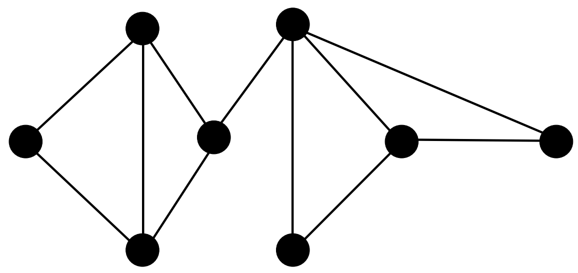
Partitioning

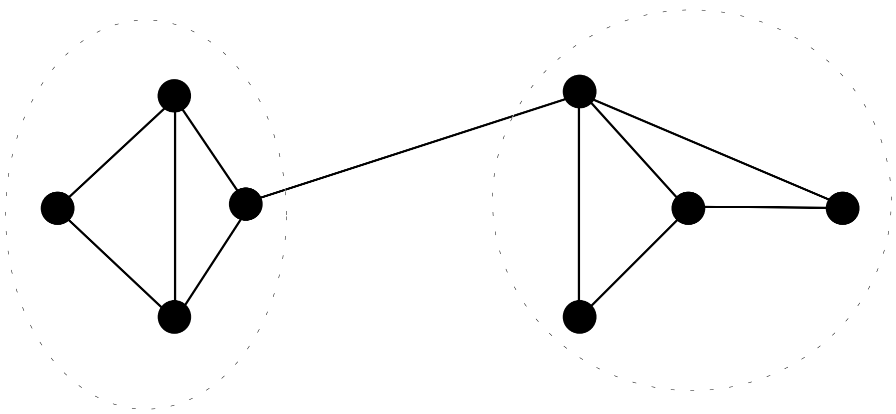
Somchai Prasitjutrakul

Partitioning Graphs

- **Problem definition :**
  Given a graph *G* with costs on its edges, partition the nodes of *G* into subsets no larger than a given maximum size, so as to minimize the total cost of the edges cut.
- **Applications :**
  assigning components of an electronic circuit on to printed circuit board or substrates, so as to minimize number of connections between cards or chips.
- **Exact solution :**
  complexity grows exponentially or factorially with the number of vertices !!! impractical !!!

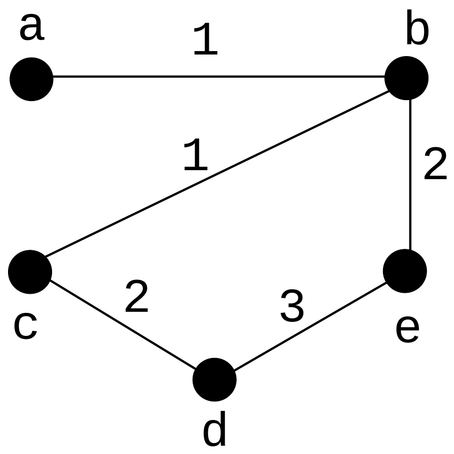
Partitioning Graphs





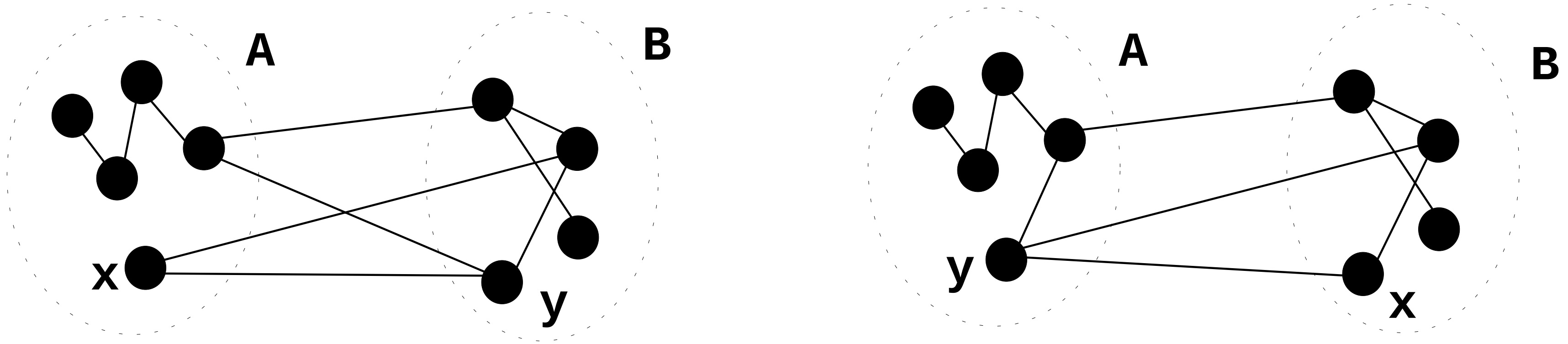
2-Way Partitioning Problem

- Let *S* be a set of *2n* vertices
- *C* be an associated cost matrix where c_{ij} is the cost of the edge connecting vertices *i* and *j*



