_n \in \mathbb{R}^r$ with $X_{ij} = \langle u_i, u_j \rangle$ for $i, j \in [n]$.

Proof. Any symmetric real matrix is *diagonalizable*, that means $X = WDW^T = \sum_{i=1}^n \lambda_i v_i v_i^T$ for a diagonal matrix D and a orthogonal matrix W where $v_1, \dots, v_n \in \mathbb{R}^n$ are the orthonormal Eigenvectors and $\lambda_1, \dots, \lambda_n \in \mathbb{R}$ are the Eigenvalues. Then we can verify the equivalences:
a) $\Rightarrow$ *b)*. For each i one has $0 \leq v_i^T X v_i = \lambda_i \|v_i\|_2^2 = \lambda_i$.
b) $\Rightarrow$ *c)*. Setting $U := W\sqrt{D}$ gives $X = WDW^T = UU^T$.
c) $\Leftrightarrow$ *d)*. Choose u_i as ith row of U.
c) $\Rightarrow$ *a)*. For any $a \in \mathbb{R}^n$, one has $a^T X a = \|Ua\|_2^2 \geq 0$. □

Definition 10.4. The *cone of PSD matrices* is

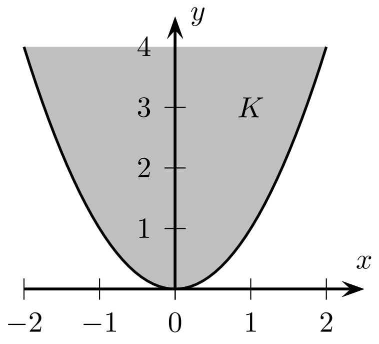
$$\begin{aligned} \mathbb{S}^n_{\geq 0} &:= \{X \in \mathbb{R}^{n\times n} \mid X \text{ symmetric}, X \succeq 0\} \\ &= \{X \in \mathbb{R}^{n\times n} \mid X \text{ symmetric}, \langle X, aa^T \rangle \geq 0 \; \forall a \in \mathbb{R}^n\} \end{aligned}$$

From the 2nd description we can immediately derive:

Fact 10.5. *$\mathbb{S}^n_{\geq 0}$ is convex.*

Note that $\mathbb{S}^n_{\geq 0}$ is *not* a polyhedron.

$$K := \left\{(x, y) \mid \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0\right\}$$



10.2 Semidefinite programs

A *semidefinite program* is of the form:

$$\begin{aligned} \max \langle C, X \rangle & & \\ \langle A_k, X \rangle &\leq & b_k \quad \forall k = 1, \dots, m \\ X & & \text{symmetric} \\ X &\succeq & 0 \end{aligned}$$

where $C, A_1, \dots, A_m \in \mathbb{R}^{n\times n}$. In fact, the set of solutions to an SDP is convex by Fact 10.5. In contrast to LPs, SDPs are less well behaved:

- *Issue 1:* There is a dual to any SDP, but strong duality can fail.
- *Issue 2:* It is possible that all solutions of an SDP are irrational even if the input C, A_k, b_k is integer.
- *Issue 3:* It it possible that exact solutions to an SDP have exponential encoding length even if the input C, A_k, b_k is integer.

However, one can in fact solve SDPs up to small inaccuraries which typically is enough for applications (in particular in the design of approximation algorithms). For the sake of completeness we state this but ignore any such inaccuracies later on.

Theorem 10.6. *Given rational input $A_1, \ldots, A_m, b_1, \ldots, b_m, C, R$ and $\varepsilon > 0$, suppose*

$$SDP = \max\{\langle C, X\rangle \mid \langle A_k, X\rangle \leq b_k \ \forall k;\ X \text{symmetric};\ X \succeq 0\}$$

is feasible and all feasible points are contained in $B(\mathbf{0}, R)$. Then one can find an X^* with

$$\langle A_k, X^*\rangle \leq b_k + \varepsilon \ \ \forall k, \quad X^* \text{ symmetric}, \quad X^* \succeq 0$$

such that $\langle C, X^*\rangle \geq SDP - \varepsilon$. The running time is polynomial in the input length, $\log(R)$ and $\log(1/\varepsilon)$ (in the Turing machine model).

Recall that $Y \succeq 0$ is equivalent to $Y_{ij} = \langle v_i, v_j\rangle$ for some vectors v_i. Making that substitution in an SDP results in an equivalent program that is called *vector program.*

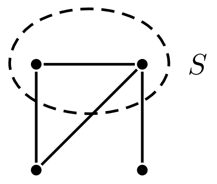
SDP:	**Vector program:**
$\max \sum_{i,j} C_{ij} Y_{ij}$	$\max \sum_{i,j} C_{ij} \langle v_i, v_j\rangle$
$\sum_{i,j} A^k_{ij} \cdot Y_{ij} \leq b_k \quad \forall k$	$\sum_{i,j} A^k_{ij} \cdot \langle v_i, v_j\rangle \leq b_k \quad \forall k$
Y sym.	$v_i \in \mathbb{R}^r \quad \forall i$
$Y \succeq 0$	

Curiously, the right hand side vector program is *not* convex, but rather is equivalent to a convex program (the SDP) and hence can be solved in polynomial time. Also note that one cannot choose the dimension r of the vectors. In fact, solving a vector program subject to the rank restriction $r = 1$ is again **NP**-hard.

10.3 MaxCut

In this chapter, we discuss one of the most prominent problems in optimization and theoretical computer science, which is the MaxCut problem and the algorithm of Goemans and Williamson [GW95].

MaxCut
Input: An undirected graph $G = (V, E)$ and weights $w : E \to \mathbb{R}_{\geq 0}$
Goal: Find the cut $S \subseteq V$ that maximizes the weight $w(\delta(S))$ of the cut edges.



It is **NP**-hard to find a solution that cuts even 94% of what the optimum cuts [Hås97]. This is might be surprising as the *minimum* cut problem is solvable in polynomial time. In terms of obvious approximation algorithms note that a simple greedy algorithm can already find a solution that cuts at least $|E|/2$ edges in the unit weight setting or $\frac{1}{2}w(E)$ in the setting with non-negative weights.

We consider the following semidefinite program with equivalent vector program:

SDP:

$$\begin{array}{rcl} \max & & \frac{1}{2}\sum_{\{i,j\}\in E} w_{ij}\cdot(1 - X_{ij}) \\ X & \succeq & 0 \\ X_{ii} & = & 1 \quad \forall i \in V \\ X & \in & \mathbb{R}^{n\times n} \end{array}$$

Vector program:

$$\begin{array}{rcl} \max & & \frac{1}{2}\sum_{\{i,j\}\in E} w_{ij}\cdot(1 - \langle u_i, u_j\rangle) \\ \|u_i\|_2 & = & 1 \quad \forall i \in V \\ u_i & \in & \mathbb{R}^r \end{array}$$

We verify that these programs indeed provide a relaxation.

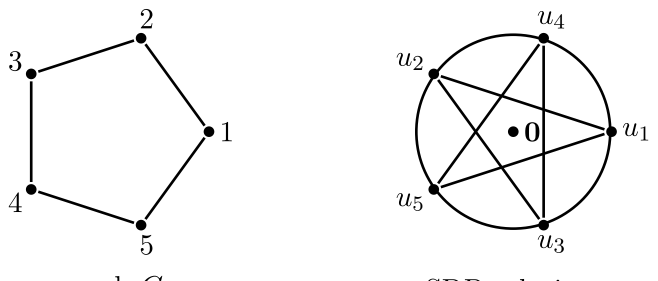
Lemma 10.7. *If $S^* \subseteq V$ is optimum solution for MaxCut, then $SDP \geq w(\delta(S^*))$.*

Proof. We choose vectors in dimension $r := 1$ and define $u_i \in \mathbb{R}^1$ by

$$u_i := \begin{cases} 1 & \text{if } i \in S^* \\ -1 & \text{if } i \in V \setminus S^* \end{cases}$$

□

Example 10.8. Consider the graph G which is a cycle on 5 nodes and unit weights $w(e) = 1$. The optimum MaxCut is 4. On the other hand, choosing $u_i \in \mathbb{R}^2$ with $u_i := (\cos(\frac{4i\pi}{5}), \sin(\frac{4i\pi}{5}))$ we get a vector program solution of value $5\cdot\frac{1}{2}(1-\cos(\frac{4}{5}\pi)) \approx 4.522$.



graph G SDP solution:

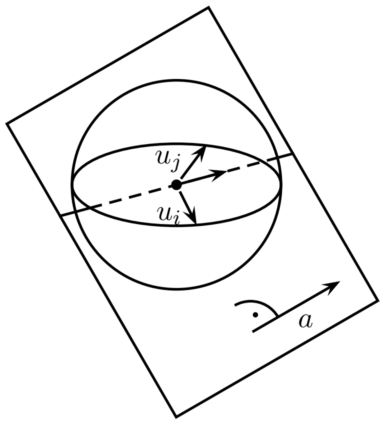
We need an algorithm that can transform a MaxCut SDP solution into an actual cut $\delta(S)$ while being close in terms of the objective function value. The algorithm is quite simple: randomly partition $\mathbb{R}^r$ into two halfspaces; then let S be the vertices whose vectors u_i lie in one of those halfspaces.

Hyperplane Rounding algorithm

Input: Graph $G = (V, E)$ and edge weights $w : E \to \mathbb{R}_{\geq 0}$
Output: Set $S \subseteq V$

```
(1) Solve the MaxCut SDP
(2) Sample a random standard Gaussian $a \in \mathbb{R}^r$
(3) Define $S := \{i \in V \mid \langle a, u_i \rangle \geq 0\}$
```

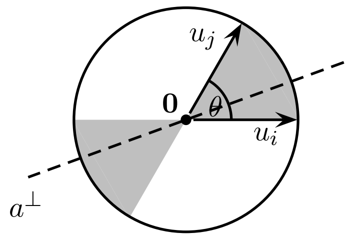
Note that the algorithm that we will analyze is *randomized* and we will prove that the *expected* value of the produced cut is good.



Lemma 10.9. *For* $\{i, j\} \in E$ *one has* $\Pr[\{i, j\} \in \delta(S)] = \frac{1}{\pi}\arccos(\langle u_i, u_j \rangle)$.

Proof. First note that the angle between vectors is exactly $\theta := \arccos(\langle u_i, u_j \rangle)$ (as $\langle u_i, u_j \rangle = \cos(\theta)$). Next, note that considering only the edge $\{i, j\}$ the random experiment is equivalent to drawing a random unit vector a in the 2-dimensional[1] space $U := \text{span}\{u_i, u_j\}$. Then we can verify that indeed $\Pr[u_i, u_j \text{ separated}] = \frac{2\theta}{2\pi}$.

[1]The case that $u_i = \pm u_j$ is obvious.

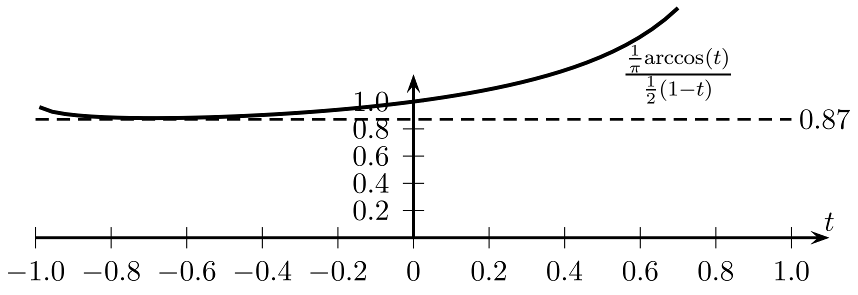


□

Theorem 10.10. *One has* $\mathbb{E}[w(\delta(S))] \geq 0.878 \cdot SDP$.

Proof. Fix an edge $e = \{i,j\} \in E$ and abbreviate $t := \langle u_i, u_j \rangle$. Then the contribution of that edge to the SDP objective function is $w_{ij} \cdot \frac{1}{2}(1-t)$. On the over hand, the expected contribution of e to the cut value is $w_{ij} \cdot \Pr[\{i,j\} \in \delta(S)] = w_{ij} \cdot \frac{1}{\pi}\arccos(t)$. We can numerically verify that the ratio is indeed

$$\frac{\frac{1}{\pi}\arccos(t)}{\frac{1}{2}(1-t)} \geq 0.878 \quad \forall t \in [-1,1]$$



The claim then follows by *linearity of expectation.* □

10.4 Grothendieck's Inequality

In this section, we discuss the integrality gap of a certain quadratic program. For a matrix $A \in \mathbb{R}^{m \times n}$ define

$$\begin{aligned} INT(A) &:= \max\Big\{ \sum_{i=1}^m \sum_{j=1}^n A_{ij} x_i y_j \mid x \in \{-1,1\}^m, y \in \{-1,1\}^n \Big\} \\ SDP(A) &:= \max\Big\{ \sum_{i=1}^m \sum_{j=1}^n A_{ij} \langle u_i, v_j \rangle \mid \|u_i\|_2 = \|v_j\|_2 = 1 \Big\} \end{aligned}$$

Note that $SDP(A)$ is indeed a vector program that can be solved in polynomial time. We will show the following result:

Theorem 10.11 (Grothendieck's Inequality). *For any matrix $A \in \mathbb{R}^{m\times n}$ one has*

$$INT(A) \leq SDP(A) \leq C_G \cdot INT(A)$$

where $C_G \leq 1.783$.

Note that the lower bound is clear. Grothendieck [Gro53] was the first to prove that there is such a universal constant C_G and later Krivine [Kri79] provided the bound if $C_G \leq 1.783$. Here we will reproduce Krivine's argument, following in part the exposition of Vershynin [Ver18].

Given a solution to the SDP in form of some vectors $u_i, v_j \in \mathbb{R}^n$, our goal will be to produce integers $x_i, y_j \in \{-1,1\}$. We want to follow the recipe that worked so smoothly in case of MaxCut. For $z \in \mathbb{R}$ we define

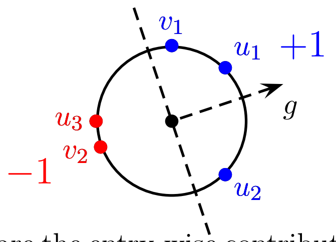
$$\text{sign}(z) := \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{if } z < 0 \end{cases}$$

Then we consider the following *random experiment:*

(1) Given vectors $u_i, v_j \in \mathbb{R}^r$.

(2) Sample a Gaussian g in $\mathbb{R}^r$ and set

$$x_i := \text{sign}(\langle u_i, g\rangle) \quad \text{and} \quad y_j := \text{sign}(\langle v_j, g\rangle)$$



If we again want to compare the entry-wise contributions, then we need to analyze how $\mathbb{E}[A_{ij}x_iy_j]$ relates to $A_{ij}\langle u_i, v_j\rangle$. That argument is very similar to the case of maxcut.

