SOS-Report

Priyanshu Kumar

May 2024

Contents

1 Algorithm and Matroid Polytopes with preliminaries on polyhedra

1.1 Convex Function

A subset C of $\mathbb{R}^n$ is said to be convex if $\lambda x + (1-\lambda)y \in C \ \forall \ x, y \in C$ and $\lambda \in [0,1]$.

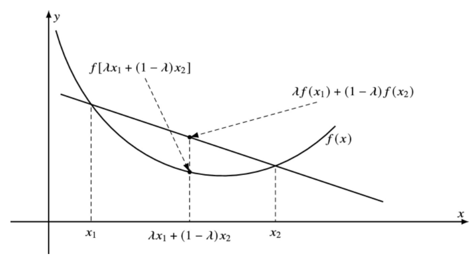

Figure 1: Example of a Convex Set

Let's explore it through some of amazing stuff like proof:-

Ques 1. :- Prove that $C = \{(x_1, x_2) : 2x_1 + 3x_2 = 7\} \subset \mathbb{R}^2$ is a convex set.

Sol:- Let us suppose that X,Y $\in C$, where X $= (x_1, x_2)$, Y $= (y_1, y_2)$. Line segment is connected by X and Y sets.
As per given in definition,

$$M = \{M \mid M = \lambda X + (1-\lambda)Y \ , 0 \leq \lambda \leq 1\}$$

Let's
for some $\lambda \in [0,1]$, $M = (m_1, m_2)$ which is point sets M.
Here, w_1 can be written as:

$$\{\lambda x_1 + (1-\lambda)y_1\}$$

$$m_2 = \{\lambda x_2 + (1-\lambda)y_2\}$$

As $X, Y \in C$, we write

$$\{2x_1 + 3x_2 = 7\}$$

and

$$\{2y_1 + 3y_2 = 7\}$$

According to the formula,

$$2m_1 + 3m_2 = 2[\lambda x_1 + (1-\lambda)x_2] + 3[\lambda x_2 + (1-\lambda)y_2]$$

on solving it we will obtain that

$$\lambda 7 + (1 - \lambda) 7 = 7$$

since, given was... $2x_1 + 3x_2 = 7$

1.2 Convex Hull

The convex hull of a set $X \subseteq \mathbb{R}^n$ is the smallest convex set containing X. Formally, it is defined as the set of all convex combinations of points in X.

$$\text{conv.hull}(X) = \left\{ \sum_{i=1}^{k} \lambda_i x_i \mid x_i \in X, \lambda_i \geq 0, \sum_{i=1}^{k} \lambda_i = 1 \right\}$$

Algorithm:-

```
findConvexHull(points, n)
Input: The points, number of points.
Output: Corner points of convex hull.
Begin
   start := points[0]
   for each point i, do
      if points[i].x < start.x, then // get the left most point
         start := points[i]
   done
   current := start
   add start point to the result set.
   define colPts set to store collinear points
   while true, do //start an infinite loop
      next := points[i]
      for all points i except 0th point, do
         if points[i] = current, then
            skip the next part, go for next iteration
         val := cross product of current, next, points[i]
         if val > 0, then
            next := points[i]
            clear the colPts array
         else if cal = 0, then
            if next is closer to current than points[i], then
               add next in the colPts
               next := points[i]
            else
               add points[i] in the colPts
      done
      add all items in the colPts into the result
```

```
        if next = start , then
            break the loop
        insert next into the result
        current := next
    done
    return result
```
End

Affine Halfspace: A subset H of $\mathbb{R}^n$ is called an affine halfspace if $H = \{x \mid A^T x \leq \delta\}$ for some $A \in \mathbb{R}^n$ with $A \neq 0$ and some $\delta \in \mathbb{R}$.

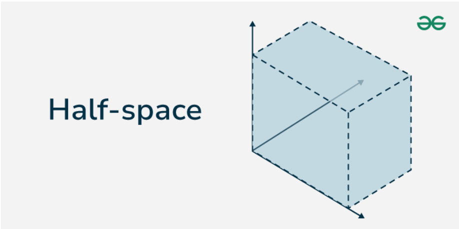

Figure 2: refrence from Gfg

Linear Halfspace: A subset H of $\mathbb{R}^n$ is called an affine halfspace if $H = \{x \mid A^T x \leq \delta\}$ for some $A \in \mathbb{R}^n$ with $A \neq 0$ and some $\delta = 0$.

Uphull of X: X $\subseteq \mathbb{R}^n$. The set of Conv.hull X + $\mathbb{R}^n_+$
downhull of X: X $\subseteq \mathbb{R}^n$. The set of Conv.hull X - $\mathbb{R}^n_+$

1.3 Cone:

$C \subseteq \mathbb{R}^n$ is called Cone if $C \neq \emptyset$ and $\lambda X + \nu Y \in C$ and $\lambda, \nu \in \mathbb{R}_+$. Cone generated by

$$\text{Cone}(X) = \left\{ \sum_{i=1}^{k} \lambda_i x_i \mid x_i \in X, k \geq 0, \lambda_i \geq 0 \right\}$$

Theorem: for any $X \in \mathbb{R}^n$ and $X \in \text{Cone}X$. $\exists$ linearly independent vectors $\{\vec{x_1}, \vec{x}_2, ..., \vec{x}_k\} \in X$ with $x \in \text{Cone}\{\vec{x_1}, \vec{x}_2, ..., \vec{x}_k\}$

1.4 Polyhedral:

A cone C is polyhedral if there is a matrix A such that:

$$\text{C} = \{X | AX \leq 0\}$$

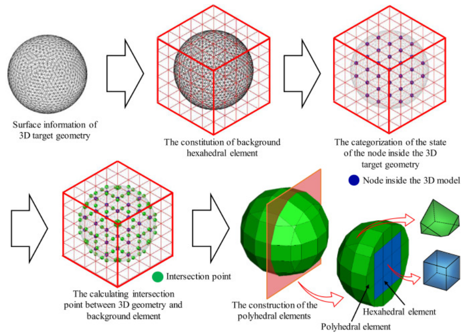

Figure 3: POlyhedral Visualisation

A polyhedral is intersection of finitely many linear halfspace.

1.5 Polyhedon:

$P \subseteq \mathbb{R}^n$ is called polyhedron, $\exists$ mXn matrix A and a vector $b \in \mathbb{R}^m$ $m \geq 0)$ such that:

$$\mathrm{P} = \{X | AX \leq b\}$$

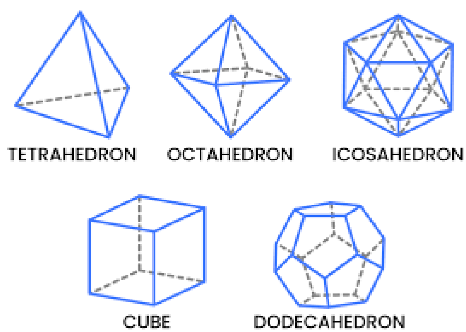

Figure 4: Example of polyhedron

A polyhedron if intersection of finitely many affine halfspace.