	a	b	c	d	e
a	0	1	0	0	0
b	1	0	1	0	2
c	0	1	0	2	0
d	0	0	2	0	3
e	0	2	1	3	0

Gain



$$E_x = \sum_{b \in B} c_{xb} \ = \ 2 \qquad E_y = \sum_{a \in A} c_{ya} \ = \ 2$$

$$I_x = \sum_{a \in A} c_{xa} \ = \ 0 \qquad I_y = \sum_{b \in B} c_{yb} \ = \ 1$$

$$D_x = E_x - I_x \ = \ 2 \qquad D_y = E_y - I_y \ = \ 1$$

$$g_{xy} = D_x + D_y - 2c_{xy} \ = \ 2+1-2 \ = \ +1$$

Kernighan & Lin Algorithm

```
GraphPartitioning( A, B )
{
  A* = A;  B* = B
  do {
    Compute D values for all vertices
    for i=1 to n step +1 {
      select x_i ∈ A* and y_i ∈ B* such that g_i is maximum
      A* = A* - {x_i};  B* = B* - {y_i};
      update D values
    }
    choose k to maximize gain G = Σ_{i=1}^{k} g_i
    for i=1 to k step +1
      interchange x_i and y_i
  } until ( no gain is obtained )
}
```

$$G = \sum_{i=1}^{k} g_i$$

Maximize the Gain

- Construct a sequence of gains
- Find the *maximum* partial sum
- Example :

  squence of gains : 4,2,1,1,-2,3,-2,-1,-3,-3

  partial sum : 4,6,7,8, 6,**9**, 7, 6, 3, 0
- The process does not terminate immediately when some gain is negative

Running Time of the K&L Alg.

- speed up technique:
  - computation of the *D* values initially is an n^2 procedure
  - sort the *D* values :

    D_x : 5,4,3,1,1,1,0,0,0,0

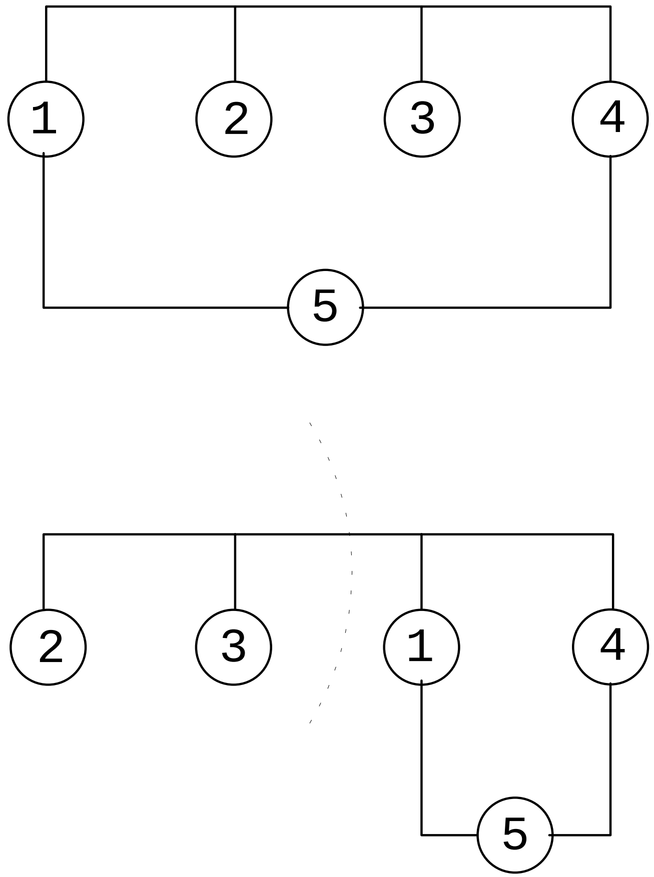
    D_y : 4,2,2,2,1,1,1,0,0,0

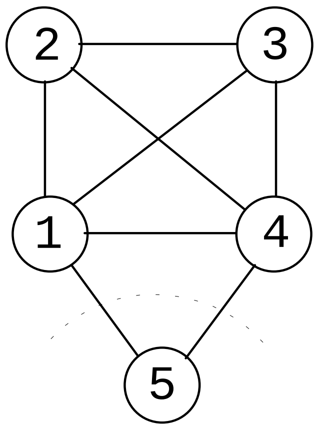
    - scanning down the set of *D*'s pairs,
    - if a pair of *D*'s is found whose sum does not exceed the maximum gain seen so far, then no bigger gain exists.
  - sorting is an $n \log n$ procedure, so the total time for sorting *D*'s in a pass is