Lemma 10.12. *Let $u, v \in \mathbb{R}^r$ with $\|u\|_2 = \|v\|_2 = 1$. Then*

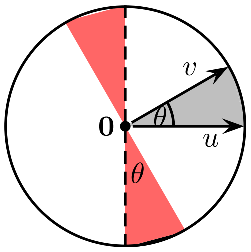
$$\mathop{\mathbb{E}}_{g \text{ Gaussian}}\big[\textit{sign}(\langle g,u\rangle)\cdot \textit{sign}(\langle g,v\rangle)\big] = \frac{2}{\pi}\textit{arcsin}(\langle u,v\rangle)$$

Proof. Let $\theta \in [0,\pi)$ be the angle between u and v, i.e. $\cos(\theta) = \langle u,v\rangle$. Then

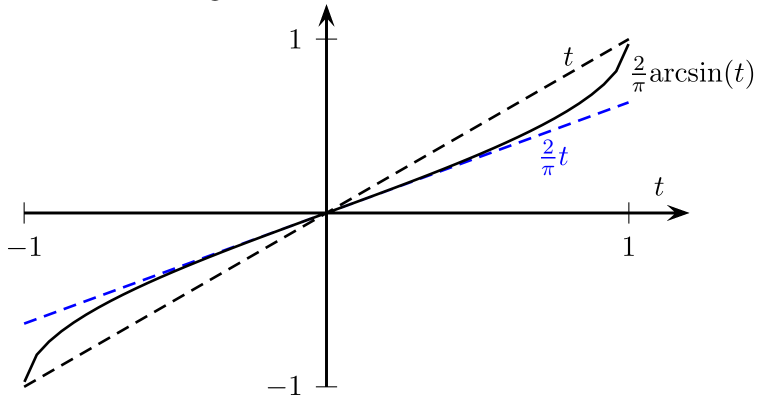
$$\begin{aligned}\mathop{\mathbb{E}}_{g \text{ Gaussian}}\big[\text{sign}(\langle g,u\rangle)\cdot \text{sign}(\langle g,v\rangle)\big] &= 1 - \Pr[u, v \text{ separated}] \\ &= 1 - \frac{2\theta}{\pi} \\ &= 1 - \frac{2}{\pi}\arccos(\langle u,v\rangle) = \frac{2}{\pi}\arcsin(\langle u,v\rangle)\end{aligned}$$

Here we use that $\arccos(t) = \frac{\pi}{2} - \arcsin(t)$. $\square$

To visualize the proof, note that the vectors u, v will be separated if the projection of g into $\mathrm{span}(\{u, v\})$ falls into one of the two red-shaded angles of value θ each.



We can visualize the resulting curve:



For example, for $t \geq 0$, one has $\frac{2}{\pi}t \leq \frac{2}{\pi}\arcsin(t) \leq t$. In fact, one can see that

- For $A_{ij} \geq 0$ and $\langle u_i, u_j \rangle \geq 0$ one has $\mathbb{E}[A_{ij}x_iy_j] \geq \frac{2}{\pi} \cdot A_{ij} \langle u_i, v_j \rangle$
- For $A_{ij} < 0$ and $\langle u_i, u_j \rangle \geq 0$ one has $\mathbb{E}[A_{ij}x_iy_j] \geq A_{ij} \langle u_i, v_j \rangle$

One might be tempted to claim that a straightforward hyperplane rounding argument shows that $INT(A) \geq \frac{2}{\pi}SDP(A)$. But that is not true! Suppose we have only two non-zero entries, one with value $A_{11} = \frac{1}{\varepsilon}$ and inner product $\langle u_1, v_1 \rangle = \varepsilon$ and the other one with $A_{22} = -1$ and $\langle u_2, v_2 \rangle = 1$. Then the SDP objective is 0. On the other hand, the expected value of a hyperplane solution is $-1 + \frac{1}{\varepsilon} \cdot \frac{2}{\pi}\arcsin(\varepsilon) \approx -1 + \frac{2}{\pi} \approx -0.36$. The issue here is the non-linearity of $t \mapsto \frac{2}{\pi}\arcsin(t)$! The fix will be to force the rounding to give a linear bound.

10.4.1 Tensors

We introduce some more linear algebra.

Definition 10.13. A *kth order tensor* $A \in \mathbb{R}^{n_1 \times \ldots \times n_k}$ is a k-dimensional array of numbers; we write $A = (A_{i_1,\ldots,i_k})_{i_1 \in [n_1],\ldots,i_k \in [n_k]}$.

For example a vector $a \in \mathbb{R}^n$ is a 1st order tensor and a matrix $A \in \mathbb{R}^{n_1 \times n_2}$ is a 2nd order tensor.

Definition 10.14. For two tensors $A, B \in \mathbb{R}^{n_1 \times \ldots \times n_k}$ we define an *inner product*

$$\langle A, B \rangle := \sum_{i_1,\ldots,i_k} A_{i_1,\ldots,i_k} \cdot B_{i_1,\ldots,i_k}$$

Again, for vectors this corresponds to the standard inner product.

Definition 10.15. For vector $u \in \mathbb{R}^n$ and $k \in \mathbb{N}$, define the *tensor product*

$$u \otimes \ldots \otimes u := u^{\otimes k} := (u_{i_1} \cdot \ldots \cdot u_{i_k})_{i_1 \in [n],\ldots,i_k \in [n]}$$

The following is a useful fact:

Fact 10.16. *For vectors $u, v \in \mathbb{R}^n$ one has $\left\langle u^{\otimes k}, v^{\otimes k} \right\rangle = \langle u, v \rangle^k$.*

Definition 10.17. We call a function $f : \mathbb{R} \to \mathbb{R}$ *(real) analytic* if it can be written as a convergent power series $f(x) = \sum_{k=0}^{\infty} a_k x^k$ for all $x \in \mathbb{R}$.

Given a fixed $r \in \mathbb{N}$, we can define a *Hilbert space / infinite-dimensional vector space* of the form

$$H = \{(U^0, U^1, U^2, U^3, \ldots) \mid U^k \text{ is a } k\text{-tensor of size } r^k\}$$

using the natural inner product.

Now we can "bend" any vectors to give any analytic function that we like:

Lemma 10.18. *Let $f(x) = \sum_{k=0}^{\infty} a_k x^k$ and fix a dimension $r \in \mathbb{N}$. Then there is a Hilbert space H and maps $F, G : \mathbb{R}^r \to H$ so that*

$$\langle F(u), G(v) \rangle = f(\langle u, v \rangle) \quad \forall u, v \in \mathbb{R}^r$$

Moreover the length of the mapped vectors satisfies

$$\|F(u)\|_2^2 = \|G(u)\|_2^2 = \sum_{k=0}^{\infty} |a_k| \cdot \|u\|_2^{2k}$$

Proof. We use the maps

$$F(u) := (\sqrt{|a_k|} \cdot u^{\otimes k})_{k \in \mathbb{Z}_{\geq 0}}, \quad G(u) := (\text{sign}(a_k) \cdot \sqrt{|a_k|} \cdot u^{\otimes k})_{k \in \mathbb{Z}_{\geq 0}}$$

Then for vectors $u, v \in \mathbb{R}^r$ one has

$$\begin{aligned} \langle F(u), G(v) \rangle &= \sum_{k \geq 0} \text{sign}(a_k) \cdot (\sqrt{|a_k|})^2 \cdot \left\langle u^{\otimes k}, v^{\otimes k} \right\rangle \\ &= \sum_{k \geq 0} a_k \cdot \langle u, v \rangle^k = f(\langle u, v \rangle). \end{aligned}$$

We can verify that the lengths of the transformed vectors are

$$\|F(u)\|_2^2 = \|G(u)\|_2^2 = \sum_{k \geq 0} (\sqrt{|a_k|})^2 \cdot \|u^{\otimes k}\|_2^2 = \sum_{k \geq 0} |a_k| \cdot \|u\|_2^{2k}$$

as claimed. $\square$

10.4.2 Proof of Grothendieck's Inequality

Now we apply Lemma 10.18 to linearize the function appearing in the hyperplane rounding. More concretely, we will find maps $F, G : \mathbb{R}^r \to H$ so that

$$\frac{2}{\pi}\arcsin(\langle F(u), G(v)\rangle) = \beta \cdot \langle u, v\rangle$$

for some constant $\beta > 0$.

Lemma 10.19. *Let $r \in \mathbb{N}$. Then there are maps $F, G : \mathbb{R}^r \to H$ so that*

$$\langle F(u), G(v)\rangle = \sin\Big(\beta\frac{\pi}{2}\langle u, v\rangle\Big)$$

where $\beta = \frac{2}{\pi}\ln(1+\sqrt{2}) \approx \frac{1}{1.783}$. Moreover $\|F(u)\|_2^2 = \|G(u)\|_2^2 = 1$ for all $u \in \mathbb{R}^r$ with $\|u\|_2^2 = 1$.

Proof. First, recall that

$$\begin{aligned}\sin(x) &= \sum_{k\geq 0}\frac{(-1)^k}{(2k+1)!}x^{2k+1} = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \dots \\ \sinh(x) &= \sum_{k\geq 0}\frac{1}{(2k+1)!}x^{2k+1}\end{aligned}$$

We apply Lemma 10.18 to the analytic function $f(x) = \sin(\beta\frac{\pi}{2}x)$ where we decide $\beta \in \mathbb{R}$ later. Then for $\|u\|_2 = 1$, we can see that the length of the transformed vector is

$$\|F(u)\|_2^2 = \sum_{k\geq 0}\Big|\frac{(-1)^k}{(2k+1)!}\cdot\Big(\beta\frac{\pi}{2}\Big)^{2k+1}\Big| = \sinh\Big(\beta\frac{\pi}{2}\Big) \overset{\beta:=\frac{2}{\pi}\text{arcsinh}(1)}{=} 1$$

Rearranging we see that

$$\beta = \frac{2}{\pi}\text{arcsinh}(1) = \frac{2}{\pi}\ln(1+\sqrt{2}) \approx \frac{1}{1.783}.$$

□

Now we can complete the proof of Grothendieck's Inequality using Krivine's rounding argument.

Theorem 10.11. Consider a matrix $A \in \mathbb{R}^{m\times n}$ and let $u_i, v_j \in \mathbb{R}^r$ be vectors with $\|u_i\|_2 = 1 = \|v_j\|_2$ attaining $SDP(A)$. Construct the maps $F, G : \mathbb{R}^r \to H$ from Lemma 10.19. Sample a Gaussian g in H^2 and set

$$x_i := \text{sign}(\langle g, F(u_i)\rangle)\ \forall i \in [m] \quad\text{and}\quad y_j := \text{sign}(\langle g, G(v_j)\rangle)\ \forall j \in [n]$$

[2] In order to avoid the discussion of what a Gaussian in an infinite dimension space should be, note that only the subspace $U := \text{span}(\{F(u_i) : i = 1,\dots,m\} \cup \{G(v_j) : j = 1,\dots,n\})$ of H with dimension $\dim(U) \leq m+n$ is relevant which is isomorphic to $\mathbb{R}^{m+n}$. So indeed it suffices to draw a Gaussian from that subspace U.

Then

$$\mathbb{E}[x_i y_j] \overset{\text{Lem 10.12}}{=} \frac{2}{\pi}\arcsin(\langle F(u_i), G(v_i)\rangle) = \beta \cdot \langle u_i, v_i\rangle$$

By linearity of expectation we obtain

$$\mathbb{E}\Big[\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}x_i y_j\Big] = \beta \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}\,\langle u_i, v_j\rangle = \underbrace{\beta}_{\approx\frac{1}{1.783}} \cdot SDP(A)$$

$\square$

Chapter 11

Equivalence of Optimization and Separation*

In this chapter, we will discuss the topic of linear optimization over polyhedra from a more abstract point of view. Similar to the interior point method, the running times will depend on the size of the numbers describing the LP. For an integer $z \in \mathbb{Z}$ we define $\langle z \rangle := 1 + \lceil \log_2(|z| + 1) \rceil$ as the *encoding length.* Then for a rational number $\alpha \in \mathbb{Q}$ with $\alpha = \frac{p}{q}$ with $p \in \mathbb{Z}$ and $q \in \mathbb{N}$ coprime we set $\langle \alpha \rangle = \langle p \rangle + \langle q \rangle$. Similarly, for a vector $c \in \mathbb{Q}^n$ and a matrix $A \in \mathbb{Q}^{m \times n}$ we denote $\langle c \rangle$ and $\langle A \rangle$ as the sum of the encoding length of all entries. Note that $\langle c \rangle \geq n$ and $\langle A \rangle \geq nm$.

11.1 The Ellipsoid Method

The *Ellipsoid method* was developed by Shor, Yudin and Nemirovski in the 1970s for nonlinear optimization; later it was applied by Khachyian [Kha79] for solving LP's in polynomial time. It was actually the first method proven to be able to solve LPs in polynomial time. While not being used in practice, the ellipsoid method has an important function in the theory of linear programs that we elaborate here. In this section, we follow the exposition of Korte and Vygen [KV12]. We will point out some of the numerical issues without going into the (usually tedious) details. We refer to [KV12] for more background.

There are several equivalent ways to define an ellipsoid. We choose the following:

Definition 11.1. Let $B \in \mathbb{R}^{n \times n}$ be a matrix with $B \succ 0$ and let $c \in \mathbb{R}^n$. Then

$$E(B, c) := \left\{ x \in \mathbb{R}^n \mid (x - c)^T B^{-1} (x - c) \leq 1 \right\}$$

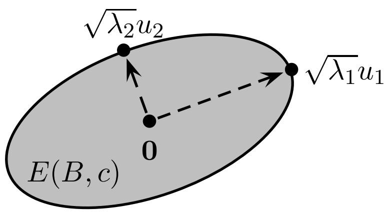
is called an *ellipsoid.*

We write $B(c, r) := E(I_n, c)$ as the *Euclidean ball* of radius r and center c. Consider the *Eigen decomposition* of B in the form $B = \sum_{i=1}^n \lambda_i u_i u_i^T$ where $\lambda_1, \ldots, \lambda_n >$

0 are the Eigenvalues and $u_1, \ldots, u_n$ are the orthonormal Eigenvectors. Then the ellipsoid can be alternatively written as

$$E(B,c) = \Big\{ x \in \mathbb{R}^n \mid \sum_{i=1}^{n} \frac{\langle x - c, u_i \rangle^2}{\lambda_i} \leq 1 \Big\}$$

In particular c is the *center* of the ellipsoid and u_i is the *ith axis* that has a length of $\sqrt{\lambda_i}$.