1.6 Polytopes:

$P \subseteq \mathbb{R}^n$ is polytopes if it's convex hull finitely many vectors in $\mathbb{R}^n$

$$P = Q + C$$

where; P is Polyhedron, Q is some Polytopes and C is some Cone.

1.7 Charecteristics Cone:

If polyhedron $P \neq \emptyset$, then C is unique is called Char.Cone(P).

$$\text{Char.Cone}(P) = \{y \in \mathbb{R}^n | \forall x \in P, \forall \lambda \geq 0 : x + \lambda y \in P\}$$

If $P = \emptyset$, then Char.Cone$(P) = \{0\}$.
A set P is polytope iff P is polyhedron.

theorem: $AX \leq b$ is feasible $\iff$ $Y^T b \geq 0$ with $Y^T A = 0^T$.

Corollary: $AX = b$ has a solution $x \geq 0$ $\iff$ $y^T b \geq 0$ for each $y \geq 0$ with $y^T A \geq 0^T$.

Corollary: Let $AX \leq b$ be a feasible system of inequalities and let $C^T X \leq \delta$ be an inequality satisfied by each x with $AX \leq b$. Then for some $\delta' \leq \delta$, the inequality $C^T X \leq \delta'$ is a non-negative linear combination of inequalities in $AX \leq b$.

2 Linear and Integer programming

2.1 Linear programming

Linear programming (L.P.) means to concern the problem of maximizing or minimizing a linear function over a polyhedron.

$$\max \left\{C^T X | AX \leq b\right\}$$

$$\min \left\{C^T X | X \geq 0, AX \geq b\right\}$$

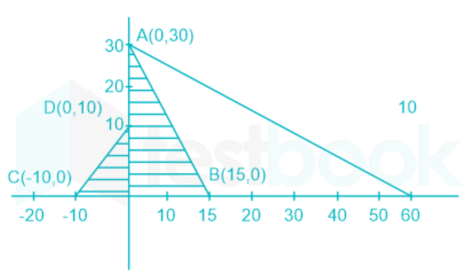

Figure 5: basic example of linear programming

- If a supremum of a linear function over a polyhedron is finite, then it is attained as a maximum (and for infimum and minimum)
- maximum is finite if the value set is non-empty and has an upperbound.

	Linear programming	**Mixed integer linear programming**
Game theory	Dominated strategies Minimax strategies Correlated equilibrium Optimal mixed strategies to commit to	Nash equilibrium Optimal mixed strategies to commit to in more complex settings
Social choice, expressive marketplaces	Winner determination in auctions, exchanges, … with partially acceptable bids	Winner determination in: auctions, exchanges, … without partially acceptable bids; Kemeny, Slater, other voting rules; kidney exchange
Mechanism design	Automatically designing optimal mechanisms that use randomization	Automatically designing optimal mechanisms that do not use randomization

Figure 6: differences

Duality theorem of linear programming: Let A be a matrix and b and c be vectors. Then:

$$\max\left\{C^T x | Ax \leq b\right\} = \min\left\{y^T b | y \geq 0, y^T A = C^T\right\}$$

atleast one of two optima is finite.
Weak duality $\rightarrow C^T X = y^T A x \leq y^T b$

2.1.1 Equivalent Statements:

$$\max\left\{C^T x | x \geq 0, Ax \leq b\right\} = \min\left\{y^T b | y \geq 0, y^T A \geq C^T\right\}$$
$$\max\left\{C^T x | x \geq 0, Ax = b\right\} = \min\left\{y^T b | y \geq 0, y^T A \geq C^T\right\}$$
$$\min\left\{C^T x | x \geq 0, Ax \geq b\right\} = \max\left\{y^T b | y \geq 0, y^T A \leq C^T\right\}$$
$$\min\left\{C^T x | Ax \geq b\right\} = \max\left\{y^T b | y \geq 0, y^T A = C^T\right\}$$

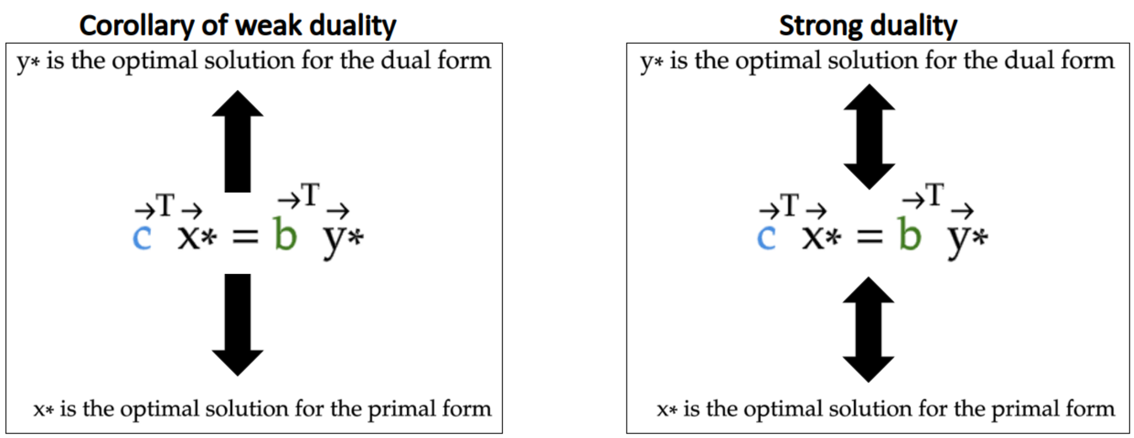

Figure 7: Types of duality theorem

One-to-one relation between constraint in a problem and variable in it's dual problem.

Maximize	**Minimize**
$\leq$ Constraint	Variable ≥ 0
$\geq$ Constraint	Variable ≤ 0
$=$ Constraint	Variable $=$
Variable ≥ 0	$\leq$ Constraint
Variable ≤ 0	$\geq$ Constraint
Unconstrained Variable	$=$ Constraint
Right Hand Side	Objective Function
Objective Function	Right Hand Side

Table 1: Some stuff

2.2 L.P. terminology:

- Polyhedron is Called feasible region.
- Any vector in polyhedron is called Feasible solution.
- Feasible region empty called Non-Feasible.
- Feasible region non-empty called Feasible.
- Function: $x \to c^T x$ called Objective function.
- Objective function is also known as Cost function.
- Any feasible solution attaining the optimum value is called Optimum solution.

