Hu’s Algorithm for Multiprocessor Scheduling

Dr. Chandan Karfa

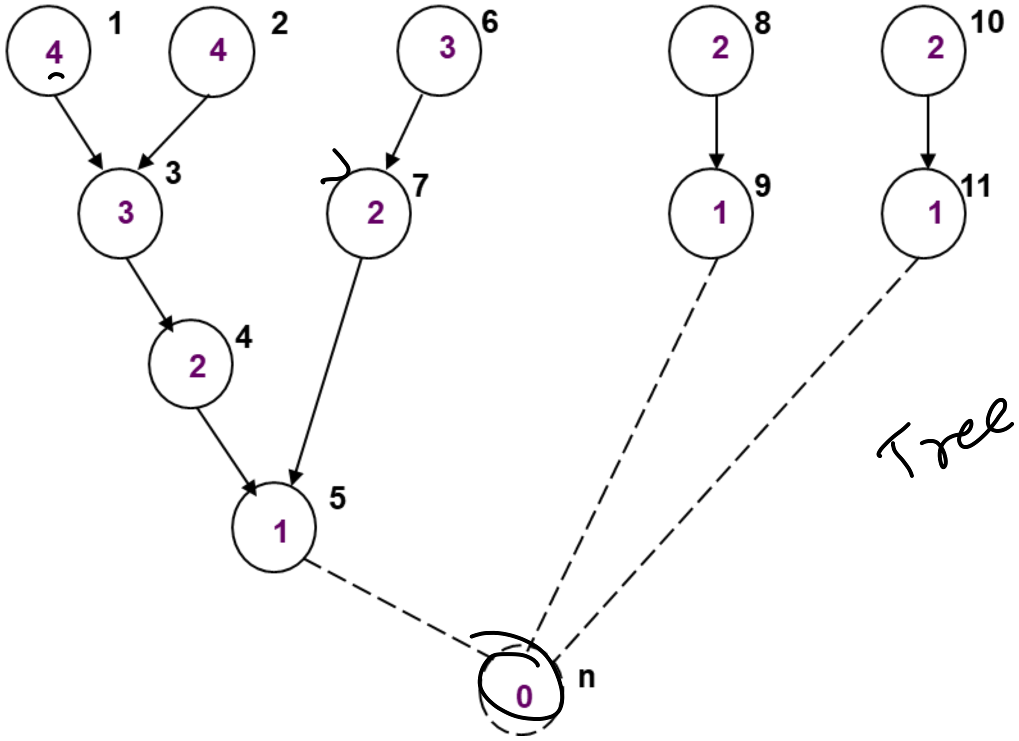
Department of Computer Science and Engineering

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

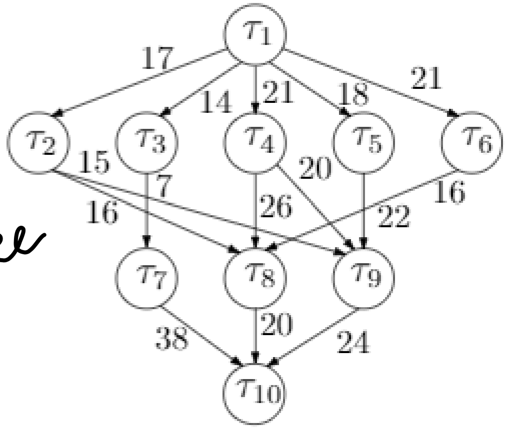
Multiprocessor Scheduling and Hu's algorithm

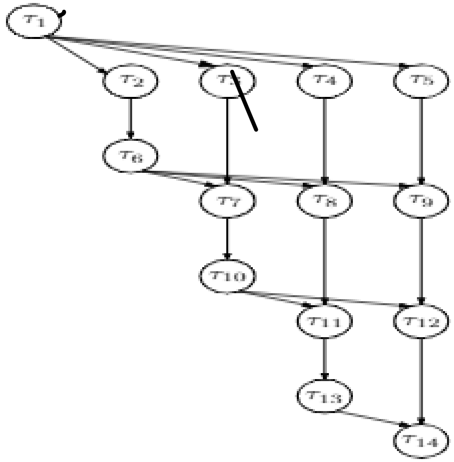
- Assumptions:
  - A1: All operations have the same type – a multiprocessor is executing the operations
  - A2: All operations have unit delay
  - A3: Sequence graph is Tree
    - Implications: There is no parallel paths.
    - have single paths from each vertex to the sink with monotonically unit-wise decreasing labels (path length from the sink node)
- The problem is in P with A1, A2 and A3
  - Greedy strategy
  - Exact solution