From this representation we can see that $E(B,c)$ is the image of the linear map $B^{1/2}$ shifted by c, i.e. $E(B,c) = \{B^{1/2}y + c \mid y \in B(\mathbf{0},1)\}$. We can also see that the volume of this ellipsoid is

$$\frac{\mathrm{Vol}_n(E(B,c))}{\mathrm{Vol}_n(B(\mathbf{0},1))} = \prod_{i=1}^{n} \sqrt{\lambda_i} = \det(B^{1/2}) = \sqrt{\det(B)}$$

The idea behind the ellipsoid method is quite simple: say we have a target polytope $P \subseteq \mathbb{R}^n$ and we know that it is contained in some large ellipsoid $E_0 = E(A_0, c_0)$. Then we check whether $c_0 \in P$ — if yes, we are done. Otherwise, by taking an inequality $a^T x \leq b$ of P that is violated by c_0 we learn that P is contained in a half-ellipsoid $E_0 \cap H$ with $H := \{x \in \mathbb{R}^n \mid a^T x \leq a^T c_0\}$. Then we can explicitly compute an ellipsoid E_1 that contains $E_0 \cap H$ but has a smaller volumne that E_0. Then we iterate.

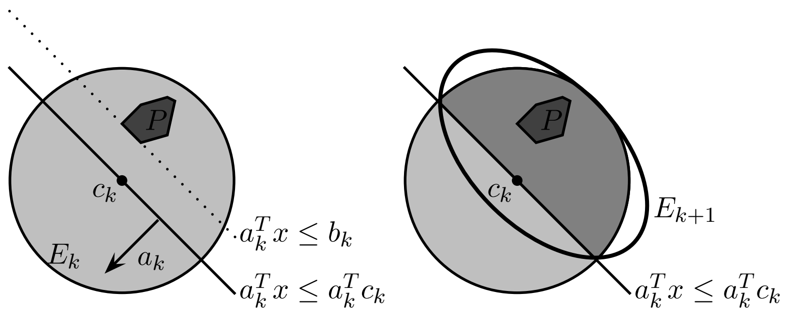
Ellipsoid method

Input: $A \in \mathbb{Q}^{m\times n}, b \in \mathbb{Q}^m$ and $R > 0$ so that $P := \{x \in \mathbb{R}^n \mid Ax \leq b\}$ is contained in $B(\mathbf{0}, R)$.

Output: Point $x \in P$.

```
(1) Set $B_0 := R^2 I_n$, $c_0 := \mathbf{0}$ so that $E_0 := E(B_0, c_0) = B(\mathbf{0}, R)$.
(2) FOR $k = 0$ TO $\infty$
        (3) If $c_k \in P$ THEN return $c_k$
        (4) Select an inequality $a_k^T x \leq \beta_k$ from system $Ax \leq b$ with $a_k^T c_k > \beta_k$
        (5) Set
                $$d_k := \frac{1}{\sqrt{a_k^T B_k a_k}} B_k a_k$$
            and update
                $$c_{k+1} := c_k - \frac{1}{n+1} d_k, \quad B_{k+1} := \frac{n^2}{n^2-1}\Big(B_k - \frac{2}{n+1} d_k d_k^T\Big)$$
            and $E_{k+1} := E(B_{k+1}, c_{k+1})$.
```

We note that the new ellipsoid E_{k+1} is obtained by moving the center of E_k into direction $-a_k$, squeezing E_k in direction a_k and slightly expanding it in directions orthogonal to a_k.



We will prove correctness of the ellipsoid method in form of the following theorem:

Theorem 11.2. *Suppose $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ is contained in $B(\mathbf{0}, R)$ and P contains some ball of radius $r > 0$. Then the ellipsoid method finds a point in P in at most $O(n^2 \log \frac{R}{r})$ many iterations.*

It suffices to prove the following two properties:

(i) For all $k \geq 0$ one has $E_{k+1} \supseteq E_k \cap \{x \in \mathbb{R}^n \mid a_k^T x \leq a_k^T c_k\}$

(ii) For all $k \geq 0$ one has $\frac{\mathrm{Vol}_n(E_{k+1})}{\mathrm{Vol}_n(E_k)} \leq \exp(-\frac{1}{2n+2})$.

By (i) we have $P \subseteq E_k$ for all k. Moreover, as P contains a radius r-ball, we know that in each iteration k we have

$$\begin{aligned} r^n \mathrm{Vol}_n(B(\mathbf{0},1)) &= \mathrm{Vol}_n(B(\mathbf{0},r)) \\ &\leq \mathrm{Vol}_n(E_k) \\ &\overset{(ii)}{\leq} \exp\Big(-\frac{k}{2n+2}\Big) \cdot \mathrm{Vol}_n(E_0) \\ &= \exp\Big(-\frac{k}{2n+2}\Big) \cdot R^n \mathrm{Vol}_n(B(\mathbf{0},1)) \end{aligned}$$

Rearranging gives

$$k \leq n \cdot (2n+2) \ln\Big(\frac{R}{r}\Big)$$

which establishes the upper bound on the number of iterations in Theorem 11.2.

It remains to prove (i) and (ii). We fix some iteration k. Note that applying an affine linear transformation does not change the properties (i) and (ii). If we transform the space so that $E_k = B(\mathbf{0},1)$ (i.e. $B_k = I_n$ and $c_k = \mathbf{0}$) and $\|a_k\|_2 = 1$ then the update formula simplifies to

$$c_{k+1} = -\frac{1}{n+1} a_k \quad \text{and} \quad B_{k+1} = \frac{n^2}{n^2-1}\Big(I_n - \frac{2}{n+1} a_k a_k^T\Big)$$

This means that in direction a_k we shrink the axis length to $\sqrt{\frac{n^2}{n^2-1}(1-\frac{2}{n+1})} = 1 - \frac{1}{n} \pm O(\frac{1}{n^2})$ while in directions orthogonal to a_k we expand the axis length to $\sqrt{\frac{n^2}{n^2-1}} = 1 + \frac{1}{2n^2} \pm O(\frac{1}{n^4})$. Note that one can prove that the update formula for the ellipsoid are chosen so that E_{k+1} is precisely the ellipsoid satisfying (i) that minimizes the volume. That ellipsoid is also called the *Löwner-John ellipsoid* with respect to the convex body $E_k \cap \{x \in \mathbb{R}^n \mid a_k^T x \leq a_k^T c_k\}$. However, in order to have a cleaner proof we will instead analyze a different ellipsoid that is slightly larger along the a_k direction so that proving (i) is easier. Without loss of generality we may also assume that $a_k = -e_1$, hence it suffices to proof the following standalone lemma:

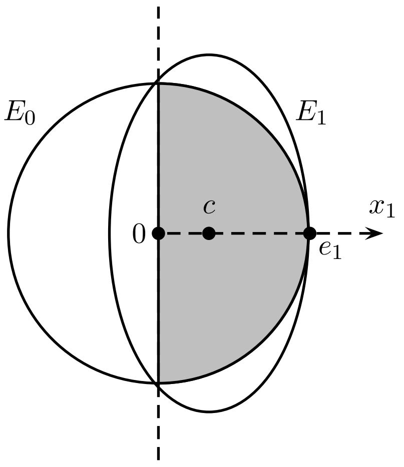
Lemma 11.3. *For $n \geq 2$, define $B \in \mathbb{R}^{n\times n}$*

$$B = diag\Big(\Big(\frac{n}{n+1}\Big)^2, \underbrace{\frac{n^2}{n^2-1}, \ldots, \frac{n^2}{n^2-1}}_{n-1 \text{ entries}}\Big) \quad \text{and} \quad c := \frac{1}{n+1} e_1$$

Then $E_0 := B(\mathbf{0},1)$ and $E_1 := E(B,c)$ satisfy

(i) One has $E_1 \supseteq E_0 \cap \{x \in \mathbb{R}^n \mid x_1 \geq 0\}$

(ii) One has $\frac{Vol_n(E_1)}{Vol_n(E_0)} \leq \exp(-\frac{1}{2n+2})$.



Proof. First note that the ellipsoid E_1 is of the form

$$\begin{aligned} E_1 &= \{x \in \mathbb{R}^n \mid (x-c)^T B^{-1}(x-c) \leq 1\} \\ &= \Big\{ x \in \mathbb{R}^n \mid \Big(\frac{n+1}{n}\Big)^2 \Big(x_1 - \frac{1}{n+1}\Big)^2 + \frac{n^2-1}{n^2}\sum_{i=2}^n x_i^2 \leq 1 \Big\} \end{aligned}$$

To show (i), we prove that for $x \in \mathbb{R}^n$ with $\|x\|_2 \leq 1$ and $x_1 \geq 0$ one has $x \in E_1$. And indeed

$$\begin{aligned} &\Big(\frac{n+1}{n}\Big)^2 \Big(x_1 - \frac{1}{n+1}\Big)^2 + \frac{n^2-1}{n^2}\underbrace{\sum_{i=2}^n x_i^2}_{\leq 1 - x_1^2} \\ \leq\ & \Big(\Big(\frac{n+1}{n}\Big)^2 - \frac{n^2-1}{n^2}\Big)x_1^2 - 2\frac{1}{n+1}\Big(\frac{n+1}{n}\Big)^2 x_1 + \Big(\Big(\frac{n+1}{n}\Big)^2 \frac{1}{(n+1)^2} + \frac{n^2-1}{n^2}\Big) \\ =\ & \frac{2n+2}{n^2} \cdot \underbrace{(x_1^2 - x_1)}_{\leq 0} + 1 \leq 1 \end{aligned}$$

using that $0 \leq x_1 \leq 1$. To verify claim (ii) we estimate that

$$\begin{aligned} \frac{\mathrm{Vol}_n(E_1)}{\mathrm{Vol}_n(E_0)} &= \sqrt{\det(B)} = \prod_{i=1}^n \sqrt{B_{ii}} \\ &= \frac{n}{n+1} \cdot \Big(\frac{n^2}{n^2-1}\Big)^{(n-1)/2} = \Big(1 - \frac{1}{n+1}\Big) \cdot \Big(1 + \frac{1}{n^2-1}\Big)^{(n-1)/2} \\ &\leq \exp\Big(-\frac{1}{n+1}\Big) \cdot \exp\Big(\frac{1}{n^2-1} \cdot \frac{n-1}{2}\Big) = \exp\Big(-\frac{1}{2(n+1)}\Big) \end{aligned}$$

Here we use that B is diagonal and the fact that $1 + y \leq e^y$ for all $y \in \mathbb{R}$. □

11.1.1 Technical issues

If our goal is to solve any LP

$$\max\{c^T x \mid Ax \leq b\} \tag{11.1}$$

in time polynomial in the encoding length $\langle A\rangle$, $\langle b\rangle$ and $\langle c\rangle$ using the ellipsoid method, then there are several technical issues to be resolved. In the following let $P := \{x \in \mathbb{R}^n \mid Ax \leq b\}$.

- *Reduction from optimization to feasibility.* There are two popular reductions.
  (1) We can apply binary search to find the maximal value of $\beta \in \mathbb{R}$ so that $P \cap \{x \in \mathbb{R}^n \mid c^T x \geq \beta\}$ is feasible.
  (2) We can combine the primal and the dual to the system
  $$c^T x = b^T y, \quad Ax \leq b, \quad A^T y = c, \quad y \geq \mathbf{0} \tag{11.2}$$
  Then any feasible solution to (11.2) must be an optimum solution to the original LP in (11.1).
- *Boundedness.* The ellipsoid method is only defined for bounded polyhedra P. But one can choose a large enough value $M := 2^{\text{poly}(\langle A\rangle,\langle b\rangle)}$ so that $P \neq \emptyset \Leftrightarrow P \cap [-M, M]^n \neq \emptyset$.
- *Full-dimensionality.* It may be that the polytope P is not full dimensional, and so there is no positive radius ball contained in P in which case the ellipsoid method might never terminate. Again, there are two strategies to solve this issue:
  (1) For $\varepsilon > 0$, let $P_\varepsilon := \{x \in \mathbb{R}^n \mid Ax \leq b + \varepsilon\mathbf{1}\}$. One can pick a value $\varepsilon := 2^{-\text{poly}(\langle A\rangle,\langle b\rangle)}$ so that $P \neq \emptyset \Leftrightarrow P_\varepsilon \neq \emptyset$. In particular, for such a choice of ε one can prove that a basis is feasible for P if and only if it is feasible for P_ε.
  (2) There is a threshold $\delta := 2^{-\text{poly}(\langle A\rangle,\langle b\rangle)}$ so that if $\text{Vol}_n(E_k) \leq \delta$, then one can infer a hyperplane H that must contain all rational vectors in E_k that have encoding length at most $\text{poly}(\langle A\rangle, \langle b\rangle)$. In particular, $P \subseteq H$. Then one can recurse. We cover this non-trivial argument later in Chapter 12.
- *Rationality.* The update formula
  $$c_{k+1} := c_k - \frac{1}{n+1}\frac{1}{\sqrt{a_k^T B_k a_k}} B_k a_k$$
  for the center contains a square root and in particular it is not true that the numbers at every step of the ellipsoid method are rationals with polynomially bounded encoding length. But one can choose the updated ellipsoid slightly larger and then truncate all numbers after $\text{poly}(\langle A\rangle, \langle b\rangle)$ many bits.

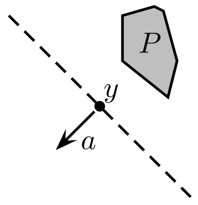
We will not elaborate on these details and again refer to the book of Korte and Vygen [KV12]. We summarize the exact statement that can be derived from the ellipsoid method (or from the interior point method for that matter):

Theorem 11.4. *Let $A \in \mathbb{Q}^{m\times n}$, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^n$. Then the LP $\max\{c^T x \mid Ax \leq b\}$ can be solved in time polynomial in $\langle A\rangle$, $\langle b\rangle$ and $\langle c\rangle$ (or one can decide that the LP is infeasible or unbounded).*

11.1.2 The ellipsoid method with a separation oracle

Consider again a polytope $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ with $A \in \mathbb{Q}^{m\times n}$ and $b \in \mathbb{Q}^m$ with $B(c^*, r) \subseteq P \subseteq B(\mathbf{0}, R)$ for some unknown point $c^* \in \mathbb{R}^n$ and $r \leq R$. In Theorem 11.2 we have seen that the ellipsoid method takes $O(n^2 \log \frac{R}{r})$ many iterations to find a feasible point. That means the number of iterations does not depend on the number m of inequalities. Instead of maintaining an explicit list of all m inequalities in the system $Ax \leq b$, it would suffice if in every iteration, the algorithm could determine *one* inequality violated by the current center c_k. This is called the *separation problem* for P.