Primal and Dual Algebra

Primal

$$
\begin{aligned}
Max \quad & \sum_j c_j X_j \\
s.t. \quad & \sum_j a_{ij} X_j \leq b_i \quad i = 1, \ldots, m \\
& X_j \geq 0 \quad j = 1, \ldots, n
\end{aligned}
\qquad
\begin{aligned}
Max \quad & C'X \\
s.t. \quad & AX \leq b \\
& X \geq 0
\end{aligned}
$$

Dual

$$
\begin{aligned}
\text{Min} \quad & \sum_i b_i Y_i \\
\text{s.t.} \quad & \sum_i a_{ij} Y_i \geq c_j \quad j = 1, \ldots, n \\
& Y_i \geq 0 \quad i = 1, \ldots, m
\end{aligned}
\qquad
\begin{aligned}
Min \quad & b'Y \\
s.t. \quad & A'Y \geq C \\
& Y \geq 0
\end{aligned}
$$

Figure 8: Types of duality in linear programming

- Any inequality $c^T x \leq \delta$ called Active or Tight for some x' if $cx' = \delta$.
- Minimization prorblem is called Dual problem of the maximization problem (then called Primal problem) Conversly.
- Feasible solution of dual problem is called Dual solution.

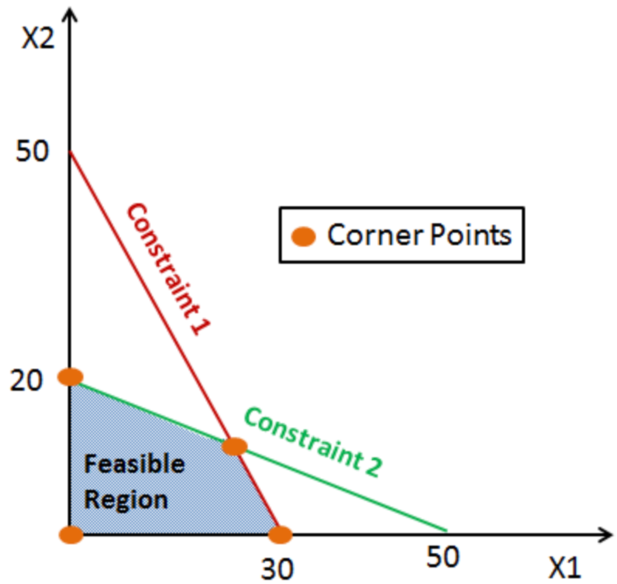

Figure 9: example of feasible region

Now let's move on another topic.

2.3 Complementary Slackness:

This condition of a pair of feasible solution x,y of the linear programs :- x and y are optimum solution iff $(Ax)_i = b_i$ for each i with $y_i > 0$.
It is similar for other dual linear programs.

It says that if a dual variable is greater than zero (slack) then the corresponding primal constraint must be an equality (tight).
It also says that if the primal constraint is slack then the corresponding dual variable is tight (or zero.)
Notice that if y0 were an extreme point in the dual, the complementary slackness condition relates a dual solution y0 to a point x0 in the set F in the primal. When we add to this, the fact that x0 is feasible, we may infer that both points should be optimal.

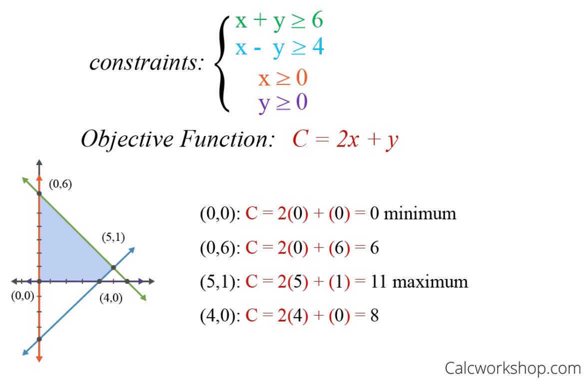

Figure 10: Example of objective function.

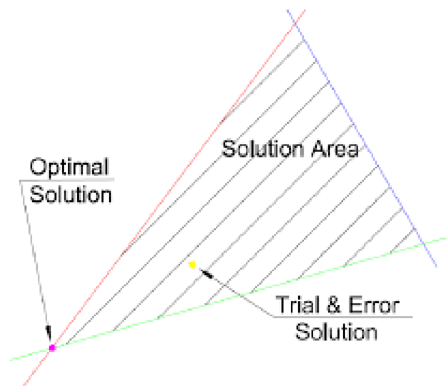

Figure 11: Example of optimum solution

3 Face, Facets and Vertices:

Let,

$$P = \{x|Ax \leq b\}$$

be a polyhedron in $\mathbb{R}^n$

3.1 Supporting hyperplane:

H If C is non-zero vector and

$$\delta = max\left\{C^T x|Ax \leq b\right\},$$

the affine halfspace $\left\{x|C^T x = \delta\right\}$ is called supporting hyperplane of P.

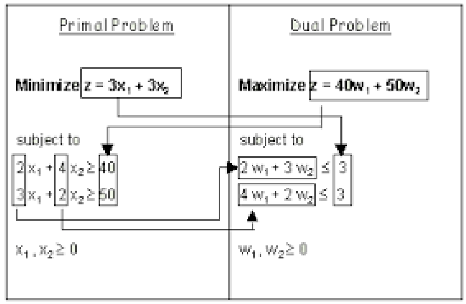

Figure 12: Example of dual problem

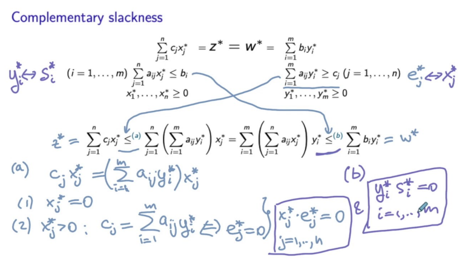

Figure 13: example complementary slackness

3.2 Face:

SUbset F of P is called Face if

$$F = P$$

$$F = P \cap H$$

for some supporting hyperplane H of P.
F is a face of $P \iff F$ is the set of optimum solution of $\max\{C^T x | Ax \leq b\}$ for some $C \in \mathbb{R}^n$.

3.3 Induce face / Determine face:

For any $C^T x \leq \delta$

$$F = \{x \in P \mid C^T x = \delta\}$$

$$F = \{x \in P | A'x = b'\}$$

for some subsystem $A'x \leq b'$ of $Ax \leq b$. So face non-empty polyhedron is non-empty.

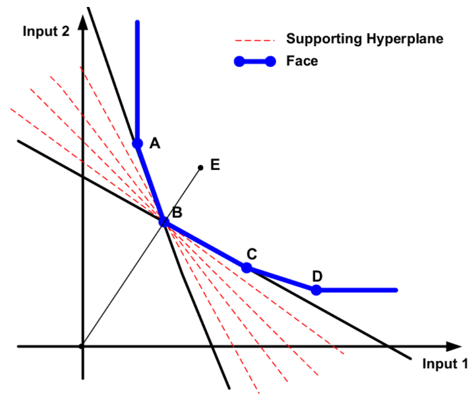

Figure 14: Example of supporting hyperplane and Face

Tight/Active: $Ax \leq b \implies a^T x = \beta$. Constraint $a^T x \leq \beta$ from F face.

