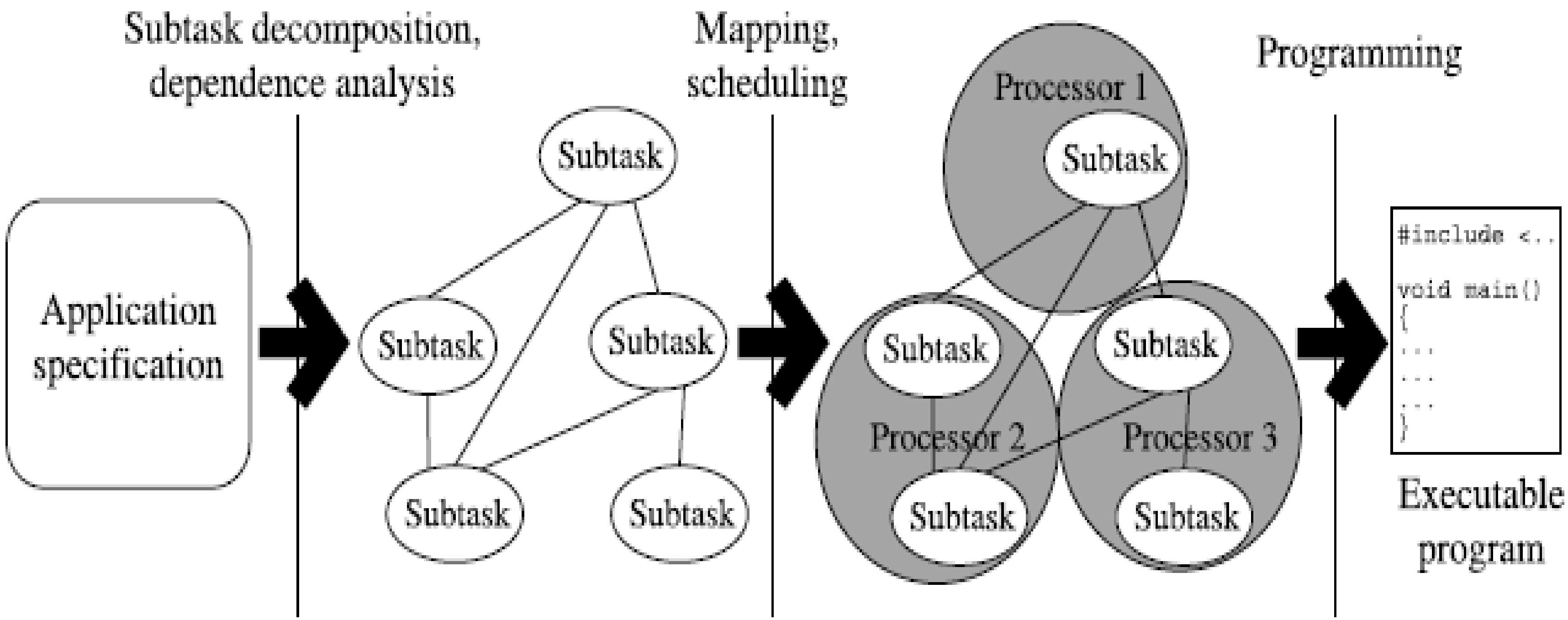
Task Scheduling

Chapter 5

Parallel programming—process of parallelization



The Scheduling Problem

Scheduling model

– The schedule

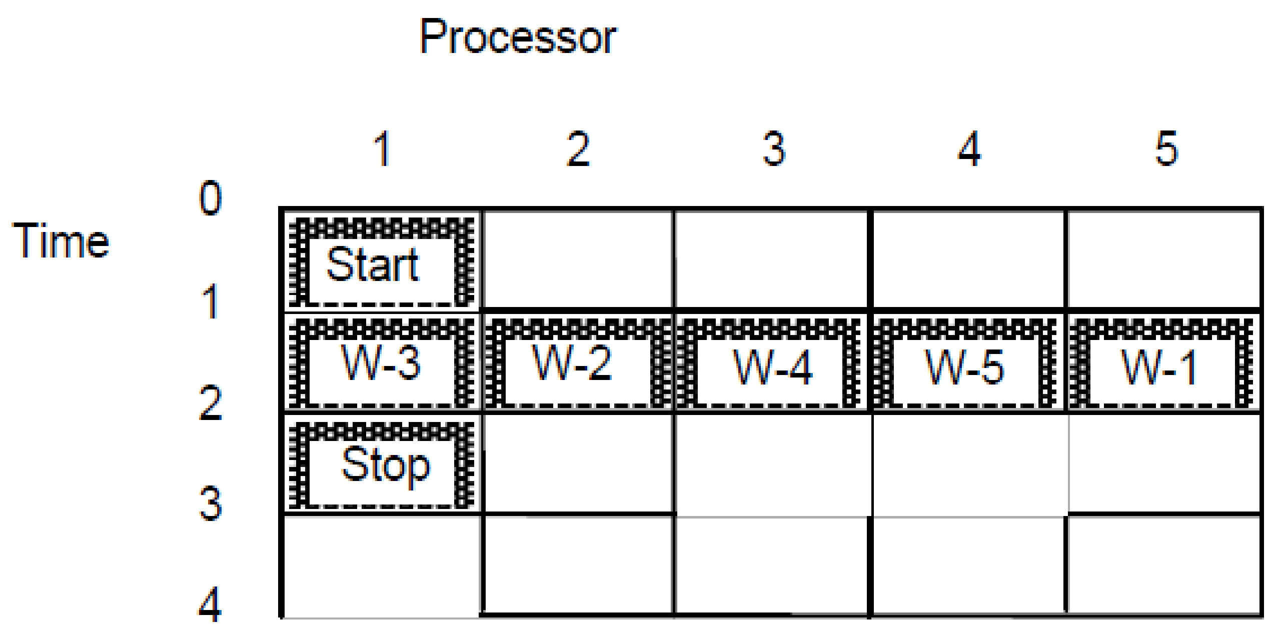
- Gantt Chart
- Mapping (f) of tasks to a processing element and a starting time.
- Formally:

– f: T ⍰ {1,2,3, …, m} x [0,infinity]

– f(v) = (i,t) task v is scheduled to be processed by processor I starting at time t