SEPARATION PROBLEM FOR POLYHEDRON $P \subseteq \mathbb{R}^n$
Input: Point $y \in \mathbb{Q}^n$
Goal: Find a $a \in \mathbb{Q}^n$ with $a^T y > a^T x \ \forall x \in P$ or assert $y \in P$.



Then running the ellipsoid method with a separation oracle and solving again all the technical issues that we mentioned earlier gives the following:

Theorem 11.5. *Let $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ be a polyhedron with $A \in \mathbb{Q}^{m\times n}, b \in \mathbb{Q}^m$, and let $c \in \mathbb{Q}^n$ be an objective function and let $\varphi \geq \max_{i=1,\dots,m}\{\langle A_i\rangle + \langle b_i\rangle\} + \langle c\rangle$. Suppose one can solve the following problem in time poly(φ):*

> SEPARATION PROBLEM. *Given $y \in \mathbb{Q}^n$ with encoding length poly(φ) as input. Decide, whether $y \in P$. If not find an $a \in \mathbb{Q}^n$ with $a^T y > a^T x \ \forall x \in P$.*

Then there is an algorithm that on input of (c, φ) yields in time poly(φ) either

- *$x^* \in \mathbb{Q}^n$ attaining $\max\{c^T x \mid x \in P\}$ (x^* will be a vertex if P has vertices)*
- *P empty*

- *Vectors $x, y \in \mathbb{Q}^n$ with $x + \lambda y \in P \; \forall \lambda \geq 0$ and $c^T y \geq 1$.*

Note that the algorithm would need to know φ in order to construct the initial ellipsoid E_0 containing P and in order to decide when to recurse on a lower dimensional subspace.

11.2 Optimization vs. separation

As we have just seen, we can optimize a linear function $c^T x$ over a polyhedron P as long as we can solve the separation problem for P. In this section, we will prove that if we can optimize over P, then we can also solve the separation problem for P. In other words, optimization and separation is equivalent from the point of view of polynomial time algorithms. To some extend we will follow the very readable exposition of Cook, Cunningham, Pulleyblank and Schrijver [CCPS98] and add some details from [KV12].

We want to make this equivalence formal which requires some care. For the sake of simplicity, we restrict our attention to the (more interesting) bounded case. We will also talk about *class* of polytopes rather than a *single* polytope. We have set of *objects* $\mathcal{T}$ and with every object $t \in \mathcal{T}$ we associate a rational polytope $P_t \subseteq \mathbb{R}^{n_t}$. For example each object $t \in \mathcal{T}$ might be a (weighted) graph or a matrix or a flow network etc. For each such object we write $\langle t \rangle$ as the encoding length. We need a technical condition that is usually trivial to verify.

Definition 11.6. A class of polyhedra $\mathcal{P} = \{P_t : t \in \mathcal{T}\}$ with $P_t = \{x \in \mathbb{R}^{n_t} \mid A^{(t)} x \leq b^{(t)}\}$ is called *proper* if (A) each P_t is bounded and $A^{(t)}$ and $b^{(t)}$ are rational; (B) there is a polynomial time algorithm that takes the object $t \in \mathcal{T}$ as input and provides the dimension $n_t \in \mathbb{N}$ and a number $s_t \in \mathbb{N}$ with $s_t \geq \langle A_i^{(t)} \rangle + \langle b_i^t \rangle$ for all constraints i.

Then we want an algorithm that can optimize any linear function over a polyhedron from that class $\mathcal{P}$:

Optimization problem for class $\mathcal{P} = \{P_t : t \in \mathcal{T}\}$.
Input: $t \in \mathcal{T}$ and $c \in \mathbb{Q}^{n_t}$
Goal: Find a point $x^* \in P_t$ attaining $\max\{c^T x \mid x \in P_t\}$ or decide that P_t is empty.

We can also formalize what it means to find a violated inequality for the class:

Separation problem for class $\mathcal{P} = \{P_t : t \in \mathcal{T}\}$.
Input: $t \in \mathcal{T}$ and $z \in \mathbb{Q}^{n_t}$
Goal: Either decide correctly that $z \in P_t$ or provide a vector $a \in \mathbb{Q}^{n_t}$ with $a^T x < a^T z$ for all $x \in P_t$.

Before we come to the main theorem, we need to introduce a new concept. Note that a *convex body* is a set $K \subseteq \mathbb{R}^n$ that is convex, compact (i.e. closed and bounded) and fulldimensional.

Definition 11.7. For a convex body $K \subseteq \mathbb{R}^n$, the *polar* is defined as

$$K^\circ = \{y \in \mathbb{R}^n \mid y^T x \leq 1 \ \forall x \in K\}$$

Intuitively, the polar is obtained by turning feasible points into valid inequalities and vice versa. We write $\text{int}(K) := \{x \in \mathbb{R}^n \mid \exists \varepsilon > 0 : B(x, \varepsilon) \subseteq K\}$ as the *interior* of K. One can show the following properties:

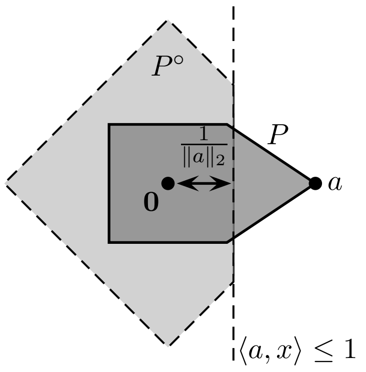
Proposition 11.8. *Let $P \subseteq \mathbb{R}^n$ be a polytope with $\mathbf{0} \in int(P)$. Then:*

(A) P° is a polytope with $\mathbf{0} \in int(P)$

(B) $(P^\circ)^\circ = P$

(C) a is a vertex of $P \Leftrightarrow a^T x \leq 1$ is a facet-defining inequality for P°.

Note that similar statements can be shown for general convex bodies.



Theorem 11.9 (Equivalence of Optimization and Separation — Grötschel, Lovasz, Schrijver [GLS88])**.** *Let $\mathcal{P} = \{P_t : t \in \mathcal{T}\}$ be a proper class of polyhedra. Then the optimization problem for $\mathcal{P}$ is solvable in polynomial time if and only if the separation problem for $\mathcal{P}$ is solvable in polynomial time.*

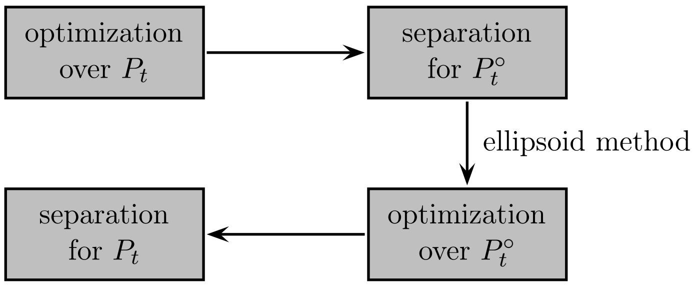
Proof. To keep the argument simple, we assume that the polyhedra P_t are full dimensional with $\mathbf{0} \in P_t$ and the inequalities are normalized so that $P_t = \{x \in \mathbb{R}^{n_t} \mid a^T x \leq 1 \ \forall a \in \mathcal{A}_t\}$. We show both directions separately.

(I) *Separation $\Rightarrow$ Optimization.* This is exactly the task done by the ellipsoid method as stated in Theorem 11.5.

(II) *Optimization ⇒ Separation.* Suppose we are given a point $z \in \mathbb{Q}^n$ for which have to find a violated constraint, that means a vector $a \in \mathcal{A}_t$ with $a^T z > 1$ (if there is any). Consider the *polar* of P_t which is

$$P_t^\circ = \{y \in \mathbb{R}^{n_t} : y^T x \leq 1 \;\; \forall x \in P_t\}$$

Observe that for the separation problem for P_t° one is given a y^* and has to determine whether there is an $x \in P_t$ with $x^T y^* > 1$. But that is exactly the optimization problem for P_t for which we have a polynomial time algorithm by assumption. Then by (I), we can solve the optimization problem for P_t° in polynomial time using the ellipsoid method. By Proposition 11.8.(B) we know that $(P_t^\circ)^\circ = P_t$. Applying (I) with reversed roles, the optimization oracle for P_t° implies we can solve the separation problem for $(P_t^\circ)^\circ = P_t$. That concludes the claim.



Overview over the
optimization ⇒ separation direction

□

11.3 Clarkson's algorithm for linear programming

Consider a linear program of the form $\max\{c^T x \mid Ax \leq b\}$ with $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. In this section we are particularly interested in the case that m is very large compared to n. We will discuss the remarkable argument by Clarkson [Cla95] which rather surprisingly shows that one can solve any LP by solving $O(n \log(m))$ many LPs that contain a subset of only $O(n^2)$ many constraints each. In the exposition we follow the lecture notes by Eisenbrand[1].

It will be convinient to change the notation. Instead of relying on matrix notation $Ax \leq b$, we denote the set of *constraints* by $\mathcal{H}$. For concreteness, one may imagine that the elements of $\mathcal{H}$ are then of the form $(A_i, b_i) \in \mathcal{H}$. We will make a few assumptions on the LP that are all without loss of generality. We assume that the LP is bounded and all solutions are contained in a bounded region describe by

[1]See `https://www.epfl.ch/labs/disopt/wp-content/uploads/2018/09/lec3.pdf`.

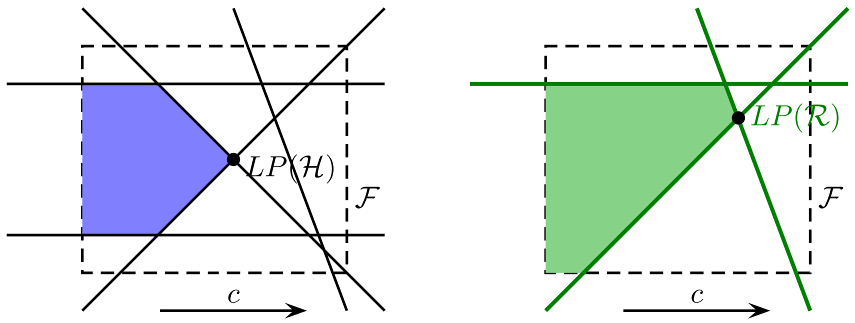
constraint $\mathcal{F}$ (for example $\mathcal{F}$ could be the constraints $x_i \leq M$ and $x_i \geq -M$ for a large enough M). We will also assume that $\max\{c^T x \mid x \text{ satisfies } \mathcal{F} \cup \mathcal{R}\}$ has a unique optimum for all subsets $\mathcal{R} \subseteq \mathcal{H}$ of constraints.

11.3.1 The Sampling Lemma

For $\mathcal{R} \subseteq \mathcal{H}$, let

$$LP(\mathcal{R}) := \text{argmax}\{c^T x \mid x \text{ satisfies } \mathcal{F} \cup \mathcal{R}\}$$

be the (unique) optimum LP solution respecting the constraints in $\mathcal{R}$ as well as the bounding constraints $\mathcal{F}$. One might be tempted to believe that for a small subset $\mathcal{R} \subseteq \mathcal{H}$ of constraints, the solution $LP(\mathcal{R})$ has little to do with the actual solution $LP(\mathcal{H})$ — after all using fewer constraints will in general provide a solution that has a strictly better objective function value but is not feasible for $\mathcal{H}$.



Surprisingly one can prove that on *average*, $LP(\mathcal{R})$ violates only few constraints in $\mathcal{H}$. This will be the core argument making the algorithm work. In the following we write $\mathcal{R} \sim \binom{\mathcal{H}}{r}$ if we draw r many constraints uniformly at random (without repetition) from $\mathcal{H}$.

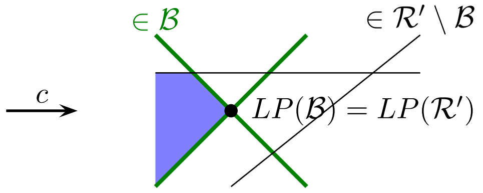
Lemma 11.10 (Sampling Lemma)**.** *Let $m := |\mathcal{H}|$ and $1 \leq r \leq m$. Draw $\mathcal{R} \sim \binom{\mathcal{H}}{r}$ and let $\mathcal{V} := \{h \in \mathcal{H} \mid h \text{ is violated by } LP(\mathcal{R})\}$. Then $\mathbb{E}[|\mathcal{V}|] \leq \frac{mn}{r+1}$.*

Proof. We note that for $\mathcal{R} \sim \binom{\mathcal{H}}{r}$ and $h \sim \mathcal{H} \setminus \mathcal{R}$ independently, the tuple $(\mathcal{R}, h)$ has the same distribution as $(\mathcal{R}' \setminus h', h')$ where we sample $\mathcal{R}' \sim \binom{\mathcal{H}}{r+1}$ and $h' \sim \mathcal{R}'$. With this observation we claim that

$$\begin{aligned} \mathbb{E}[|\mathcal{V}|] &= m \cdot \Pr_{(\mathcal{R},h)}[h \text{ is violated by } LP(\mathcal{R})] \\ &= m \mathop{\mathbb{E}}_{\mathcal{R}' \sim \binom{\mathcal{H}}{r+1}} \Big[\underbrace{\Pr_{h' \sim \mathcal{R}'}[h' \text{ is violated by } LP(\mathcal{R}' \setminus h')]}_{\leq \frac{n}{r+1} \text{ by Claim I}} \Big] \leq \frac{mn}{r+1} \end{aligned}$$

It remains to argue the following:
Claim I. *For any subset $\mathcal{R}' \subseteq \mathcal{H}$ one has $\Pr_{h' \sim \mathcal{R}'}[h' \text{ violated by } LP(\mathcal{R}' \setminus h')] \leq \frac{n}{|\mathcal{R}'|}$.*

Proof of Claim I. By assumption, the LP is bounded and there is a unique optimum solution $x^* := LP(\mathcal{R}')$ (before removing constraint h'). From Chapter 3.2 we know that x^* is determined by a basis $\mathcal{B} \subseteq \mathcal{F} \cup \mathcal{R}'$ of $|\mathcal{B}| = n$ many constraints which in particular satisfies $LP(\mathcal{B}) = LP(\mathcal{R}')$.