Implicitly equality: If $Ax \leq b \implies a^T x \leq \beta$.

3.3.1 Theorem:

Let$P = \{x|Ax \leq b\}$ be a polyhedron in $\mathbb{R}^n$. Let $A'x \leq b'$ be the subsystem of implicit inequalities in $Ax \leq b$. Then:

$$dimP = n - rank(A)'$$

.

3.3.2 Theorem:

Let $P = \{x|Ax \leq b\}$be a polyhedorn in $\mathbb{R}^n$ and let $z \in P$. Then z is a vertex og P iff $rank(A_Z) = n$

4 Polarity:

4.1 Polar:

$$\text{For } C \subseteq \mathbb{R}^n$$

$$\text{Polar of } C(C^*) = \left\{z \in \mathbb{R}^n | x^T z \leq 1, \forall x \in C\right\}$$

4.2 Polar Cone:

$$\text{For } C \subseteq \mathbb{R}^n$$

$$\text{Polar Cone of } C(C^*) = \left\{ z \in \mathbb{R}^n | x^T z \leq 0, \forall z \in C \right\}$$

The polar C^* is equal to the cone generated by the temperature of the rows of A. So,
$C^{**} = C$ for each polyhedral cone$C$.

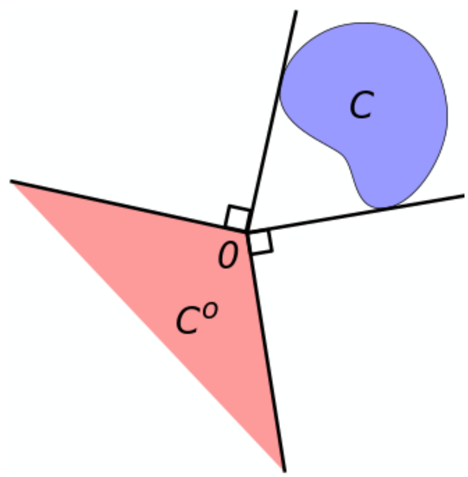

Figure 15: A set C and its polar cone C_o

- Polar cone is always convex even if C is not convex.
- If C is empty set, $C^* = \mathbb{R}^n$.
- Polarity may be seen as a generalisation of orthogonality.

Let C, C_1 and C_2 be non-empty sets in $\mathbb{R}^n$ then the following statements are true:-

- C^* is a closed convex cone.
- $C \subseteq C^{**}$ where C^{**} is a polar cone of C^*.
- $C_1 \subseteq C_2 \implies C_2^* \subseteq C_1^*$.

5 Blocking Polyhedra:

Blocking pairs of polyhedra are intimately related to maximum packing problems.
Anti-blocking pairs to minimum covering problems.

Subsets P of $\mathbb{R}^n$ of up-monotone if $x \in P$ and $y \geq x \implies y \in P$.

$$P^{\uparrow} = \{y \in \mathbb{R}^n | \exists x \in P : y \geq x\} = P + \mathbb{R}^n_+$$

Subsets P of $\mathbb{R}^n$ of down-monotone if $x \in P$ and $y \leq x \implies y \in P$.

$$P^{\downarrow} = \{u \in \mathbb{R}^n | \exists x \in P : y \leq x\}$$

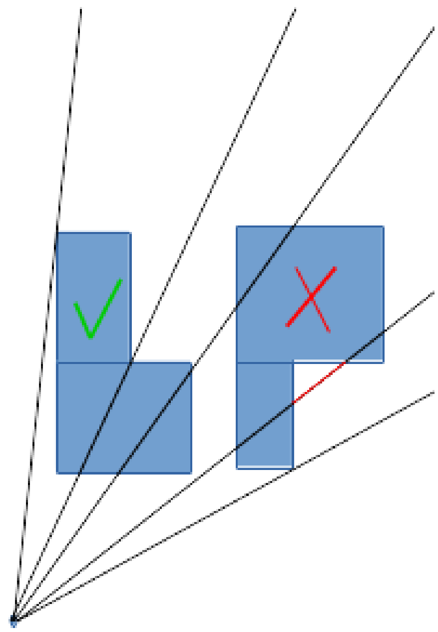

Figure 16: Example of monotone polyhedra

6 Anti-blocking Polyhedra:

Set $P \subseteq \mathbb{R}^n$ is of antiblocking type if P is a non-empty closed convex subsets of $\mathbb{R}^n$ that is down-monotone in $\mathbb{R}^n_+$.
P is polyhedron of anti-blocking type iff:-

$$P = \left\{x \in \mathbb{R}^n_+ | Ax \leq b\right\}$$

For any subsets P of $\mathbb{R}^n$, the anti-blocking set $A(P)$ of P is defined by :

$$A(P) = \left\{z \in \mathbb{R}^n_+ | x^T z \leq 1 \text{ for each } x \in P\right\}$$

6.1 Theorem:

Let $P \subseteq \mathbb{R}^n_+$ be antiblocking type. Then $A(P)$ is again of antiblocking type and $A(A(P)) = P$

$P = conv.hull \left\{x_1, x_2, ..., x_k\right\}^{\downarrow} \cap \mathbb{R}^n_+$ iff $A(P) = \left\{z \in \mathbb{R}^n_+ | x_i^T z \leq 1 \text{ for } i = 1, 2, ..., k\right\}$

7 Method for linear programming:

7.1 Simplex method:

It consist of both finding a path in the 1-seleton of the feasible region, ending at an optimum vertex.

It's in practice on avg. quite efficient, but no polynomial-time worst-case running time bound has been proved.

Niki holds two part-time jobs, Job I and Job II. She never wants to work more than a total of 12 hours a week. She has determined that for every hour she works at Job I, she needs 2 hours of preparation time, and for every hour she works at Job II, she needs one hour of preparation time, and she cannot spend more than 16 hours for preparation. If she makes $40 an hour at Job I, and $30 an hour at Job II, how many hours should she work per week at each job to maximize her income?

Solution

In solving this problem, we will follow the algorithm listed above.

STEP 1. Set up the problem. Write the objective function and the constraints.

Since the simplex method is used for problems that consist of many variables, it is not practical to use the variables x, y, z etc. We use symbols x_1, x_2, x_3, and so on.

Let

- x_1 = The number of hours per week Niki will work at Job I and
- x_2 = The number of hours per week Niki will work at Job II.

It is customary to choose the variable that is to be maximized as Z.

The problem is formulated the same way as we did in the last chapter.

$$\begin{array}{ll} \textbf{Maximize} & Z = 40x_1 + 30x_2 \\ \textbf{Subject to:} & x_1 + x_2 \le 12 \\ & 2x_1 + x_2 \le 16 \\ & x_1 \ge 0; x_2 \ge 0 \end{array}$$

STEP 2. Convert the inequalities into equations. This is done by adding one slack variable for each inequality.