Tree

Not a tree





Multiprocessor Scheduling and Hu's algorithm

- Assumptions:
  - A1: All operations have the same type – a multiprocessor is executing the operations
  - A2: All operations have unit delay
  - A3: Sequence graph is Tree
    - Implications: There is no parallel paths.
    - have single paths from each vertex to the sink with monotonically unit-wise decreasing labels (path length from the sink node)
- The problem is in P with A1, A2 and A3
  - Greedy strategy
  - Exact solution

Labeling of Nodes

- A labeling of a sequencing graph consists of marking each vertex with the weight of its longest path to the sink, measured in terms of edges
- Let us denote the labels by (α_i; i = 1,2,. . . , *n)*

$$\alpha = \max_{1 \le i \le n} \alpha_i .$$

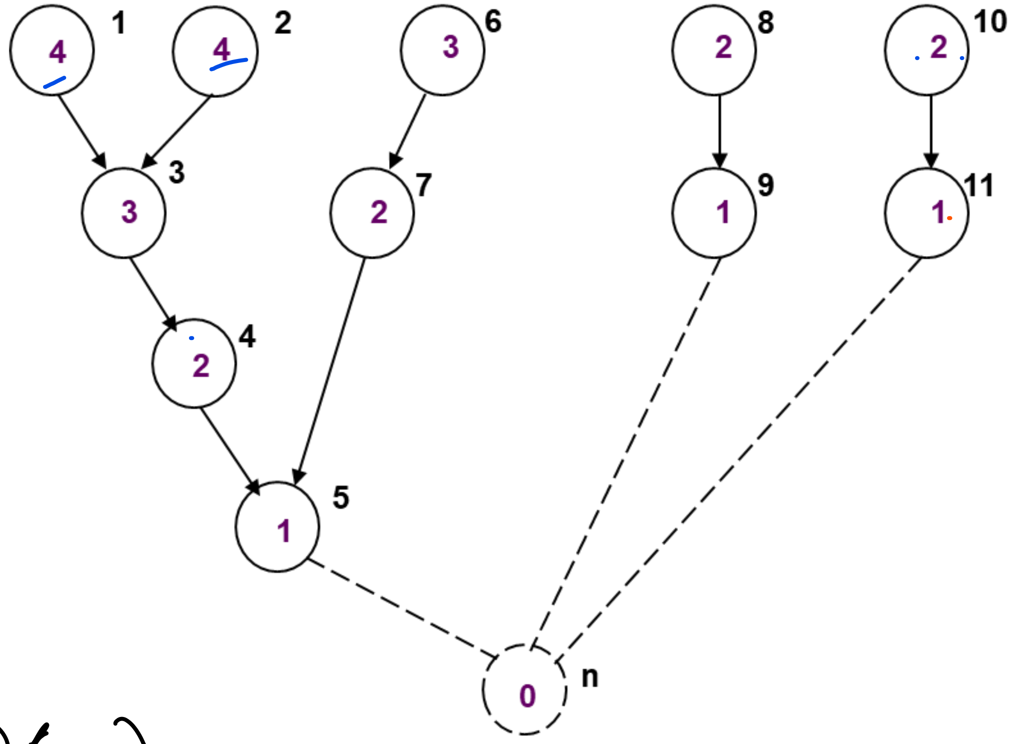
- p(j) = number of vertices with label equal to j, i.e.,

$$p(j) = |\{v_i \in V : \alpha_i = j\}|.$$

- Latency is greater than or equal to the weight of the longest path

Example

MLRC



Max No of any node

$\alpha = 4 \leq \lambda$

$\lambda \geq 4$. Also $p(0) = 1$, $p(1) = 3$, $p(2) = 4$, $p(3) = 2$, $p(4) = 2$.