$\in \mathcal{B}$
$\in \mathcal{R}' \setminus \mathcal{B}$
c
$LP(\mathcal{B}) = LP(\mathcal{R}')$

If $h' \notin \mathcal{B}$, then the LP optimum does not change if we delete h'. Hence

$$\Pr_{h' \sim \mathcal{R}'}[h' \text{ violates } LP(\mathcal{R}' \setminus h')] \leq \Pr_{h' \sim \mathcal{R}'}[h' \in \mathcal{B}] \leq \frac{n}{|\mathcal{R}'|}$$

as claimed. $\square$

11.3.2 The algorithm

Now we turn the insight from the Sampling Lemma into an algorithm. We use the *multiplicative weight update method.* We start by giving each constraint $h \in \mathcal{H}$ a weight of $w_h := 1$. Then we run the Sampling Lemma and double the weight of those constraints that have been violated, hence making it more likely that those constraints appear in the sample $\mathcal{R}$ in future iterations. Then we run the Sampling Lemma again and so on and so forth. Eventually we will draw a sample $\mathcal{R}$ so that $LP(\mathcal{R}) = LP(\mathcal{H})$ which means that $LP(\mathcal{R})$ is an optimum feasible solution.

Clarkson's algorithm

Input: Constraints $\mathcal{H}$ and objective function c.
Output: Optimum LP solution and constraints $\mathcal{R} \subseteq \mathcal{H}$ with $|\mathcal{R}| \leq r$ and $LP(\mathcal{R}) = LP(\mathcal{H})$.

```
(1) Set $w_h := 1$ for all $h \in \mathcal{H}$
(2) FOR $t = 1$ TO $T := \Theta(n \ln(m))$ DO
        (3) Sample $r$ many contraints $\mathcal{R}$ from $\mathcal{H}$, treating $\mathcal{H}$ as a multiset
            containing $w_h$ copies of constraint $h$.
        (4) Let $\mathcal{V} := \{h \in \mathcal{H} : LP(\mathcal{R}) \text{ violates } h\}$
        (5) If $\mathcal{V} = \emptyset$ then return $LP(\mathcal{R})$ else for each $h \in \mathcal{V}$, update $w_h := 2w_h$
(6) RETURN "fail"
```

Theorem 11.11. *Choosing $T := \Theta(n \ln(m))$ and $r := \Theta(n^2)$, Clarkson's algorithm succeeds with probability at least $1 - \frac{1}{2m}$.*

Proof. For the sake of analysis we modify the algorithm and instead of terminating in (5), we will always let it run for exactly T iterations even if we already found a solution. Let $w_h^{(t)}$ be the weight of constraint h at the beginning of iteration t and let $W^{(t)} := \sum_{h\in\mathcal{H}} w_h^{(t)}$ be the total weight. Similarly let $\mathcal{R}^{(t)}$ and $\mathcal{V}^{(t)}$ denote the corresponding constraints in iteration t. By the Sampling Lemma (Lemma 11.10), the expected weight of the violated constraints in iteration t is $\mathbb{E}[\sum_{h\in\mathcal{V}^{(t)}} w_h^{(t)}] \leq \frac{n}{r+1} \cdot W^{(t)}$. Hence the total updated weight is

$$\mathbb{E}[W^{(t+1)}] \leq \Big(1 + \frac{n}{r+1}\Big) \cdot W^{(t)}$$

Hence the algorithm terminates with an expected weight of

$$\mathbb{E}[W^{(T+1)}] \leq \Big(1 + \frac{n}{r}\Big)^T \cdot \underbrace{W^{(1)}}_{=m} \overset{1+x\leq e^x}{\leq} \exp\Big(\frac{nT}{r}\Big) \cdot m$$

By Markov's inequality we know that $W^{(T+1)} \leq 2m^2 \cdot \exp(\frac{nT}{r})$ with probability at least $1 - \frac{1}{2m}$. We claim that if this event happens, then the algorithm will have been successful. For the sake of contradiction suppose that we had $\mathcal{V}^{(t)} \neq \emptyset$ in each iteration t. Let $\mathcal{B} \subseteq \mathcal{H}$ be a basis characterizing $LP(\mathcal{H})$, i.e. $|\mathcal{B}| = n$ and $LP(\mathcal{B}) = LP(\mathcal{H})$. Every intermediate solution $LP(\mathcal{R}^{(t)})$ has a strictly better objective than $LP(\mathcal{H})$ and so each such solution must violate at least one constraint in $\mathcal{B}$. By averaging, there must be some constraint $h \in \mathcal{B}$ in the basis that has been violated at least $\frac{T}{n}$ times during the course of the algorithm. Then the weight of that constraint alone at the end provides

$$2^{T/n} \leq w_h^{(T+1)} \leq W^{(T+1)} \leq 2m^2 \cdot \exp\Big(\frac{nT}{r}\Big) \leq 2m^2 \cdot 2^{\frac{T}{2n}}$$

choosing $r := 4n^2$ and using $e \leq 2^2$. Then taking $\log_2$'s on both sides gives

$$\frac{T}{n} \leq \frac{T}{2n} + \log_2(2m^2)$$

which can be rearranged to $T \leq O(n \log(m))$. □

We would like to point out that Clarkson's algorithm uses very few properties of LPs. It can be made to work for many other problems where the optimum solution is always determined by a smaller set of constraints. One example where this applies is *integer linear programming* $\max\{c^T x \mid Ax \leq b, x \in \mathbb{Z}^n\}$ where one can always find a subsystem of 2^n many constraints that have the same optimum solution as the whole system.

Chapter 12

The Frank-Tardos Algorithm*

There are two popular computational models:

- The *Turing machine model* where the alphabet has constant size and one operation manipulates one symbol on a tape.
- The *Random access machine (RAM)* where one memory slot can store any integer or rational number and we can add / multiply / divide in one operation.

If the input consists of vector $a \in \mathbb{Q}^n$, then the *input length* in the Turing machine model is the encoding length $\langle a \rangle$ and in the RAM model it is the number of objects n. We say that an algorithm that takes $a \in \mathbb{Q}^n$ as input has *strongly polynomial time* if its running time is $\text{poly}(\langle a \rangle)$ if counted in the Turing machine model and $\text{poly}(n)$ if counted in the RAM model. Here we use the same notation $\langle a \rangle$ that we introduced in Chapter 11. For example, the algorithm by Tardos for minimum cost circulation (see Section 6.6) is strongly polynomial while the interior point method (Section 8) is only weakly polynomial, i.e. to solve an LP $\max\{c^T x \mid Ax \leq b\}$ it takes $\text{poly}(\langle A \rangle, \langle b \rangle, \langle c \rangle)$ iterations in both the Turing machine model and the RAM model. In this chapter, we want to provide a useful tool for turning some weakly polynomial time algorithms into strongly polynomial ones. During this chapter we will say that on input of $a \in \mathbb{Q}^n$, an algorithm *runs in at most poly(n) many RAM operations* where we implicitly mean that also the intermediate numbers have an encoding length of at most $\text{poly}(\langle a \rangle)$[1].

For $z \in \mathbb{R}$ we set

$$\text{sign}(z) := \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z = 0 \\ -1 & \text{if } z < 0 \end{cases}$$

The central result of this chapter will be:

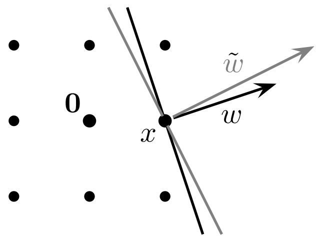
[1]This technicality is needed. For example it is possible to solve the **NP**-hard Knapsack problem in $\text{poly}(n)$ many RAM operations if arbitrary intermediate numbers are allowed.

Theorem 12.1 (Frank, Tardos [FT87]). *Given a vector $w \in \mathbb{Q}^n$ and $N \in \mathbb{N}$, one can compute a vector $\tilde{w} \in \mathbb{Z}^n$ with $\|\tilde{w}\|_\infty \leq 2^{O(n^3)} N^{O(n^2)}$ so that*

$$\mathit{sign}(w^T x) = \mathit{sign}(\tilde{w}^T x) \quad \forall x \in \{-N, \ldots, N\}^n$$

The number of RAM operations is bounded by a polynomial in n and $\log(N)$.

Intuitively this means that if we have any optimization problem where the solutions are contained in $\{-N, \ldots, N\}^n$, then we can replace a linear objective function $x \mapsto w^T x$ by another one $x \mapsto \tilde{w}^T x$ so that the set of optimum solutions does not change. Then the encoding length of the objective vector as been reduced to $\langle \tilde{w} \rangle \leq \text{poly}(n, \log(N))$.



12.1 Simultaneuous Diophantine Approximation

The most important ingredient for our algorithm is a consequence of the celebrated LLL-algorithm by Lenstra, Lenstra and Lovász [LLL82] to efficiently find a *simultanuous diophantine approximation* to a rational vector. First, recall *Minkowski's Theorem* which is usually stated as the fact that any symmetric convex body $K \subseteq \mathbb{R}^n$ with $\text{Vol}_n(K) \geq 2^n$ must have $K \cap (\mathbb{Z}^n \setminus \{\mathbf{0}\}) \neq \emptyset$. It will be convinient for us to restate a lattice variant:

Theorem 12.2 (Lattice variant of Minkowski's First Theorem). *Let $B \in \mathbb{R}^{n \times n}$ be a full rank matrix. Then there exists an $x \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$ so that $\|Bx\|_\infty \leq |\det(B)|^{1/n}$.*

Minkowski's result is based on the *pigeonhole principle* and does not provide an algorithm for finding the vector x. However, the *lattice basis reduction algorithm* by Lenstra, Lenstra and Lovász (also called the *LLL-algorithm*) can be used to find a 2^n-approximation to the shortest vector in a lattice. In our case this gives the following:

Theorem 12.3 (LLL algorithm [LLL82]). *Let $B \in \mathbb{Q}^{n \times n}$ be a full rank matrix. Then in time poly($\langle B \rangle$) one can find an $x \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$ so that $\|Bx\|_\infty \leq 2^{n-1} \cdot |\det(B)|^{1/n}$.*

Now to our application to find good rational approximations:

Theorem 12.4. *Given $\alpha \in [-1,1]^n$ and $N \in \mathbb{N}$, there is an algorithm to find integers $p_1,\dots,p_n,q \in \mathbb{Z}$ so that $q \in \{1,\dots,2^{2n^2}N^n\}$ so that*

$$|q\alpha_i - p_i| \leq \frac{1}{N} \quad \forall i \in [n]$$

The number of RAM operations is bounded by a polynomial in n and $\log(N)$.

Note that the condition means that the numbers α_i are approximated by a rational number $\frac{p_i}{q}$ with common denominator up to an error of $\frac{1}{qN}$. This is a lot better than just picking any q and rounding each α_i to the nearest multiple of $\frac{1}{q}$.

Proof of Theorem 12.4. For some parameter $Q > 0$ that we determine later, consider the $(n+1)\times(n+1)$ matrix

$$B = \begin{pmatrix} 1 & 0 & \dots & 0 & -\alpha_1 \\ 0 & 1 & \dots & 0 & -\alpha_2 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \dots & 1 & -\alpha_n \\ 0 & 0 & \dots & 0 & \frac{1}{Q^{n+1}} \end{pmatrix}$$

We note that $\det(B) = \frac{1}{Q^{n+1}}$ and hence by Minkowski's Theorem (Theorem 12.2) there is a nonzero integer vector $x = (p_1,\dots,p_n,q)$ with $\|Bx\|_\infty \leq \frac{1}{Q}$. However, instead we use the constructive version from Theorem 12.3 and compute an $x = (p_1,\dots,p_n,q)$ with $\|Bx\|_\infty \leq \frac{2^n}{Q}$ in polynomial time. By symmetry we may assume that $q \geq 0$. Writing out the properties we see that

$$|p_i - \alpha_i q| \leq \frac{2^n}{Q} \ \ \forall i = 1,\dots,n \quad \textit{and} \quad |q| \leq 2^n Q^n$$

Setting $Q := 2^n N$ then gives the claim (also note that $q = 0$ is impossible). However there is one small technicality: the way we stated the argument, the running time would depend on $\langle\alpha\rangle$. But we can first truncate α_i's after poly$(n, \log(N))$ bites before applying the LLL algorithm and absorb the made error into the slack that we have. $\square$

12.2 Warmup

To warm up, we want to prove the following:

Lemma 12.5. *Let $w \in \mathbb{Q}^n$ and $N \in \mathbb{N}$. Then there is an algorithm to find $\tilde{w} \in \mathbb{Z}^n$ with $\|\tilde{w}\|_\infty \leq 2^{3n^2}N^n$ in poly$(n, \log(N))$ many RAM operations so that*

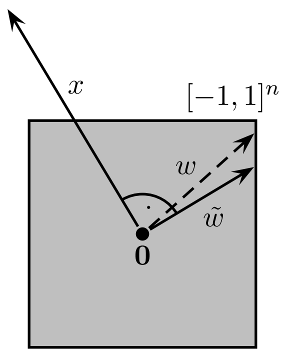
$$\forall x \in \{-N,\dots,N\}^n : \quad sign(\tilde{w}^T x) \neq 0 \quad \Longrightarrow \quad \big(sign(w^T x) = sign(\tilde{w}^T x)\big)$$

Proof. We may scale w so that $\|w\|_\infty = 1$. Applying Theorem 12.4 with w and parameter $N' := 2nN$, we can obtain a $q \in \mathbb{N}$ with $q \leq 2^{2n^2}(2nN)^n \leq 2^{3n^2}N^n$ and $\tilde{w} \in \mathbb{Z}^n$ so that $|qw_i - \tilde{w}_i| \leq \frac{1}{2nN}$ for all $i = 1, \dots, n$. Fix an $x \in \{-N, \dots, N\}^n$ with $\text{sign}(\tilde{w}^T x) \neq 0$. For symmetry reasons suppose $\tilde{w}^T x \geq 1$. Then

$$|(qw - \tilde{w})^T x| \leq \sum_{i=1}^n \underbrace{|qw_i - \tilde{w}_i|}_{\leq \frac{1}{2nN}} \cdot \underbrace{\|x\|_\infty}_{\leq N} \leq \frac{1}{2}$$

and so $qw^T x \geq \tilde{w}^T x - \frac{1}{2} \geq \frac{1}{2}$, i.e. $\text{sign}(w^T x) = \text{sign}(\tilde{w}^T x)$. □

Note that if $\tilde{w}^T x = 0$, then we cannot say anything about the sign of $w^T x$. On the other hand, there must be some coordinate where $\|w\|_\infty = 1$ is attained — say $|w_1| = 1$. Then $|qw_1 - \tilde{w}_1| \leq \frac{1}{2nN} < 1$. Hence $qw_1 - \tilde{w}_1 = 0$. So one could iterate and approximate $w - \frac{\tilde{w}}{q}$ which is a lot shorter than the original vector w and also has support at most $n - 1$.



12.3 The main proof

Now, we want to extend Lemma 12.5 to an iterative argument.