For example to convert the inequality $x_1 + x_2 \le 12$ into an equation, we add a non-negative variable y_1, and we get

$$x_1 + x_2 + y_1 = 12$$

Figure 17:

Here the variable y_1 picks up the slack, and it represents the amount by which $x_1 + x_2$ falls short of 12. In this problem, if Niki works fewer than 12 hours, say 10, then y_1 is 2. Later when we read off the final solution from the simplex table, the values of the slack variables will identify the unused amounts.

We rewrite the objective function $Z = 40x_1 + 30x_2$ as $-40x_1 - 30x_2 + Z = 0$.

After adding the slack variables, our problem reads

$$\begin{array}{ll} \text{Objective function} & -40x_1 - 30x_2 + Z = 0 \\ \text{Subject to constraints:} & x_1 + x_2 + y_1 = 12 \\ & 2x_1 + x_2 + y_2 = 16 \\ & x1 \geq 0; x2 \geq 0 \end{array}$$

STEP 3. Construct the initial simplex tableau. Each inequality constraint appears in its own row. (The non-negativity constraints do *not* appear as rows in the simplex tableau.) Write the objective function as the bottom row.

Now that the inequalities are converted into equations, we can represent the problem into an augmented matrix called the initial simplex tableau as follows.

x1	x2	y1	y2	Z	C
1	1	1	0	0	12
2	1	0	1	0	16
–40	–30	0	0	1	0

Here the vertical line separates the left hand side of the equations from the right side. The horizontal line separates the constraints from the objective function. The right side of the equation is represented by the column C.

The reader needs to observe that the last four columns of this matrix look like the final matrix for the solution of a system of equations. If we arbitrarily choose $x_1 = 0$ and $x_2 = 0$, we get

$$\begin{bmatrix} y_1 & y_2 & Z & | & C \\ 1 & 0 & 0 & | & 12 \\ 0 & 1 & 0 & | & 16 \\ 0 & 0 & 1 & | & 0 \end{bmatrix}$$

which reads

Figure 18:

7.2 Solving linear programming problems: the Simplex Method

7.2.1 Standard form of the LP-problem

- All main constraints are equations.
- All variables are nonnegative.

In general, the standard form of an LP-problem can be written in the following form:

$$\begin{aligned} \max \quad & z = \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, \ldots, m \\ & x_j \geq 0, \quad j = 1, \ldots, n \end{aligned}$$

This can be written in the following matrix form:

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

where $\mathbf{x} = (x_1, \ldots, x_n)^T$, $\mathbf{c} = (c_1, \ldots, c_n)^T$, $\mathbf{b} = (b_1, \ldots, b_m)^T$, and $A = [a_{ij}]$ is an $(m \times n)$ matrix.

$$y_1 = 12 \quad y_2 = 16 \quad Z = 0$$

The solution obtained by arbitrarily assigning values to some variables and then solving for the remaining variables is called the **basic solution** associated with the tableau. So the above solution is the basic solution associated with the initial simplex tableau. We can label the basic solution variable in the right of the last column as shown in the table below.

x1	x2	y1	y2	Z		
1	1	1	0	0	12	y1
2	1	0	1	0	16	y2
–40	–30	0	0	1	0	Z

STEP 4. The most negative entry in the bottom row identifies the pivot column.

The most negative entry in the bottom row is -40; therefore the column 1 is identified.

x1	x2	y1	y2	Z		
1	1	1	0	0	12	y1
2	1	0	1	0	16	y2
–40	–30	0	0	1	0	Z
↑						

Question Why do we choose the most negative entry in the bottom row?

Answer The most negative entry in the bottom row represents the largest coefficient in the objective function - the coefficient whose entry will increase the value of the objective function the quickest.

The simplex method begins at a corner point where all the main variables, the variables that have symbols such as x_1, x_2, x_3 etc., are zero. It then moves from a corner point to the adjacent corner point always increasing the value of the objective function. In the case of the objective function $Z = 40x_1 + 30x2$, it will make more sense to increase the value of x_1 rather than x_2. The variable x_1 represents the number of hours per week Niki works at Job I. Since Job I pays \$40 per hour as opposed to Job II which pays only \$30, the variable x_1 will increase the objective function by \$40 for a unit of increase in the variable x_1.

STEP 5. Calculate the quotients. The smallest quotient identifies a row. The element in the intersection of the column identified in step 4 and the row identified in this step is identified as the pivot element.

Figure 19:

Following the algorithm, in order to calculate the quotient, we divide the entries in the far right column by the entries in column 1, excluding the entry in the bottom row.

x1	x2	y1	y2	Z			
1	1	1	0	0	12	y1	$12 \div 1 = 12$
$\boxed{2}$	1	0	1	0	16	y2	$\leftarrow 16 \div 2 = 8$
–40	–30	0	0	1	0	Z	
↑							

The smallest of the two quotients, 12 and 8, is 8. Therefore row 2 is identified. The intersection of column 1 and row 2 is the entry 2, which has been highlighted. This is our pivot element.

Figure 20:

7.2.2 How to convert an LP to standard form

1. Constraint $\sum a_{ij}x_j \leq b_i \quad \Rightarrow \quad \sum a_{ij}x_j + s_i = b_i, \quad s_i \geq 0$
2. Constraint $\sum a_{ij}x_j \geq b_i \quad \Rightarrow \quad \sum a_{ij}x_j - s_i = b_i, \quad s_i \geq 0$ where s_i is called a slack variable.
3. $x_j \leq 0 \quad \Rightarrow \quad x_j^{\text{new}} = -x_j \geq 0$
4. x_j is unrestricted in sign, i.e. x_j is a free variable $\quad \Rightarrow \quad x_j = x_j^+ - x_j^-$, where $x_j^+, x_j^- \geq 0$

7.3 Ellipsoid method:

In mathematical optimization, the ellipsoid method is an iterative method for minimizing convex functions over convex sets. The ellipsoid method generates a sequence of ellipsoids whose volume uniformly decreases at every step, thus enclosing a minimizer of a convex function.

When specialized to solving feasible linear optimization problems with rational data, the ellipsoid method is an algorithm which finds an optimal solution in a number of steps that is polynomial in the input size.

This method is very slow in practice.

It's decide polyhedron is empty or not.

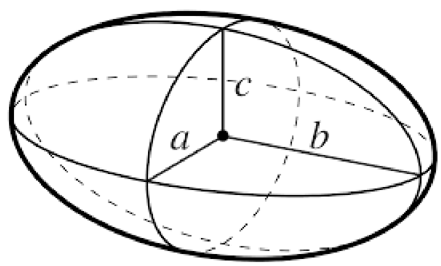

Figure 21: ellipsoid method

7.4 Numerical Example:

Unconstrained example: We consider the problem of minimizing a piecewise affine function:

$$\text{minimize } f(x) = \max_{i=1,2,\ldots,m}(a_i^T x + b_i)$$

with variable $x \in \mathbb{R}^n$. We use $n = 20$ variables and $m = 100$ terms , with problem data a_i and b_i generated from a unit normal distribution.