Hu's Algorithm with ā resources

- Label operations with distance to sink
- Set step l = 1
- Repeat until all ops are scheduled:
  - Select $s \leq \bar{a}$ resources with
    - All predecessors scheduled
    - Maximal labels
  - Schedule the s operations at step l
  - Increment step $l = l + 1$

Hu's Algorithm (MLRC)

HU ($G(V,E)$, a) {

Label the vertices // a = resource constraint (scalar)

// *label* = length of longest path passing through the vertex

$l = 1$

repeat {

U = unscheduled vertices in V whose predecessors have been already scheduled (or have no predecessors)

available

Select $S \subseteq U$ such that $|S| \leq a$ and labels in S are *maximal*

Schedule the S operations at step l by setting $\boxed{t_i = l,}\ \forall v_i \in S$;

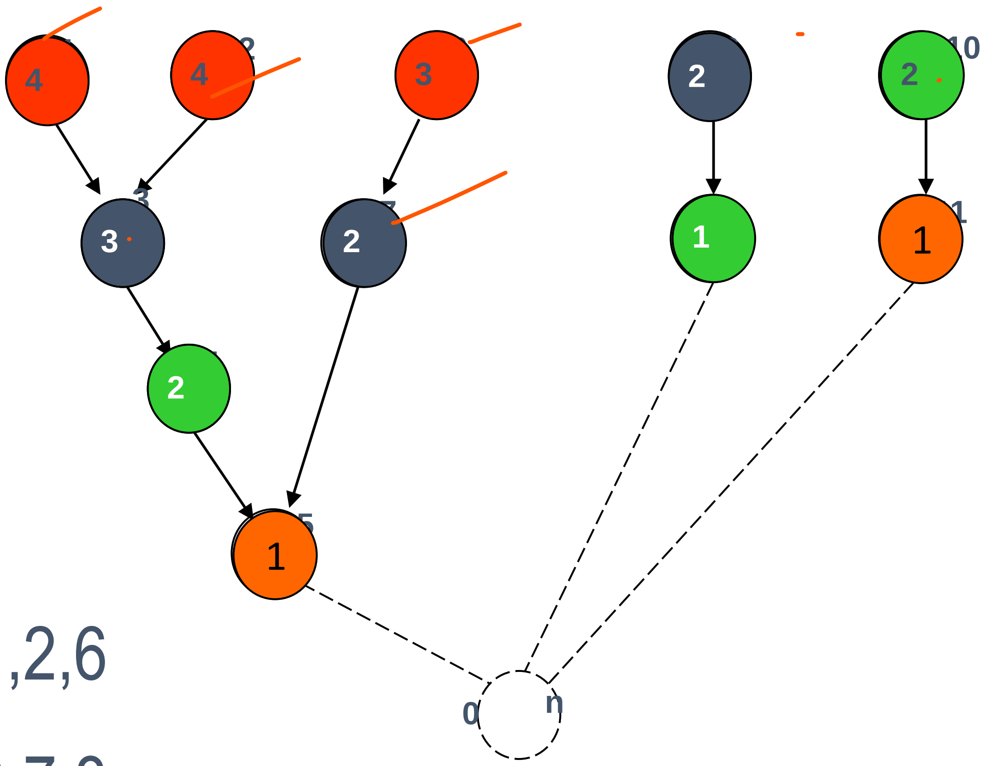
$l = l + 1$;

} until v_n is scheduled.