Lemma 12.6. *Let $w \in \mathbb{Q}^n$ and $N \in \mathbb{N}$. There is an algorithm that in time poly$(n, \log(N))$ computes vectors $v_1, \dots, v_k \in \mathbb{Z}^n$ and coefficients $\lambda_1, \dots \lambda_k > 0$ with $k \leq n$ so that*

(i) $w = \sum_{i=1}^k \lambda_i v_i$

(ii) $\|v_i\|_\infty \leq 2^{2n^2} N^n$ *for all* $i \geq 1$.

(iii) $\frac{\lambda_i}{\lambda_{i-1}} \leq \frac{1}{N\|v_i\|_\infty}$ *for all* $i \geq 2$.

(iv) $\|w\|_\infty = \|\lambda_1 v_1\|_\infty$

Proof. The claim is invariant under scaling w, hence we may assume that $\|w\|_\infty = 1$. We apply Theorem 12.4 to w and parameter N and obtain a vector $v_1 \in \mathbb{Z}^n$ and $q_1 \in \{1, \ldots, 2^{2n^2} N^n\}$ so that $\|q_1 w - v_1\|_\infty \leq \frac{1}{N}$. Note that in particular $\|v_1\|_\infty \leq 2^{2n^2} N^n$. We set $\lambda_1 := \frac{\|w\|_\infty}{q_1} = \frac{1}{q_1}$. Similar as before, $|\text{supp}(w - \lambda_1 v_1)| = |\text{supp}(q_1 w - v_1)| \leq n - 1$. Also note that $\|w\|_\infty = \|\frac{v_1}{q_1}\|_\infty = \|\lambda_1 v_1\|_\infty$ which gives (iv). We apply induction (over the support of the vector) to the remainder $w - \lambda_1 v_1$ and obtain $v_2, \ldots, v_k \in \mathbb{Z}^n$ and $\lambda_2, \ldots, \lambda_k > 0$ so that

$$(i') \quad w - \lambda_1 v_1 = \sum_{i=2}^{k} \lambda_i v_i \qquad (ii') \quad \|v_i\|_\infty \leq 2^{2n^2} N^n \;\; \forall i \geq 2$$

$$(iii') \quad \frac{\lambda_i}{\lambda_{i-1}} \leq \frac{1}{N\|v_i\|_\infty} \;\; \forall i \geq 3 \qquad (iv') \quad \|w - \lambda_1 v_1\|_\infty = \|\lambda_2 v_2\|_\infty$$

We note that (i') can be rearranged to $w = \sum_{i=1}^k \lambda_i v_i$ which gives (i). Combining (ii') and the choice of v_1 gives (ii). Finally to complete (iii), we note that

$$\frac{\lambda_2}{\lambda_1} \overset{(iv')\ \&\ \lambda_1 = \frac{1}{q_1}}{=} \frac{1}{\|v_2\|_\infty} \underbrace{q_1 \|w - \lambda_1 v_1\|_\infty}_{\leq 1/N} \leq \frac{1}{N\|v_2\|_\infty}$$

□

Now we can finish the proof of the main result.

Theorem 12.1. We are given a vector $w \in \mathbb{Q}^n$ and a parameter $N \in \mathbb{N}$. Apply Lemma 12.6 with vector w and parameter $N' := 4nN$. Let $v_1, \ldots, v_k \in \mathbb{Z}^n$ and $\lambda_1, \ldots, \lambda_k > 0$ be the obtained vectors and coefficients with $\|v_i\|_\infty \leq 2^{2n^2}(N')^n \leq 2^{3n^3} N^n$. We set $M := 4 \cdot 2^{4n^2} N^{n+1}$ and set

$$\tilde{w} := \sum_{i=1}^{k} M^{k-i} v_i$$

Note that $\tilde{w} \in \mathbb{Z}^n$ and $\|\tilde{w}\|_\infty \leq \sum_{i=1}^k M^{k-i}\|v_i\|_\infty \leq 2M^k 2^{2n^2}(N')^n \leq 2^{O(n^3)} N^{O(n^2)}$ as claimed. Next fix an $x \in \{-N, \ldots, N\}^n$; our goal is to prove that $\text{sign}(w^T x) = \text{sign}(\tilde{w}^T x)$. First consider the case that $v_1^T x = \ldots = v_k^T x = 0$. Then $w^T x = \sum_{i=1}^k \lambda_i v_i^T x = 0$ and $\tilde{w}^T x = \sum_{i=1}^k M^{k-i} v_i^T x = 0$ and hence $\text{sign}(w^T x) = \text{sign}(\tilde{w}^T x)$ is satisfied. It remains to analyze the case where $v_i^T x \neq 0$ for at least one index. In that case, the least such index will determine the sign.

Claim I. *Suppose there is an index $j \in \{1, \ldots, k\}$ with $v_1^T x = \ldots = v_{j-1}^T x = 0$ and $v_j^T x \neq 0$. Then sign$(w^T x) =$ sign$(v_j^T x)$.*

Proof of Claim I. Assume w.l.o.g. that $v_j^T x \geq 1$. Then

$$w^T x \geq \sum_{i=1}^{j-1} \lambda_i \underbrace{v_i^T x}_{=0} + \lambda_j \underbrace{v_j^T x}_{\geq 1} - n \sum_{i=j+1}^{k} \underbrace{\lambda_i \|v_i\|_\infty}_{\leq \lambda_{i-1}/(4nN)} \underbrace{\|x\|_\infty}_{\leq N} \geq \lambda_j - \frac{1}{4} \underbrace{(\lambda_j + \ldots + \lambda_k)}_{\leq 2\lambda_j} > 0 \quad \square$$

Now back to the main statement. Suppose that $v_1^T x = \ldots = v_{j-1}^T x = 0$ and $v_j^T x \geq 1$. Then

$$\begin{aligned}
\tilde{w}^T x &= \sum_{i=1}^{k} M^{k-i} v_i^T x \\
&\geq M^{k-j} \underbrace{v_j^T x}_{\geq 1} - \sum_{i=j+1}^{k} M^{k-i} \cdot n \cdot \underbrace{\|v_i\|_\infty}_{\leq 2^{3n^3} N^n} \cdot \underbrace{\|x\|_\infty}_{\leq N} \\
&\geq M^{k-j} - 2^{4n^2} N^{n+1} \underbrace{\Big(\sum_{i=j+1}^{k} M^{k-i} \Big)}_{\leq 2M^{k-(j+1)}} \\
&\geq M^{k-j} - 2 \cdot 2^{4n^2} N^{n+1} M^{k-(j+1)} > 0
\end{aligned}$$

as $M = 4 \cdot 2^{4n^2} N^{n+1}$. Hence $\text{sign}(\tilde{w}^T x) = \text{sign}(v_j^T x) = \text{sign}(w^T x)$ using Claim I. $\square$

12.4 Applications

Finally, we discuss a few applications. First, we can turn any binary weakly polynomial optimization problem into a strongly polynomial one.

Theorem 12.7. *Let $\mathcal{F} \subseteq \{0,1\}^n$. If there is a weakly polynomial time algorithm for $\max\{c^T x \mid x \in \mathcal{F}\}$, then there is also a strongly polynomial time algorithm for the same problem.*

Proof. We use Theorem 12.1 to replace the vector $c \in \mathbb{Q}^n$ in poly(n) arithmetic operations by a vector $\tilde{c} \in \mathbb{Z}^n$ with $\|\tilde{c}\|_\infty \leq 2^{O(n^3)}$ so that $\text{sign}(c^T x) = \text{sign}(\tilde{c}^T x)$ for all $x \in \{-1,0,1\}^n$. Then every solution $x \in \mathcal{F}$ that is optimal for $\tilde{c}$ is also optimal for c. Hence we can run the weakly polynomial time algorithm on objective function $\tilde{c}$ with takes time poly(n) since $\langle \tilde{c} \rangle \leq \text{poly}(n)$. $\square$

Theorem 12.8. *Given $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$ and $c \in \mathbb{Q}^n$. Then one can solve the LP $\max\{c^T x \mid Ax \leq b\}$ in poly$(\langle A \rangle, \langle b \rangle)$ many RAM operations.*

Proof. For the sake of simplicity assume the LP is feasible and bounded. Let $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. There is some $M > 0$ with $\|M\|_\infty \leq \text{poly}(\langle A \rangle, \langle b \rangle)$ so that $Q := P \cap [-M, M]^n$ contains an optimum solution. Next, one can choose $N \leq \text{poly}(\langle A \rangle, \langle b \rangle)$ so that all extreme points of $N \cdot Q$ are integral. Then apply Theorem 12.1 with vector c and parameter MN and obtain an approximation $\tilde{c}$. Then solve the LP $\max\{\tilde{c}^T x \mid x \in N \cdot Q\}$. $\square$

We can extend this to the LP feasibility problem and drop the dependence on $\langle b \rangle$:

Theorem 12.9. *Let $A \in \mathbb{Q}^{m\times n}$ and $b \in \mathbb{Q}^m$ and set $P := \{x \in \mathbb{R}^n \mid Ax \leq b\}$. Then one can compute a point $x^* \in P$ in RAM-time poly($\langle A\rangle$) or decide that $P = \emptyset$.*

Proof. First we show that we can test whether $P \neq \emptyset$ in time polynomial in $\langle A\rangle$. And in fact, by the Farkas Lemma (Prop 3.22.(I)) we have

$$P \neq \emptyset \quad \iff \quad \min\{b^T y \mid A^T y = \mathbf{0},\ y \geq \mathbf{0}\} = 0$$

and the right hand side can be evaluated in time poly($\langle A\rangle$) using Theorem 12.8.

Now assume $P \neq \emptyset$. In order to actually find a feasible point, we define

$$P(I,J) := \big\{x \in \mathbb{R}^n \mid Ax \leq b \text{ and } A_i x = b_i\ \forall i \in I,\ \ x_j = 0\ \forall j \in J\big\}$$

for index sets $I \subseteq [m]$ and $J \subseteq [n]$. Then using the above test repeatedly on $P(I,J)$ we can compute an inclusion-wise maximum pair (I^*, J^*) so that $P(I^*, J^*) \neq \emptyset$. Then there will be a unique solution x to system of linear equations $A_i x = b_i\ \forall i \in I^*,\ \ x_j = 0\ \forall j \in J^*$ and that solution can be computed using Gaussian elimination. □

It is indeed possible to solve any LP $\max\{c^T x \mid Ax \leq b\}$ in time poly($\langle A\rangle$), that means without any dependence on b and c. This was proven by Tardos [Tar86]. On the other hand, whether any LP can be solved in strongly polynomial time is still a major open problem.

Chapter 13

Submodular functions*

Numerous optimization problems from machine learning over approximation algorithms to combinatorial optimization share a *diminishing returns* property. Instead of problem-specific approaches one can capture many such problems with the framework of submodular functions. We already mentioned submodular functions briefly in Chapter 9, however this chapter is written so that it can be read independently of Chapter 9.

13.1 Introduction

The key definition is as follows:

Definition 13.1. Let X be a finite set. A function $f : 2^X \to \mathbb{R}$ is called *submodular* if

$$f(A \cup \{j\}) - f(A) \geq f(B \cup \{j\}) - f(B) \quad \forall A \subseteq B \subseteq X \quad \forall j \in X \setminus B$$

In other words: a set function is submodular if adding a new element j to a smaller set A causes greater (or equal) increase than adding it to a larger set B. Before we discuss concrete examples, we want to elaborate that there are several equivalent definitions:

Lemma 13.2 (Equivalent Characterizations Submodularity)**.** *Let $f : 2^X \to \mathbb{R}$. Then the following is equivalent*

(I) One has $f(A \cup \{j\}) - f(A) \geq f(B \cup \{j\}) - f(B) \quad \forall A \subseteq B \subseteq X \quad \forall j \in X \setminus B$
(II) One has $f(A) + f(B) \geq f(A \cap B) + f(A \cup B) \quad \forall A, B \subseteq X$

Additional desirable properties or subclasses of submodular functions are:

- *Non-negativity*, i.e. $f(S) \geq 0 \; \forall S \subseteq X$
- *Monotonicity*, i.e. $f(A) \leq f(B) \; \forall A \subseteq B \subseteq X$
- *Symmetry*, i.e. $f(S) = f([n] \setminus S) \; \forall S \subseteq X$

We want to give a few examples of submodular functions.

- **Coverage functions.** Let $S_1, \dots, S_n \subseteq U$ be a *set family* over a *ground set* U. Then the number of covered elements given by $f(I) := |\bigcup_{i \in I} S_i|$ is a monotone submodular function.

- **Cut functions in graphs.** Let $G = (V, E)$ be an undirected graph with edge weights $w : E \to \mathbb{R}_{\geq 0}$. Then the function $f(S) := w(\delta(S)) = \sum_{\{i,j\} \in E : |\{i,j\} \cap S| = 1} w(i, j)$ giving the value of the cut S is a symmetric submodular function (though it is not monotone!). Similar for directed graphs $D = (V, A)$, the value $w(\delta^+(S))$ of a directed cut is submodular (though neither symmetric nor monotone).

- **Matroid rank functions.** As we already showed in Section 9.4, for any matroid $M = (X, \mathcal{I})$, the rank function $r_M : 2^X \to \mathbb{Z}_{\geq 0}$ is monotone submodular.

- **Log determinant function.** Let $A \in \mathbb{R}^{n \times n}$ be symmetric and positive definite matrix. For $S \subseteq [n]$ we denote $A_{S,S}$ as the $|S| \times |S|$ principal submatrix with rows and columns from S. Then using Ky Fan's inequality one can show that the function $f(S) := \ln(\det_{|S|}(A_{S,S}))$ is submodular.

The two main results that we want to prove in the remainder of this chapter are:

(i) For the problem of maximizing a monotone submodular function subject to a cardinality constraint one can find a $(1 - \frac{1}{e})$-approximate solution using a simple greedy algorithm,

(ii) One can minimize any submodular function in polynomial time.

For both algorithms it suffices to have *oracle access* to the function f, that means the algorithms only need to be able to evaluate the function value $f(S)$ for any given set S. Note that the standard algorithm to find a minimum cut in a graph is not quite trivial. One consequence of (ii) is that one find a minimum cut in a graph in polynomial time without knowing the graph, as long as we can determine the values of the cuts.

13.2 Maximizing a monotone submodular functions with a cardinality constraint

In this section, we study the following problem

MONOTONE SUBMODULAR FUNCTION MAXIMIZATION
Input: A non-negative monotone submodular function $f : X \to \mathbb{R}_{\geq 0}$ and a parameter $k \in \mathbb{N}$.
Goal: Solve $\max\{f(S) : S \subseteq X \text{ and } |S| \leq k\}$.

