Synchronizing parallel processes using threshold graphs

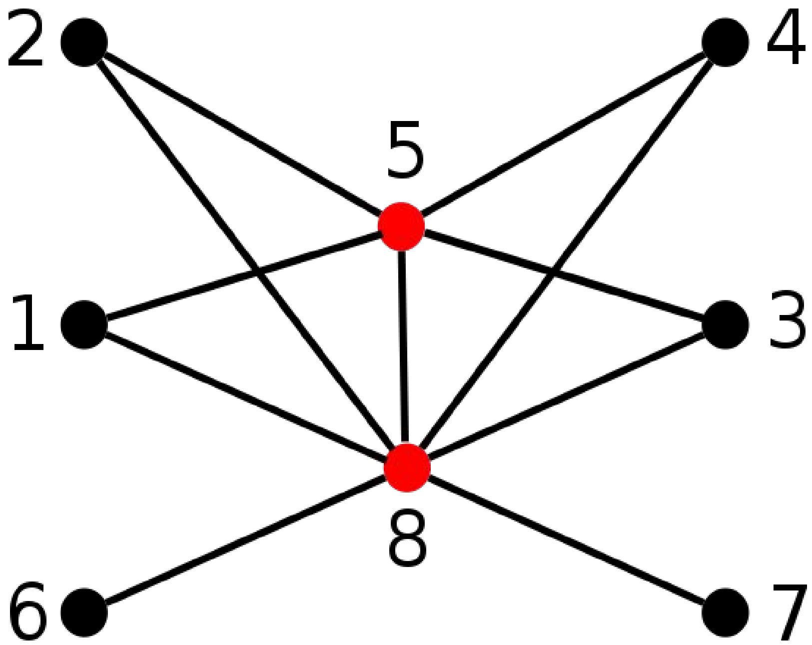
Presented by Christian D. Newman

Threshold Graphs

- Named due to their use of a ‘threshold’
  - Number used to differentiate between stable and unstable sets of nodes
  - Determined using node/edge weights
- Hypergraphs and hyperedges
  - Discussion stays low level but this underlies the concept

Example

- Numbering is order of insertion
- Black nodes are isolated vertices
- Red nodes are dominating vertices



Threshold Dimension

- Minimum number k of linear inequalities such that
  - We can separate stable set from non-stable set
- Abstract characteristic that extends to hypergraphs
  - For simplicity, we assume threshold dimension = 1
  - I will provide short discussion of when it is >1

Node label and threshold

- Nodes can be labeled in a variety of ways. For example, the label could be their degree. Threshold is defined using the labels:

$$X \text{ is stable} \Leftrightarrow \sum_{x \in X} a(x) \le t \qquad (X \subseteq V).$$

- Where a(x) is a label for x, X is a set of nodes in V. General labeling is np-complete.

Building Threshold Graph

- Threshold graph can be built by repeating the following:
    - Add a single, isolated vertex to the graph or
        - Isolated vertex is connected to no other vertex
    - Add a single, dominating vertex to the graph
        - Dominating vertex is a vertex connected to all others currently in the graph

Special Properties

- There exists a threshold, t, after which a stable set becomes unstable
- We can determine whether a node is stable or unstable with a single inequality (assuming threshold dimension <=1)

Problem 1

- Have a set of programs, P, to be run in parallel.
- Memory constraints cause conflict to arise when certain subsets of P run simultaneously
- E is the collection of all such forbidden P

Problem 1 (cont)

- Given the previous, a set of programs X in P can be run if and only if X contains no member of E
- Assuming (P, E) is a hypergraph, then we can do the following:

Solution

- Call P(s,ci) before each program Pi
- Call V(s,ci) after each program Pi
- Initialize a new global variable s with value t

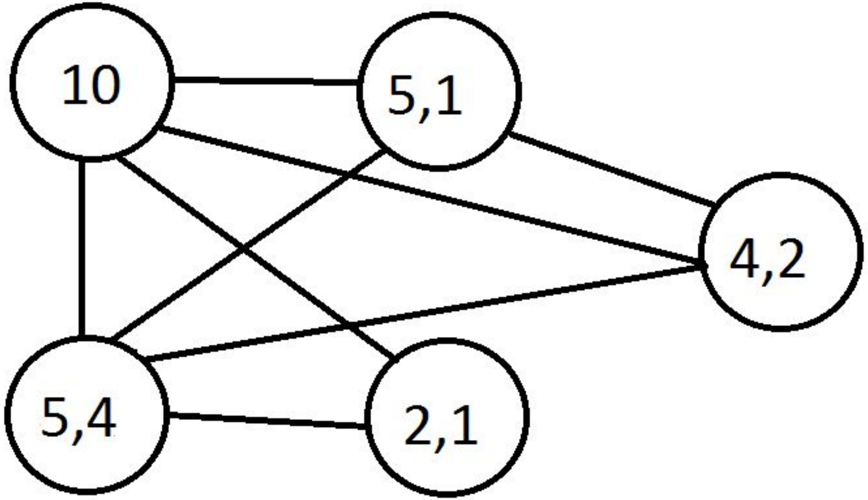
procedure $P(s, c)$:
  if $s \geq c$ **then**
    $s \leftarrow s - c$
    enter P_i
  else
    call again
return