Basic ellipsoid algorithm: We use the basic ellipsoid algorithm described in $\S_1$ 1, starting with $P^{(0)} = I$, and $x^{(0)} = 0$.

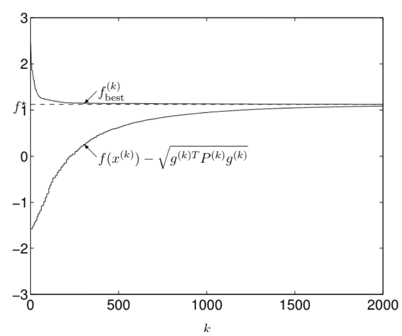

Figure 22: Convergence of $f_{\text{best}}^{(k)}$ and $f(x^{(k)}) - \sqrt{g^{(k)T}P^{(k)}g^{(k)}}$ (a lower bound on f^*) to f^* with the iteration number K.

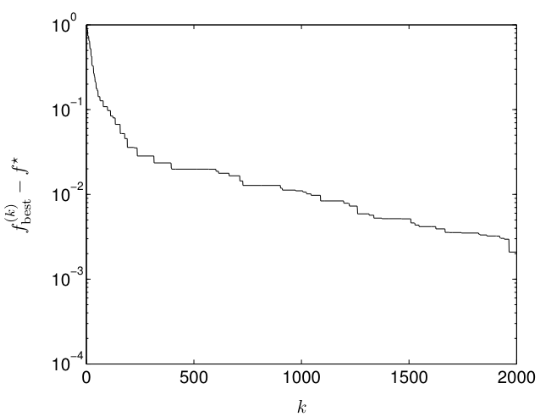

Figure 23: The suboptimality gap $f_{\text{best}}^{(k)}$ - f^* versus iteration k for the basic ellipsoid algorithm and the deep-cut ellipsoid algorithm

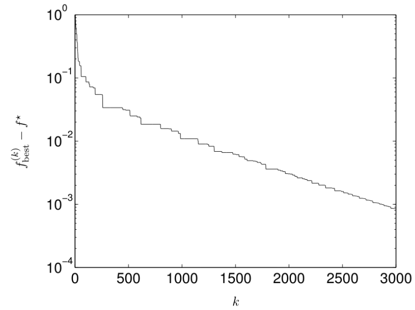

Figure 24: The suboptimality gap $f_{\rm best}^{(k)} - f^*$ versus iteration k for the constrained ellipsoid algorithm.

Constrained Example: Here we address a box-constrained version of:

$$\text{minimize } f(x) = \max_{i=1,2,\ldots,m}(a_i^T x + b_i),$$

$$\text{Subject to } |x_i| \leq 0.1,$$

with problem data generated as in the first example. We use the constrained ellipsoid method the solve the problem. Figure 21 shows the convergence of $f_{\rm best}^{(k)} - f^*$

8 Matroid:

In this chapter we assume that:

$$X + Y = X \cup Y$$

and

$$X - Y = X \backslash Y$$

8.1 Matroid definition:

A pair of $(S, \mathcal{I})$ is called *matroid*.

Where S is a finite set.
And $\mathcal{I}$ is a non-empty collection of subsets of S satisfying:-

1. if $I \in \mathcal{I}$ and $J \subseteq I,$ then $J \in \mathcal{I}$. (Hereditary Property)
2. if $I, J \in \mathcal{I}$ and $|I| < |J|, then I + z \in \mathcal{I}$ for some $z \in J \backslash I,$ (Exchange Property)

Let's discuss some of property:-

(I1) $\emptyset \in \mathcal{I}$ or $\mathcal{I} = \emptyset$.

(I2) If $A' \subseteq A$ and $A \in \mathcal{I}$ then $A' \in \mathcal{I}$.

(I3) If $A, B \in \mathcal{I}$ and $|A| < |B|$, then $\exists e \in (B \setminus A)$ such that $A \cup \{e\} \in \mathcal{I}$.

8.2 Matroids and Greedy Algorithm

Theorem *A set family $M = (E, \mathcal{I})$ is a matroid if and only if the greedy algorithm leads to a set Y in $\mathcal{I}$ of maximum weight $w(Y)$, for each weight function $w : E \to \mathbb{R}^{\geq 0}$.*

Greedy Algorithm for Activity Selection

Algorithm 1 Activity Selection

Input: A set of activities with start times $S = \{s_1, s_2, \ldots, s_n\}$ and finish times $F = \{f_1, f_2, \ldots, f_n\}$
Output: A maximum-size subset of mutually compatible activities
Sort the activities by their finish times so that $f_1 \leq f_2 \leq \ldots \leq f_n$
Let $A = \{1\}$ ▷ Select the first activity
$k \leftarrow 1$
for $m = 2$ to n **do**
  if $s_m \geq f_k$ **then**
    $A \leftarrow A \cup \{m\}$
    $k \leftarrow m$
  end if
end for
Return A

Note: Matroid M = $(S, \mathcal{I})$, a subset $\mathcal{I}$ of S is called independent if $I \in \mathcal{I}$ and dependent otherwise.

Question:-

Given a directed graph, $D(V, A)$, check whether D has arborescence.

Graph G is an arborescence[**?**] of root r, an r-arborescence, or just an arborescence iff:

1. Each vertex $v \neq r$, v has indegree exactly 1.
2. Indegree of r is 0.
3. For each $v \in V$ such that $v \neq r$ there is exactly one directed walk from r to v.

$M_1 = (E, \mathcal{I}_1)$ be the graphic matroid of underlying undirected graph. So E is the same set of edges as A ignoring direction.
$M_2 = (A, \mathcal{I}_2)$
$\mathcal{I}_2 = \{X \subset A | \text{if } (i_1, j_1) \in X \text{ and } (i_2, j_2) \in X \text{ and } j_1 \neq j_2\}$ i.e., no two edges in X are incident on the same vertex, i.e., the indegree of each vertex is at most one.

Claim: M_2 is a matroid

Proof. Property (I1) and (I2)holds trivially.
Suppose (I3) does not hold. Then there is $|A|, |B| \in \mathcal{I}_2$. $|A| < |B|$ then there is at least one vertex, v in B with indegree 1 that has indegree 0 in A. But then we can add the edge incident on vertex v to A i.e., A can be extended. So this holds (I3).

8.3 Base:

For the given $U \subseteq S$, a $B \subseteq U$ is called base of U if B is an inclusionwise maximal independent subsets of U.