Without the cardinality constraint $|S| \leq k$, the optimum would simply be attained by $S = X$. Note that this problem already captures the maximum set coverage problem $\max\{|\bigcup_{i\in I} S_i| : |I| \leq k\}$ which is known to be **NP**-hard to approximate within a factor of $1 - \frac{1}{e} - \varepsilon$ for any $\varepsilon > 0$. On the positive side, we will see that a simple greedy algorithm can already achieve this constant. We denote $f(A \mid B) := f(A \cup B) - f(A)$ as the *marginal increase* when adding A to B. In particular for single elements $i \in X$ we will write $f(i \mid A) = f(A \cup \{i\}) - f(A)$.

The Submodular Greedy Algorithm

Input: A monotone submodular function $f : 2^X \to \mathbb{R}_{\geq 0}$ and $k \in \mathbb{Z}_{\geq 0}$

Output: A set $(1 - \frac{1}{e})$-approximation to $\max\{f(S) \mid |S| = k\}$.

```
(1) Set $S_0 := \emptyset$
(2) FOR $i = 1$ TO $k$ DO
        (3) Let $x_i \in X \setminus S_{i-1}$ be the element maximizing $f(x_i \mid S_{i-1})$
        (4) Set $S_i := S_{i-1} \cup \{x_i\}$
(5) Return $S_k$
```

The analysis is due to Nemhauser while in the exposition we follow the survey of Buchbinder and Feldman[BF18]. A useful inequality is the following:

Lemma 13.3. *Let $f : 2^X \to \mathbb{R}$ be a submodular function. Then for any $A, B \subseteq X$ one has $\sum_{i\in B} f(i \mid A) \geq f(B \mid A)$.*

Proof. W.l.o.g. suppose $B = \{1, \dots, k\}$. Then

$$f(B \mid A) = \sum_{i=1}^{k} f(i \mid A \cup \{1, \dots, i-1\}) \overset{f \text{ submod.}}{\leq} \sum_{i=1}^{k} f(i \mid A).$$

□

Now we can analyze the algorithm:

Theorem 13.4 (Nemhauser, Wolsey, Fisher [NWF78])**.** *The greedy algorithm gives a $(1 - \frac{1}{e})$-approximation for maximizing a non-negative monotone submodular function subject to a cardinality constraint.*

Proof. Let $S^* := \text{argmax}\{f(S) \mid S \subseteq X, |S| = k\}$ be the optimum solution. We will prove that $f(S_k) \geq (1 - \frac{1}{e}) f(S^*)$ where S_k is the last iterate in the algorithm. Let $i \in \{1, \dots, k\}$. The crucial argument is that the *marginal increase* $f(x_i \mid S_{i-1})$ is at least as large as if we were adding an *average* element from S^*. More precisely,

$$\begin{aligned}
f(S_i) - f(S_{i-1}) &= f(x_i \mid S_{i-1}) \\
&\geq \mathop{\mathbb{E}}_{x\sim S^*}\big[f(x \mid S_{i-1})\big] \\
&\overset{f \text{ subm.+Lem 13.3}}{\geq} \frac{1}{k}\big(f(S^* \cup S_{i-1}) - f(S_{i-1})\big) \\
&\overset{f \text{ monotone}}{\geq} \frac{1}{k}\big(f(S^*) - f(S_{i-1})\big)
\end{aligned}$$

In other words, in every iteration we close at least a $\frac{1}{k}$-fraction of the remaining gap to the optimum value. This can be rearranged to

$$f(S^*) - f(S_i) \leq \Big(1 - \frac{1}{k}\Big) \cdot (f(S^*) - f(S_{i-1})).$$

Iterating this inequality k times gives

$$f(S^*) - f(S_k) \leq \underbrace{\Big(1 - \frac{1}{k}\Big)^k}_{\leq 1/e} \cdot \big(f(S^*) - \underbrace{f(S_0)}_{\geq 0}\big) \leq \frac{1}{e} \cdot f(S^*)$$

which means that $f(S_k) \geq (1 - \frac{1}{e}) \cdot f(S^*)$ as claimed. □

We would also like to note that the greedy algorithm makes at most nk queries to the function f.

13.3 Polymatroids

In order to design a polynomial time algorithm to minimize any submodular function we need to take a detour and discuss a polyhedral generalization of matroids first. Our exposition will follow Chapter 44 of Schrijver [Sch03]. For a vector $x \in \mathbb{R}^X$ and $S \subseteq X$ we use as before the notation $x(S) := \sum_{i\in S} x_i$.

Definition 13.5. For a submodular function $f : 2^X \to \mathbb{R}$ we define the *polymatroid associated with* f as

$$P_f := \big\{x \in \mathbb{R}^X_{\geq 0} \mid x(S) \leq f(S)\ \forall S \subseteq X\big\}$$

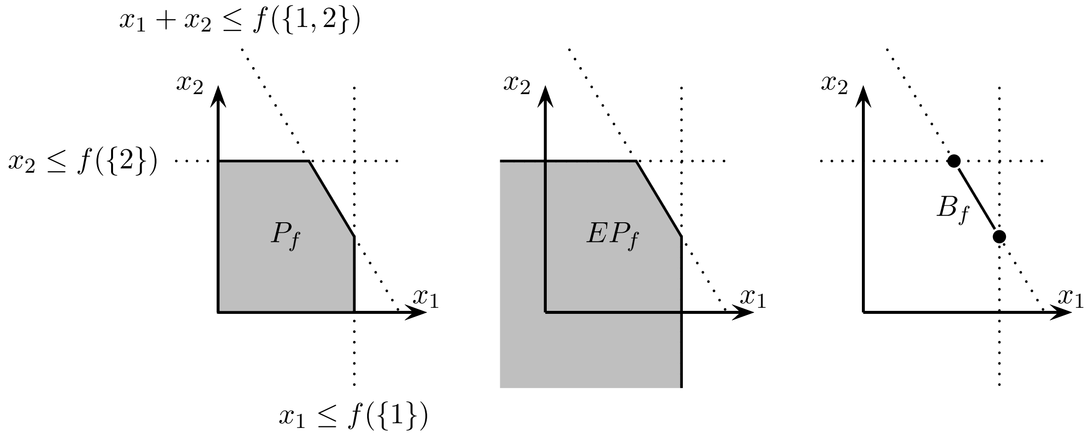
The *extended polymatroid* is

$$EP_f := \big\{x \in \mathbb{R}^X \mid x(S) \leq f(S)\ \forall S \subseteq X\big\}$$

The *base polytope of* f is

$$B_f := EP_f \cap \big\{x \in \mathbb{R}^n \mid x(X) = f(X)\big\}$$

Here is an example in dimension $X = \{1, 2\}$:



For example, for a matroid $M = (X, \mathcal{I})$, the matroid independence polytope $P(\mathcal{I})$ (see Sec 9.3) is identical to the polymatroid P_{r_M} associated with the rank function of the matroid. Moreover we have $\text{conv}\{\mathbf{1}_S : S \subseteq X \text{ is basis of } M\} = B_{r_M}$.

In order to obtain some more intuition about the geometry of polymatroids, we will show that the inequalities inducing a *face* of the polyhedron EP_f come from sets S that are closed under union and intersection:

Lemma 13.6. *Let $x \in EP_f$ where $f : 2^X \to \mathbb{R}$ is submodular. Then the family of sets $\mathcal{F} := \{S \subseteq X \mid x(S) = f(S)\}$ is closed under taking unions and intersections.*

Proof. We fix $x \in EP_f$ and consider the function $g(S) := f(S) - \sum_{i \in S} x_i$. This is again a submodular function and $g(S)$ gives the *slack* that the point x has with respect to the inequality induced by S. Then $g(S) \geq 0$ for all $S \subseteq X$ and $g(S) = 0$ for all $S \in \mathcal{F}$. Now consider $A, B \in \mathcal{F}$. Then

$$0 \leq \underbrace{g(A \cup B)}_{\geq 0} + \underbrace{g(A \cap B)}_{\geq 0} \overset{\text{subm.}}{\leq} \underbrace{g(A)}_{=0} + \underbrace{g(B)}_{=0} \overset{A,B \in \mathcal{F}}{=} 0$$

Hence all inequalities have to be equalities and so $A \cup B, A \cap B \in \mathcal{F}$. $\square$

As a side remark, the proof also shows the following:

Corollary 13.7. *Let $f : 2^X \to \mathbb{R}$ be a submodular function and let $\mathcal{F}$ be the family of minimizers of f. Then $\mathcal{F}$ is closed under taking unions and intersections.*

Note that the same claim is in general false for the set of maximizers.

13.4 The Polymatroid Greedy algorithm

Next, we want to find a polynomial time algorithm to optimize a non-negative objective function $c \in \mathbb{R}^X_{\geq 0}$ over the extended polymatroid EP_f. In order to develop some intuition, let us first revisit the matroid greedy algorithm from Chapter 2.2. Clearly, that algorithm uses the notion of independent sets which does not exist for polymatroids, so we first develop a different view on the matroid greedy algorithm.

13.4.1 Another look at the Matroid Greedy algorithm

Consider a matroid $M = ([n], \mathcal{I})$ and a objective function vector $c \in \mathbb{R}^n$ with the indices sorted so that $c_1 \geq c_2 \geq \ldots \geq c_n$. Recall that the matroid greedy algorithm goes through the elements $1, \ldots, n$ in this order and picks elements if the current set remains independent. In each iteration i the greedy algorithm maintains a basis of $\{1, \ldots, i\}$ and so we know that if $r_M(\{1, \ldots, i\}) = r_M(\{1, \ldots, i-1\}) + 1$, then we select the ith element into the independent set and otherwise — that means if $r_M(\{1, \ldots, i\}) = r_M(\{1, \ldots, i-1\})$ — then we would not pick the ith element. Then the matroid greedy algorithm could be equivalently described as follows:

```
Matroid Greedy algorithm
Input: Matroid $M = ([n], \mathcal{I})$ and $c \in \mathbb{R}^n$.
Output: $x^* = \mathrm{argmax}\{c^T x \mid x \in P(\mathcal{I})\}$
  (1) Set $x^* := \mathbf{0}$
  (2) Sort indices so that $c_1 \geq c_2 \geq \ldots \geq c_n \geq 0$
  (3) FOR $i = 1$ TO $n$ DO
        (4) Set $x_i := r_M(\{1, \ldots, i\}) - r_M(\{1, \ldots, i-1\})$
  (5) Return $x^*$
```

From our previous analysis it then follows that the computed vector x^* is the characteristic vector of an optimum independent set and hence also optimum solution to $\max\{c^T x \mid x \in P(\mathcal{I})\}$. The above algorithm does not rely on the notion of independent sets but just on the rank function. Moreover — at least syntactically — the algorithm appears to make sense also without the special properties of the rank function such as integrality, monotonicity and the fact that $r_M(S \cup \{i\}) - r_M(S) \in \{0, 1\}$.

13.4.2 The Greedy algorithm for extended polymatroids

Now we want to extend these ideas for a more general problem:

EXTENDED POLYMATROID OPTIMIZATION
Input: A submodular function $f : 2^{[n]} \to \mathbb{R}$ and an objective $c \in \mathbb{R}^n$.
Goal: Solve $\max\{c^T x \mid x \in EP_f\}$.

For an algorithm that solves the extended polymatroid optimization problem, we can make 3 simplifying assumptions:

- *Assumption I. One has* $f : 2^{[n]} \to \mathbb{R}_{\geq 0}$. Reason is that if f is not non-negative, then $EP_f = \emptyset$ and the problem is infeasible.
- *Assumption II. One has* $f(\emptyset) = 0$. Note that if f is non-negative and $f(\emptyset) > 0$, then one can set $f(\emptyset) := 0$ without changing EP_f and while keeping f submodular.
- *Assumption III. One has* $c \in \mathbb{R}^n_{\geq 0}$. Easy to see that if c is not non-negative, then the problem is unbounded.

Then inspired by the matroid greedy algorithm, we suggest the following generalization

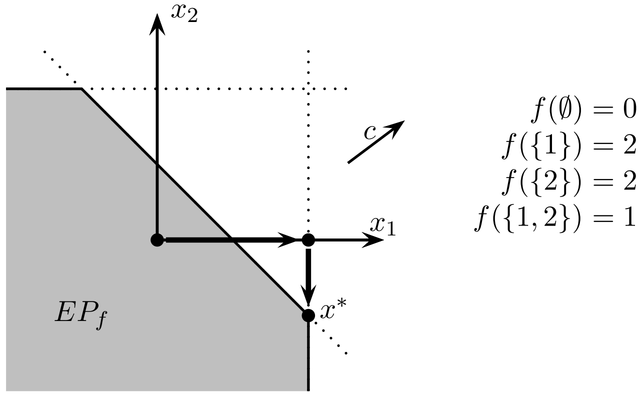
Greedy algorithm for Extended Polymatroids

Input: Submodular function $f : 2^{[n]} \to \mathbb{R}$ with $f(\emptyset) = 0$ and $c \in \mathbb{R}^n_{\geq 0}$.

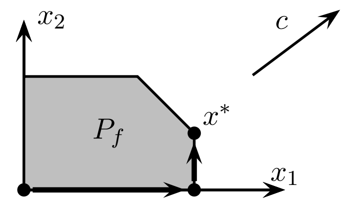
Output: $x^* = \text{argmax}\{c^T x \mid x \in EP_f\}$

```
(1) Set $x^* := \mathbf{0}$
(2) Sort indices so that $c_1 \geq c_2 \geq \ldots \geq c_n \geq 0$
(3) FOR $i = 1$ TO $n$ DO
        (4) Set $x^*_i := f(\{1, \ldots, i\}) - f(\{1, \ldots, i-1\})$.
```

Note that we do not make the assumption that f is monotone, so if one thinks of the computed point x^* as a sequence of iterates, then it is not true that all the iterates are in EP_f. This is different from the matroid greedy algorithm where we always maintain an independent set:



On the other hand, if f is monotone, then clearly $x^* \geq \mathbf{0}$ and automatically $x^* \in P_f \subseteq EP_f$.



Visualization of polymatroid greedy algorithm if f is monotone

13.4.3 Correctness of the Polymatroid Greedy Algorithm

It remains to prove that the Polymatroid Greedy Algorithm indeed computes an optimum solution. First we verify that the computed solution is feasible.