}

Example

$\bar{a} = 3$



Step 1: Op 1,2,6

Step 2: Op 3,7,8

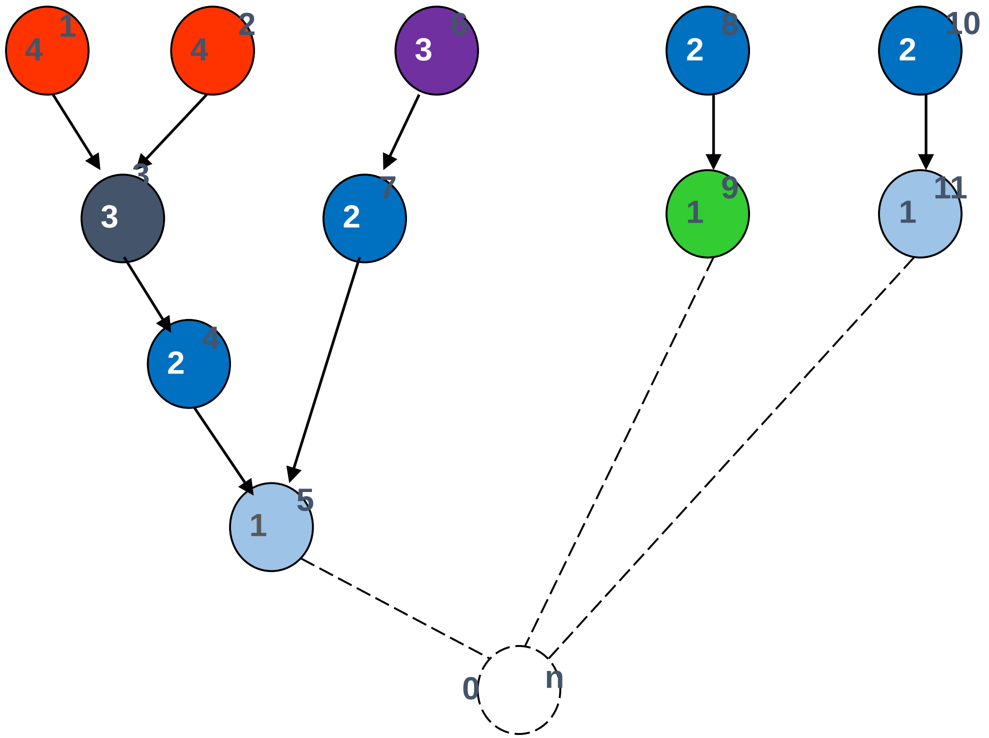
Step 3: Op 4,9,10

Step 4: Op 5,11

Exactness of Hu's algorithm

- The idea in Hu's algorithm is very simple and intuitive, its power lies in the fact that the computed solution is an optimum one.
- We show first that the algorithm can always achieve the latency λ with ā resources, where ā is a bound and is a function of λ.
- Since the number of resources used by the algorithm is equal to the lower bound for the problem, the algorithm achieves an optimum schedule with minimum resource usage under latency constraints (MRLC).
- As a consequence, the algorithm achieves also the minimum-latency schedule under resource constraints (MLRC).
- Thus, the proof of exactness is in two steps:
  - determine that ā is a lower bound for the problem
  - Show that the algorithm always reaches ā

Example



α = 4

p (4) = 2

p (3) = 2

p (2) =4

p (1) =3

Lower bound of Resource for a given Latency

- **Theorem1:**
  - Given a dag with operations of the same type

$$\bar{a} = \max_{\gamma} \left\lceil \frac{\sum_{j=1}^{\gamma} p(\alpha + 1 - j)}{\gamma + \lambda - \alpha} \right\rceil$$

  - $\bar{a}$ is a lower bound on the number of resources to complete a schedule with latency λ
  - γ is a positive integer
- Theorem2:
  - Hu's algorithm applied to a tree with $\bar{a}$ unit-cycle resources achieves latency λ
- Corollary:
  - Since $\bar{a}$ is a lower bound on the number of resources for achieving λ, then λ is minimum

Lower bound of Resource

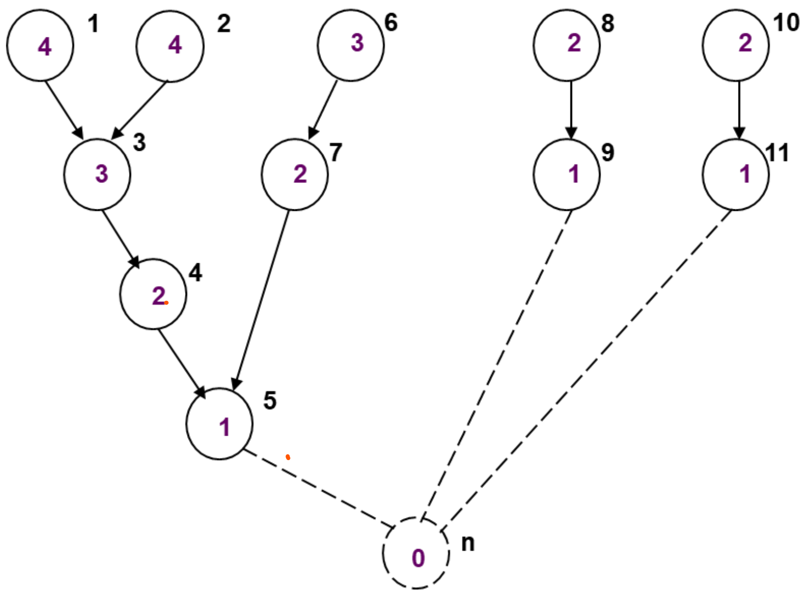
- Theorem 1:

$$\bar{a} = \max_{\gamma} \left\lceil \frac{\sum_{j=1}^{\gamma} p(\alpha + 1 - j)}{\gamma + \lambda - \alpha} \right\rceil$$

$$\bar{a} = \left\lceil \max \left\{ \frac{p(4)}{1}, \frac{p(4)+p(3)}{2}, \frac{p(4)+p(3)+p(2)}{3}, \frac{p(4)+p(3)+p(2)+p(1)}{4}, \frac{p(4)+p(3)+p(2)+p(1)+p(0)}{5} \right\} \right\rceil$$

$$= \lceil \max\{2, 2, 8/3, 11/4, 12/5\} \rceil$$

$$= 3$$



$\alpha = 4$

$p(4) = 2$

$p(3) = 2$

$p(2) = 4$

$p(1) = 3$

Lower bound of Resource

- **Theorem 1: A lower bound on the number of resources to complete a schedule with latency λ is (γ is an integer)**

$$\bar{a} = \max_{\gamma} \left\lceil \frac{\sum_{j=1}^{\gamma} p(\alpha + 1 - j)}{\gamma + \lambda - \alpha} \right\rceil$$

Proof:

$$\gamma^* = \arg\max \left\lceil \frac{\sum_{j=1}^{\gamma} p(\alpha + 1 - j)}{\gamma + \lambda - \alpha} \right\rceil$$

- For the sake of contradiction, let us assume that a schedule exists with a resources that satisfy the latency bound a

Exactness of Hu's algorithm

$$a < \left\lceil \frac{\sum_{j=1}^{\gamma^*} p(\alpha + 1 - j)}{\gamma^* + \lambda - \alpha} \right\rceil$$

- The vertices scheduled up to step *l* cannot exceed a. l.
- For $l = \gamma^* + \lambda - \alpha$

$$a \cdot (\gamma^* + \lambda - \alpha) < \sum_{j=1}^{\gamma^*} p(\alpha + 1 - j)$$

- This implies that at least a vertex with label $\alpha_i = p(\alpha + 1 - \gamma^*)$ has not been scheduled yet.

Exactness of Hu's algorithm

- Therefore, to schedule the remaining portion of the graph we need at least ($\alpha + 1 - \gamma^*$) steps.
- Thus, the schedule length is at least

$$\gamma^* + \bar{\lambda} - \alpha + \alpha + 1 - \gamma^* \;=\; \lambda + 1$$

- which contradicts our hypothesis of satisfying a latency bound of λ.
- Therefore, at least a resources are required $\bar{a}$

$a < \bar{a}$

Exactness of Hu’s algorithm

- Theorem 2: Hu's algorithm achieves latency λ with as many resources as:

$$\overline{a} = \max_{\gamma} \left\lceil \frac{\sum_{j=1}^{\gamma} p(\alpha + 1 - j)}{\gamma + \lambda - \alpha} \right\rceil$$

- Corollary: Let λ be the latency of the schedule computed by Hu's algorithm with *a* resources. Then, any schedule with *a* resources has latency larger than or equal to A.

Importance of Hu's algorithm

- Hu's algorithm is very simple and intuitive, its power lies in the fact that the computed solution is an optimum one.

- Since the number of resources used by the algorithm is equal to the lower hound for the problem, the algorithm achieves an optimum schedule with minimum resource usage under latency constraints (MRLC)

- The algorithm achieves also the minimum-latency schedule under resource constraints (MLRC)
- List based Heuristic algorithms are based on HU's ideas.

Thank You