Question:1 *Phrase bipartite matching problem as intersection of two matroids*

Let $M_1 = (E, \mathcal{I}_1)$ where
$I_1 = \{X \subset E \mid \text{if } (i_1, j_1) \in X \text{ and } (i_2, j_2) \in X \text{ and } i_1, i_2 \in A \text{ and } j_1, j_2 \in B \text{ and } i_1 \neq i_2\}$
that is $\forall x_i \in X$ every x_i are incident on different vertex on A and
$M_2 =$ (E, $\mathcal{I}_2$) where
$I_2 = \{X \subset E \mid \text{if } (i_1, j_1) \in X \text{ and } (i_2, j_2) \in X \text{ and } i_1, i_2 \in B \text{ and } j_1, j_2 \in A \text{ and } i_1 \neq i_2\}$

We want to show that set family M_1 and M_2 are matroids.

Proof. M_1 satisfies (I2), since if all the edges in X are incident on different vertices of A then all its subsets are also incident on different vertex of A.
Suppose (I3) does not hold. That is there is set X_1 and $X_2 \in \mathcal{I}$. $|X_1| < |X_2|$. For all edges e $\in X_2 \setminus X_1$. e is incident on a vertex v such that $\exists$ an edge $e' \in X_1 \setminus X_2$. e' is also incident on v. But since $|X_2| > |X_1|$ and all edges in X_2 are incident on differet vertices then $\exists$ atleast one vertex u such that there is no edge $(u, v) \in X_1$. Then we can extend X_1 using that edge, Hence (I3) holds.

Therefore M_1 is a matroid. Using the same reason M_2 is also a matroid. □

Therefore $M_1 \cap M_2$ gives the matching in the bipartite graph.

$B \in \mathcal{I}$ and there is no $\mathcal{Z} \in \mathcal{I}$ with $B \subset \mathcal{Z} \subseteq U$

8.4 Ranks:

The common size of the bases of a subset U of S is called rank of U.

It is represented by $r_m(U)$ or $r(U)$.
All bases of U have same-size, called rank of U.
Common size of all bases is called rank of matroid.

8.4.1 Spanning:

It contains a base as a subsets.

8.5 Circuit:

Matroid is an inclusionwise minimal dependent set.

8.5.1 Loop:

It is an element S such that $\{S\}$ in a circuit.

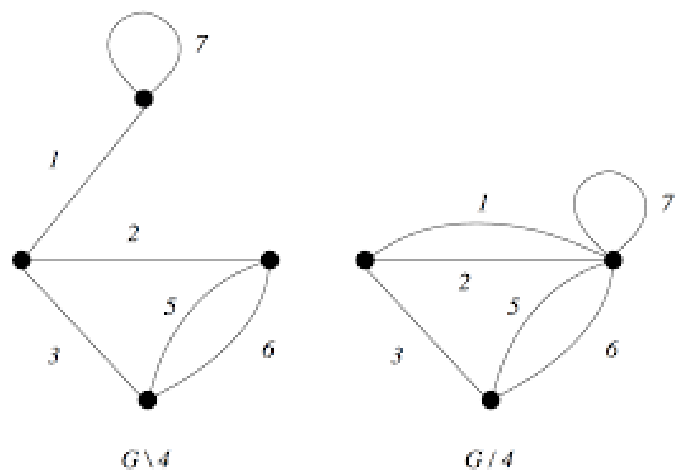

Figure 25: Example of Matroid

8.5.2 Parallel:

Two element s, t of S are called parallel if $\{s, t\}$ is a circuit.

8.5.3 Theorem:

Let S be a finite set and let $\mathcal{I}$ be a non-empty collection of subsets satisfying if $I \in \mathcal{I}$ and $J \subseteq I$, then $J \in \mathcal{I}$.

Then if $I, J \in \mathcal{I}$ and $|I| < |J|$, then $I + z \in \mathcal{I}$ for some $z \in J \backslash I$ is equivalent to:-
if $I, J \in \mathcal{I}$ and $|I \backslash J| = 1, |J \backslash I| = 2,$ then $I + z \in \mathcal{I}$ for some $z \in J \backslash I$

8.6 Dual matroid:

Each matroid M, a dual matroid M^* can be associated, in such a way that $(M^*)^* = M$. Let $M = (S, \mathcal{I})$ be a matroid, and define:-

$$\mathcal{I}^* = \{I \subseteq S | S \backslash I \text{ is a spanning set of } M\}$$

8.6.1 Theorem:

$M^* = (S, \mathcal{I}^*)$ is a matroid.

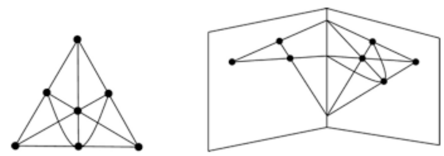

Figure 26: a matroid and it's dual

Some different stuff:-

- Circuit M^* are called cocircuit of M.
- M^* loops co-loops or bridge of M. Parallel elements of M^* are co-parallel or series of M.
- M is connected if $r_m(U) + r_m(S\backslash U) > r_m(S)$.

8.7 Deletion:

For $A \subseteq S$, the deletion $M\backslash A$ is the matroid on $S\backslash A$ with independent sets being $I \cap (S\backslash A)$ for $I \in \mathcal{I}$.

8.8 Contraction:

For $A \subseteq S$, the contraction $M\backslash A$ is defined on $S\backslash A$ with independent sets.

$$\{I \subseteq S|A : I \cup J \in \mathcal{I} \text{ for some } J \in \mathcal{I} \text{ for some independent } J \subseteq A\}$$

8.9 Truncation:

The truncation $T_K(M)$ of a matroid M reduces the rank of every subsets to a maximum of K.

Note: Restriction of M to Y $M|Y$

$$Y = S\backslash Z \text{ with } \mathcal{Z} \subseteq S \text{ and,}$$

$$S\backslash Z = \{x \in S|x \notin Z\}$$

$$Y = S\backslash Z \text{ then we denote } M \cdot Y = M\backslash Z$$

$$r_{M/Z}(X) = r(X \cup Z) - r(Z)$$

$$r_{M\backslash X/Y}(Z) = r_{M\backslash X}(Z \cup \{Y\}) - r_m(\{Y\}) = r_M(Z \cup \{Y\}) - r_M(\{Y\}) = r_{M/Y}(Z) = r_{M/Y\backslash X}$$

$M = (S, \mathcal{I})$ be a matroid and let K be a natural number($\mathbb{N}$) define :
$\mathcal{I}' = \{I \in \mathcal{I}|\,|I| \leq k\}$. Then $(S, \mathcal{I}')$ is again a matroid called K-truncation of MAtroid.

9 Some types of Matroid:

9.1 Uniform Matroids:

Matroid in which the independent sets are exactly the sets containing at most r elements, for some fixed integer r.

Note: Determined by a set S and a number K : the independent sets are the subsets I of S with $|I| \leq k$.