Lemma 13.8. *The point x^* computed in the Polymatroid Greedy Algorithm is in EP_f.*

Proof. Assume again that $c_1 \geq \ldots \geq c_n \geq 0$. Let us denote $P_i := \{x \in \mathbb{R}^n \mid x(S) \leq f(S)\ \forall S \subseteq \{1,\ldots,i\}\}$. Then by induction over i we can prove that $x^* \in P_i$. The base case is true as $x^* \in \mathbb{R}^n = P_0$. Now suppose $x^* \in P_{i-1}$ and for $S \subseteq \{1,\ldots,i\}$ with $i \in S$ we want to prove that also $x^*(S) \leq f(S)$. And indeed,

$$\begin{aligned} x^*(S) &= x^*(S \setminus \{i\}) + x_i^* \\ &\overset{\text{induction}}{\leq} f(S \setminus \{i\}) + x_i^* \\ &\overset{\text{choice } x_i^*}{=} f(\underbrace{S \setminus \{i\}}_{S \cap [i-i]}) + f(\underbrace{[i]}_{S \cup [i-1]}) - f([i-1]) \\ &\overset{f \text{ subm.}}{\leq} f(S) \end{aligned}$$

□

It remains to prove optimality of the Polymatroid Greedy Algorithm which will be do using LP duality. One can easily verify that the following is a pair of primal and dual LPs, see Theorem 3.26:

(Primal(f,c))	(Dual(f,c))
$\max \sum_{i=1}^n c_i x_i$	$\min \sum_{S \subseteq [n]} f(S) y_S$
$\sum_{i \in S} x_i \leq f(S) \quad \forall S \subseteq [n]$	$\sum_{S \subseteq [n]} y_S \mathbf{1}_S = c \quad \forall S \subseteq [n]$
	$y_S \geq 0 \quad \forall S \subseteq [n]$

Note that (Primal(f,c)) is precisely the polymatroid optimization problem. In order to show that the polymatroid greedy algorithm finds an optimum solution to

($\textsc{Primal}(f,c)$) we will provide a solution to ($\textsc{Dual}(f,c)$) that has the same objective function value.

For a vector $c_1 \geq \ldots \geq c_n$, the greedy algorithm constructs a solution x^* where the constraints $x^*([i]) \leq f([i])$ seem to be the only relevant constraints — then it is natural to suspect that there is an optimum dual solution that puts positive weight only on the dual variables that correspond to those constraints.

Lemma 13.9. *Let $f : 2^{[n]} \to \mathbb{R}$ be submodular with $f(\emptyset) = 0$ and let $c \in \mathbb{R}^n_{\geq 0}$ with $c_1 \geq \ldots \geq c_n$. Then the following pair (x^*, y^*) are optimum solutions to ($\textsc{Primal}(f,c)$) and ($\textsc{Dual}(f,c)$):*

$$x^*_i := f([i]) - f([i-1]) \quad \forall i \in [n] \qquad \begin{array}{rcl} y^*_{[i]} & := & c_i - c_{i+1}, \\ y^*_{[n]} & = & c_n, \\ y^*_S & := & 0 \quad \textit{otherwise} \end{array}$$

Proof. We have already proven that $x^* \in EP_f$ meaning that x^* is feasible for ($\textsc{Primal}(f,c)$). Next, clearly $y^* \geq \mathbf{0}$ and moreover $\sum_{S \subseteq [n]} y^*_S \mathbf{1}_S = \sum_{i=1}^n y^*_{[i]} \mathbf{1}_{[i]} = \sum_{i=1}^{n-1} (c_i - c_{i+1}) e_i + c_n e_n = c$. Hence y^* is feasible for ($\textsc{Dual}(f,c)$). Then it remains to verify that the objective function values coincide. And indeed

$$c^T x^* \;=\; \sum_{i=1}^n c_i\big(f([i]) - f([i-1])\big) = \sum_{i=1}^{n-1} f([i]) \cdot (c_i - c_{i+1}) + f([n]) c_n = \sum_{S \subseteq [n]} f(S) y^*_S$$

□

Overall, this implies that the greedy algorithm finds an optimum solution. We summarize:

Theorem 13.10 (Edmonds [Edm70]). *Let $c \in \mathbb{R}^n$ and let $f : 2^{[n]} \to \mathbb{R}$ be submodular. Then one can find an optimum for $\max\{c^T x \mid x \in EP_f\}$ in polynomial time. The only access to f are $O(n)$ many value oracle calls.*

13.4.4 Consequences

We want to briefly discuss a few immediate consequences of the polymatroid greedy algorithm. First, it seems a bit odd that the greedy algorithm optimizes over the unbounded object EP_f instead of over P_f. But we can obtain the following:

Theorem 13.11. *Let $f : 2^X \to \mathbb{R}$ be submodular and let $c \in \mathbb{R}^X$. Then one can solve the problem $\max\{c^T x \mid x \in P_f\}$ in polynomial time. The only access to f is through a polynomial number of oracle calls.*

Proof. By Theorem 13.10 and the *equivalence of optimization and separation* from Theorem 11.9, we can solve the *separation problem* for EP_f in polynomial time. Since

$P_f = EP_f \cap \mathbb{R}^X_{\geq 0}$ we can also separate in polynomial time over P_f. Again using the equivalence of optimization and separation we can then optimize any linear function over P_f in polynomial time. □

We can also draw some consequences if the submodular function is integral:

Lemma 13.12. *Let $f : 2^X \to \mathbb{Z}$ be submodular. Then EP_f is an integral polyhedron.*

Proof. Suppose again that $X = \{1, \dots, n\}$. It suffices to show that for any objective function $c \in \mathbb{R}^n$ the LP $\max\{c^T x : x \in EP_f\}$ is either unbounded or has an integral optimum solution. Assume $c \in \mathbb{R}^n_{\geq 0}$ since otherwise the LP is unbounded. Then the polymatroid greedy algorithm computes an optimum solution x^* whose coordinates are of the form $x^*_i = f(\{\pi(1), \dots, \pi(i)\}) - f(\{\pi(1), \dots, \pi(i-1)\}) \in \mathbb{Z}$ where π is the permutation with $c_{\pi(1)} \geq \dots \geq c_{\pi(n)} \geq 0$. □

13.5 Polynomial time minimization of submodular functions

Next, we want to show how one can use the polymatroid greedy algorithm to minimize an arbitrary submodular function f. Note that it is far from obvious how that would work as function f merely appears as right hand sides of the extended polymatroid EP_f.

SUBMODULAR FUNCTION MINIMIZATION
Input: A submodular function $f : 2^X \to \mathbb{R}$.
Goal: Solve $\min\{f(S) \mid S \subseteq X\}$.

First, we translate f so that[1] $f(\emptyset) = 0$. Note that then, the submodular function minimization problem is only non-trivial if $f(S^*) < 0$ for the optimum set $S^* \subseteq [n]$. The key ingredient is the definition of an auxiliary submodular function with interesting properties:

Lemma 13.13. *Let $f : 2^X \to \mathbb{R}$ be a submodular function with $f(\emptyset) = 0$. Define the function $f_{\min} : 2^X \to \mathbb{R}$ by $f_{\min}(S) := \min_{A \subseteq S} f(A)$. Then (I) $f_{\min}$ is submodular and (II) $EP_{f_{\min}} = EP_f \cap \mathbb{R}^X_{\leq 0}$.*

Proof. We leave it to the reader to verify (I). Next, we prove both inclusions of (II). Each $x \in EP_{f_{\min}}$ satisfies $x_i \leq f_{\min}(\{i\}) \leq f(\emptyset) = 0$ for each $i \in X$ and so $x \in \mathbb{R}^X_{\leq 0}$. Also $\sum_{i \in S} x_i \leq f_{\min}(S) \leq f(S)$ and so $x \in EP_f$. On the other hand,

[1]Formally we could minimize the function $f' : 2^{[n]} \to \mathbb{R}$ with $f'(S) := f(S) - f(\emptyset)$ which again is submodular and satisfies $f'(\emptyset) = 0$.

suppose $x \in EP_f \cap \mathbb{R}^X_{\leq 0}$. We claim that for $S \subseteq X$ one also has $x(S) \leq f_{\min}(S)$. Let $A^* := \text{argmin}\{f(A) \mid A \subseteq S\}$ be the set attaining that minimum. Then

$$x(S) \overset{x_i \leq 0\ \forall i}{\leq} x(A^*) \leq f(A^*) = f_{\min}(S)$$

and hence $x \in EP_{f_{\min}}$. □

At first it doesn't seem that the function $f_{\min}$ is helpful — even evaluating $f_{\min}(X)$ is as hard as minimizing f in the first place. Still we can use it to prove the following:

Lemma 13.14. *Let $f : 2^X \to \mathbb{R}$ be submodular with $f(\emptyset) = 0$. Then*

$$\max\Big\{\sum_{i\in X} x_i \mid x \in EP_f \cap \mathbb{R}^X_{\leq 0}\Big\} = \min\big\{f(S) \mid S \subseteq X\big\}$$

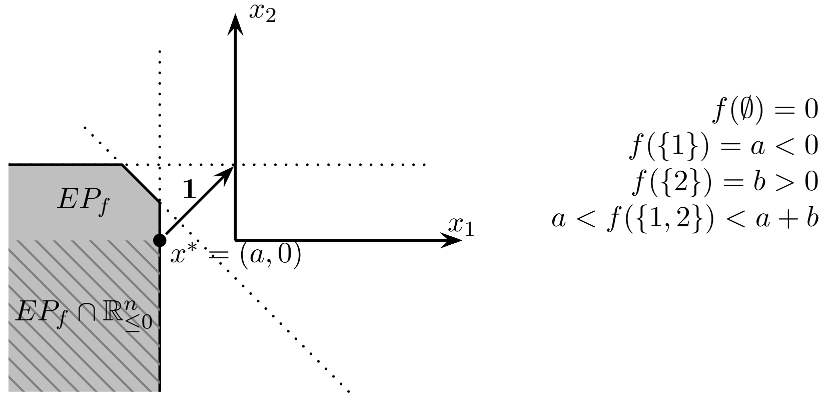
Proof. Let us revisit the characterization of the optimum primal and dual LP solution from Lemma 13.9 for the objective function $c := \mathbf{1}$ and the submodular function $f_{\min}$:

$$\max\Big\{\sum_{i\in X} x_i \mid x \in EP_{f_{\min}}\Big\} = \min\Big\{\sum_{S\subseteq X} f_{\min}(S) y_S \mid \sum_{S\subseteq X} y_S \mathbf{1}_S = \mathbf{1}\ \forall S \subseteq X,\ y \in \mathbb{R}^{2^X}_{\geq 0}\Big\}$$

Then we know that the right hand side is attained by a vector y^* with $y^*_X = 1$ and $y^*_S = 0$ for all $\emptyset \subseteq S \subset X$. With this insight we see that

$$\begin{aligned}\max\Big\{\sum_{i\in X} x_i \mid x \in EP_f \cap \mathbb{R}^X_{\leq 0}\Big\} &= \max\Big\{\sum_{i\in X} x_i \mid x \in EP_{f_{\min}}\Big\} \\ &= f_{\min}(X) = \min\big\{f(S) \mid S \subseteq X\big\}\end{aligned}$$

as claimed. □



It then follows:

Theorem 13.15 ([GLS88]). *Let $f : 2^X \to \mathbb{R}$ be a submodular function. Then one can find an optimum S^* to $\min\{f(S) \mid S \subseteq X\}$ in polynomial time.*

Proof. We can shift the function so that $f(\emptyset) = 0$. By Theorem 13.11, we can solve the separation problem for EP_f in polynomial time. Then trivially we can also separate over $EP_f \cap \mathbb{R}^n_{\leq 0}$ in polynomial time. Then using the ellipsoid method / the equivalence of optimization and separation from Theorem 11.9 can also optimize any linear function over $EP_f \cap \mathbb{R}^n_{\leq 0}$ in polynomial time. Then by Lemma 13.14 we can determine the *value* of $\min\{f(S) \mid S \subseteq X\}$ in polynomial time. Then using a standard argument we can determine one element in the optimum set after the other. $\square$

Bibliography

[BF18] Niv Buchbinder and Moran Feldman. Submodular functions maximization problems. In Teofilo F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics, Second Edition, Volume 1: Methologies and Traditional Applications*, pages 753–788. Chapman and Hall/CRC, 2018.

[Bub15] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, nov 2015.

[CCPS98] William J. Cook, William H. Cunningham, William R. Pulleyblank, and Alexander Schrijver. *Combinatorial Optimization.* John Wiley & Sons, Inc., USA, 1998.

[Cla95] Kenneth L. Clarkson. Las vegas algorithms for linear and integer programming when the dimension is small. *J. ACM*, 42(2):488–499, 1995.

[Die12] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics.* Springer, 2012.

[Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[Din70] Yefim Dinitz. Algorithm for solution of a problem of maximum flow in a network with power estimation. *Doklady Akademii Nauk SSSR*, 1970.

[Edm65] Jack Edmonds. Maximum matching and a polyhedron with $0,1$-vertices. *J. Res. Nat. Bur. Standards Sect. B*, 69B:125–130, 1965.

[Edm70] Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and their Applications (Proc. Calgary Internat. Conf., Calgary, Alta., 1969)*, pages 69–87. ,, 1970.

[EK72] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.

[FF56] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[FT87] András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Comb.*, 7(1):49–65, 1987.

[GLS88] Martin Grötschel, Lászlo Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. 1988.

[Gro53] A. Grothendieck. Résumé de la théorie métrique des produits tensoriels topologiques. *Bol. Soc. Mat. São Paulo*, 8:1–79, 1953.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

[Hås97] Johan Håstad. Some optimal inapproximability results. In *STOC*, pages 1–10. ACM, 1997.

[Kha79] L. G. Khachyian. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR*, 244(5):1093–1096, 1979.

[Kri79] J.-L. Krivine. Constantes de Grothendieck et fonctions de type positif sur les sphères. *Adv. in Math.*, 31(1):16–30, 1979.

[KV12] B. H. Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer-Verlag, New York, NY, 2012.

[LLL82] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

[McM70] P. McMullen. The maximum numbers of faces of a convex polytope. *Mathematika*, 17(2):179–184, 1970.

[NWF78] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978.

[Pri57] R. C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.

[Sch99] Alexander Schrijver. *Theory of linear and integer programming.* Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1999.

[Sch03] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency.* Springer, 2003.

[Sch17] Alexander Schrijver. A course in combinatorial optimization. 03 2017.

[Tar85] Éva Tardos. A strongly polynomial minimum cost circulation algorithm. *Comb.*, 5(3):247–256, 1985.

[Tar86] Éva Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Oper. Res.*, 34(2):250–256, 1986.

[Tut50] W. T. Tutte. The factorization of locally finite graphs. *Canad. J. Math.*, 2:44–49, 1950.

[Ver18] Roman Vershynin. *High-dimensional probability*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics.* Cambridge University Press, Cambridge, 2018. An introduction with applications in data science, With a foreword by Sara van de Geer.