procedure $V(s, c)$:
  $s \leftarrow s + c$
return

Example Graph

Processes: 10,5,4,2,1. Each number is the weight (memory) that the corresponding process requires.

t = 10



Discussion

- ‘s’ is a semaphore
  - Never allows sum of C_i to exceed t since it forces the program to call the procedure again when C_i is greater than or equal to s.
- We will look at a rough graph of the situation

Problem 2

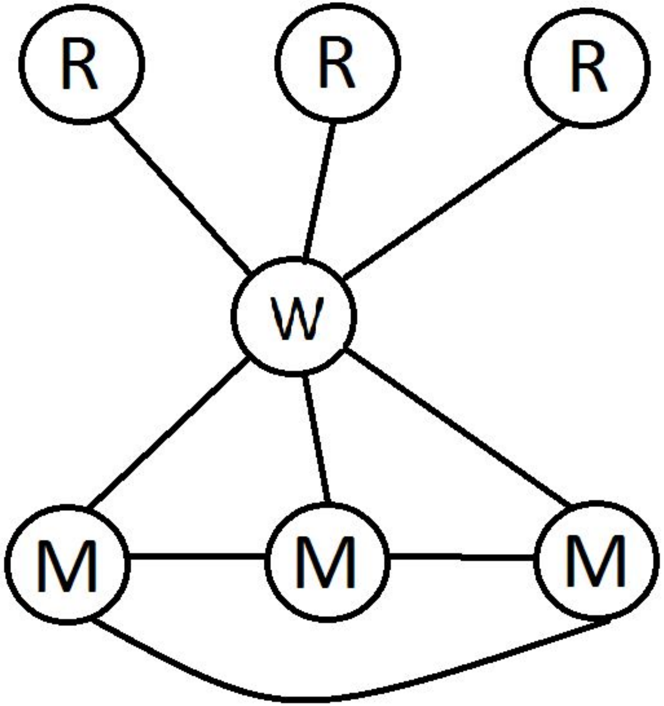
- Set of Mathematicians, readers, and writers.
- Can execute most one mathematician and an unlimited number of readers or at most one writer:

$$
\begin{aligned}
c(R_i) &= 1 && (i = 1, \ldots, r),\\
c(M_j) &= r + 1 && (j = 1, \ldots, m),\\
c(W_k) &= 2r + 1 && (k = 1, \ldots, w),\\
t &= 2r + 1.
\end{aligned}
$$

Discussion

- We can execute any number of readers at any time.
- Mathematicians need a calculator and there is only one.
- Once something is being written, no one else can access

Example Graph



$$
\begin{aligned}
c(R_i) &= 1 && (i = 1, \ldots, r),\\
c(M_j) &= r + 1 && (j = 1, \ldots, m),\\
c(W_k) &= 2r + 1 && (k = 1, \ldots, w),\\
t &= 2r + 1.
\end{aligned}
$$

Conclusion

- Threshold graphs allow us to specify a threshold based on node weights
  - Perfect for synchronization problem where node weight is:
    - # of resources acquired or
    - Amount of a limited resource
    - Or both

References

- https://en.wikipedia.org/wiki/Threshold_graph
- Algorithmic Graph Theory and Perfect Graphs (Annals of Discrete Mathematics, Vol 57)

Hypergraphs

Let $V = \{v_1, v_2, \ldots, v_n\}$ be the vertex set of an undirected graph G. Any subset $X \subseteq V$ can be represented by its *characteristic vector* $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, where for all i

$$x_i = \begin{cases} 1 & \text{if} \quad v_i \in X, \\ 0 & \text{if} \quad v_i \notin X. \end{cases}$$

Hypergraph

The *threshold dimension* $\theta(G)$ of the graph $G = (V, E)$ is defined to be the minimum number k of linear inequalities

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\le t_1, \\ &\vdots \\ a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n &\le t_k, \end{aligned} \tag{1}$$

such that X is a stable set if and only if its characteristic vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ satisfies (1). Regarding each inequality of (1) as a hyperplane in n-space, X is stable iff $\mathbf{x}$ lies on or within the "good" side of each of those k hyperplanes. Since G is finite, $\theta(G)$ is finite and well defined.