Trivially a matroid called a $K-$Uniform matroid by U_n^K, where $n = |S|$.

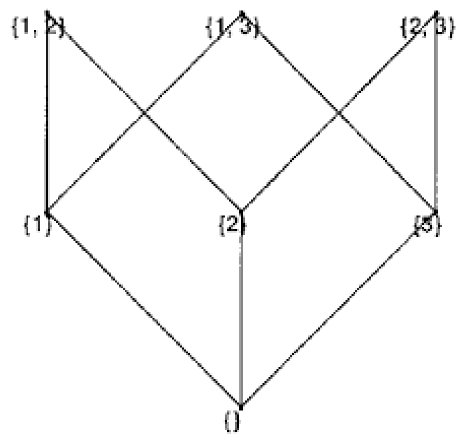

Figure 27: Uniform matroid

9.2 Linear Matroid:

A matroid is linear over $\mathbb{F}$ if $M \cong M(E, \mathbb{F})$ for some $E \subseteq \mathbb{F}^K$

Note: $\mathbb{F}$ be a field and let $E \subseteq \mathbb{F}^K$ be a finite set of vectors.

$$\mathcal{I} = \{F \subseteq E \,|\, \text{f is a linearli independent over } \mathbb{F}\}$$

$$M(E, \mathbb{F}) = (E, \mathcal{I}) \text{ is a linear matroid.}$$

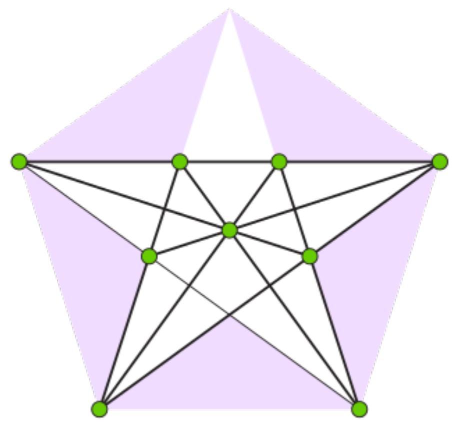

Figure 28: Linear Matroid

9.3 Binary Matroid:

A matroid representable over $GF(2)$- the field with two elements is called binary matroid.

A matroid M is binary iff for each choice of circuits

$$c_1, ..., c_t \text{ the set } c_1 \Delta ... \Delta c_t$$

can be partitioned into circuits.
Example: $U_4^2 \implies 2$ uniform matroid on 4 elements

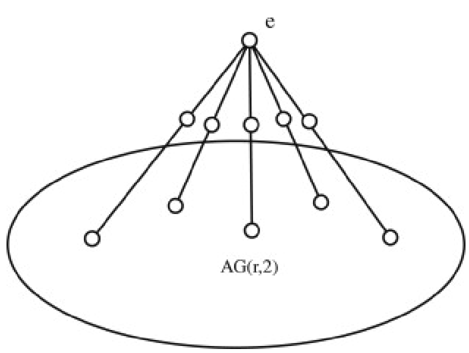

Figure 29: Binary Matroid

9.4 Regular matroid:

A regular matroid is representable over each field.

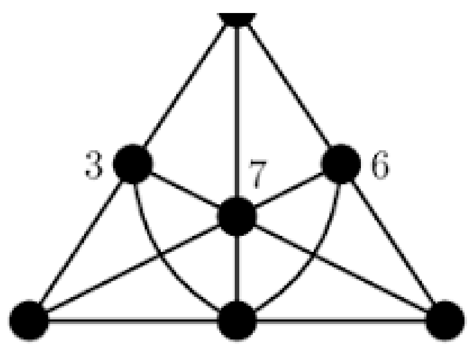

Figure 30: Regular Matroid

9.5 Algebraic matroid:

Let $\mathbb{K}$ be an extension field of $\mathbb{F}$. A set $\{x_1, x_2, ..., x_k\} \subseteq \mathbb{K}$ is algebraically dependent over $\mathbb{F}$ if there exists a polynomial p with coefficients in $\mathbb{F}$ such that $p(x_1, ..., x_k) = 0$.

Let $\mathbb{K}$ be an extension field of $\mathbb{F}$ and $E \subseteq \mathbb{K}$ be finite.

$$\mathcal{I} = \{f \subseteq E \,| f \text{ is algebraically independent over } \mathbb{F}\}$$

Then $M(E, \mathbb{F}) = (E, \mathcal{I})$ is an algebraic matroid.

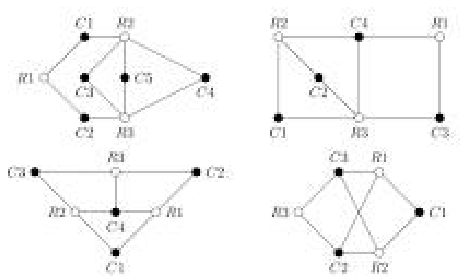

Figure 31: Algebraic Matroid

9.6 Graphical matroid:

Let $G = (V, E)$ be an unidirected graph and let

$$\mathcal{I} = \{f \subseteq E \,|(V, F) \text{ is a forest}\}$$

then $M(G) \cong M(H) \implies G \cong H$ for any graph H.

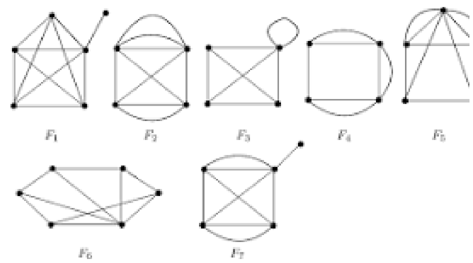

Figure 32: Graphical Matroid

9.6.1 Cographic matroids:

Derived from the dual of a graph's cyclic matroid.

9.7 Transversal matroid:

Let $A_1, ..., A_n$; $T \subseteq E$.
T is a partial transversal of $(A_1, ..., A_n)$ if $\exists$ an injective map $\phi : T \to \{1, 2, ..., n\}$ such that $t \in A_{\phi(t)}$ $\forall t \in T$.

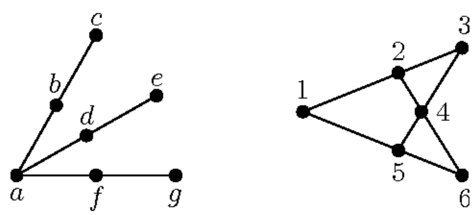

Figure 33: Transversal MAtroid

Note:
Given $A = (A_1, ..., A_n) \in (2^E)^n$, let

$$I = \{f \subseteq E \,|f \text{ is a partial transversal of } A\}$$

Then, $M(A) = (E, \mathcal{I})$ is transversal matroids.

Note: Universal matroids are transversal matroids.

$$M(k_4) \text{ isn't transversal}$$

______________End-Term______________

Thank you