    $n \log n + (n-1) \log (n-1) + \ldots + 2 \log 2 \;\underline{\approx}\; n^2 \log n$

Partitioning Electrical Circuits

- "net-cut" vs "edge-cut"

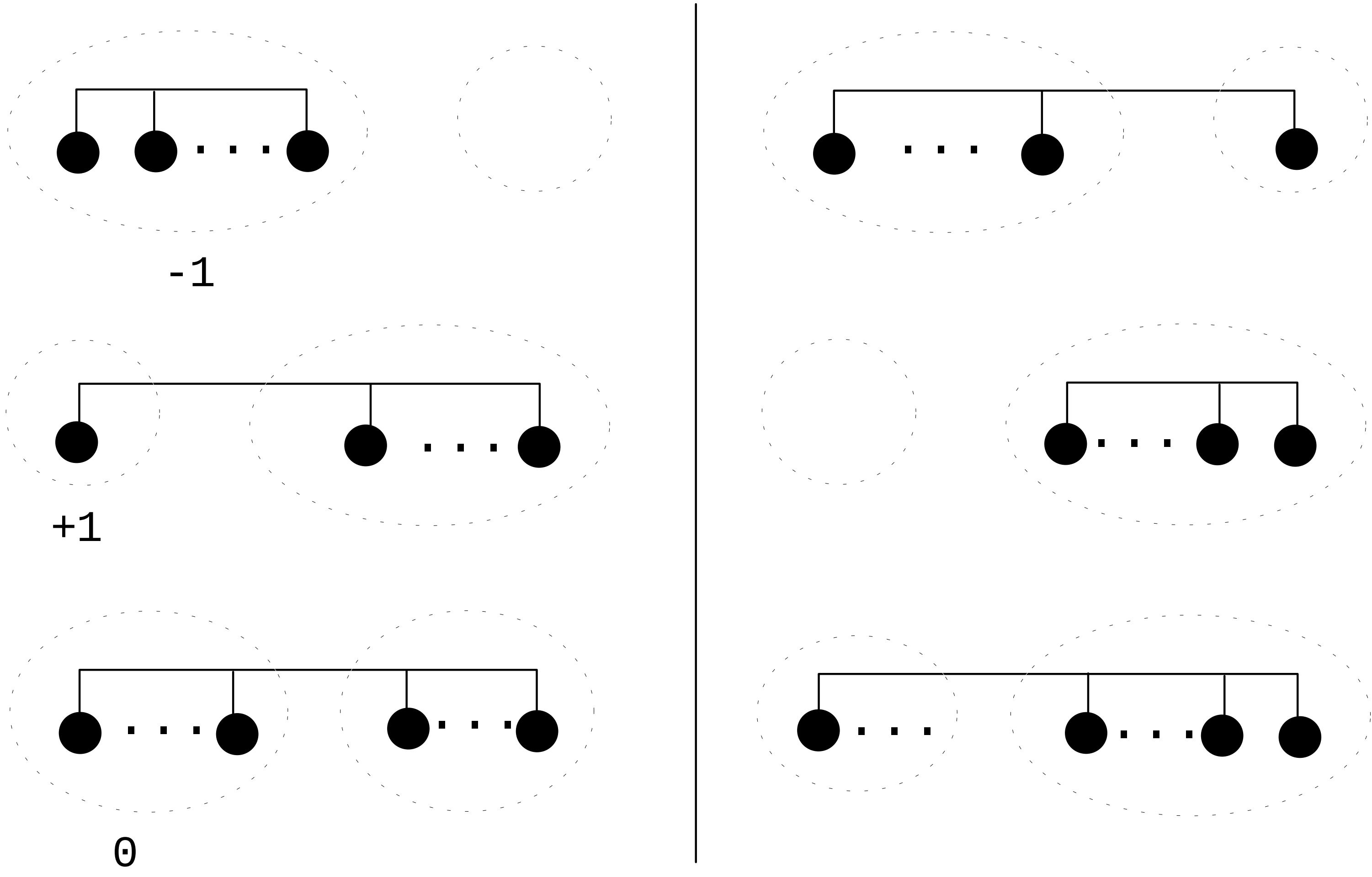




edge-cut model

net-cut model

Net-Cut Model



S. Prasitjutrakul 1992