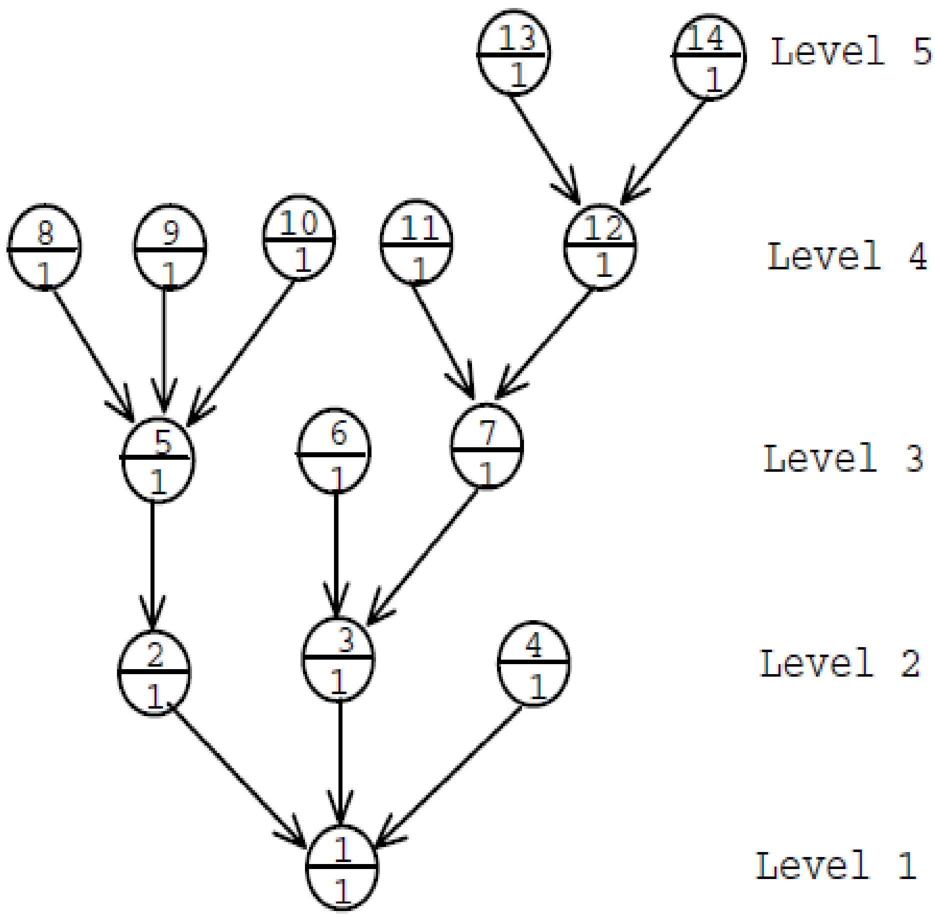
Scheduling model

– Gantt Chart:



- Scheduling in-forests/out-forests task graphs
  - The level of each node in the task graph is calculated as given above and used as each node's priority.
  - Whenever a processor becomes available, assign it the unexecuted ready task with the highest priority.

- Scheduling in-forests/out-forests task graphs

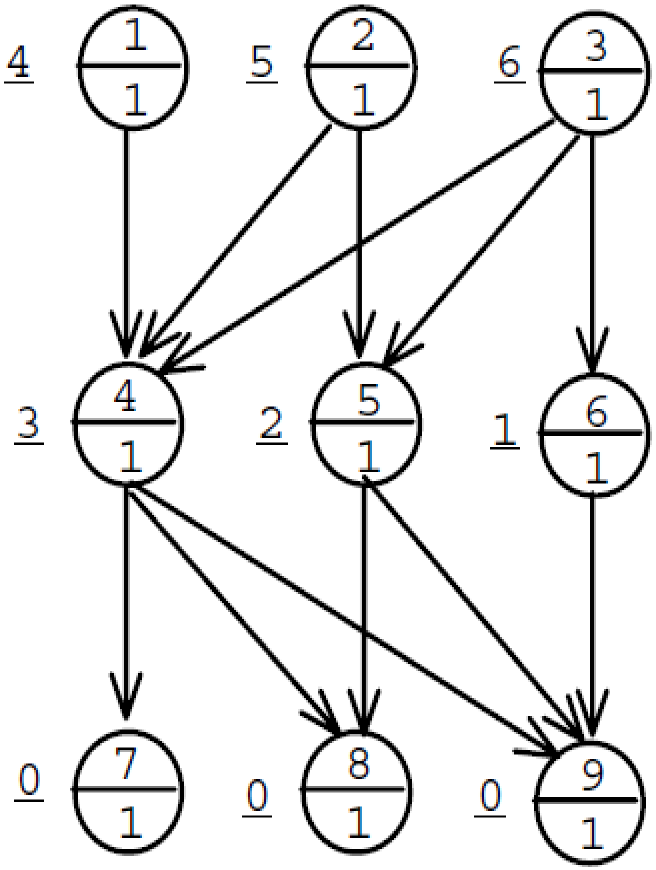


P1	P2	P3
14	13	11
12	10	9
8	7	6
5	4	3
2		
1		

Scheduling interval ordered tasks

- A task graph is an interval order when its nodes can be mapped into intervals on the real line, and two elements are related iff the corresponding intervals do not overlap.
- For any interval ordered pair of nodes u and v, either the successors of u are also successors of v or the successors of v are also successors of u.

Scheduling interval ordered tasks

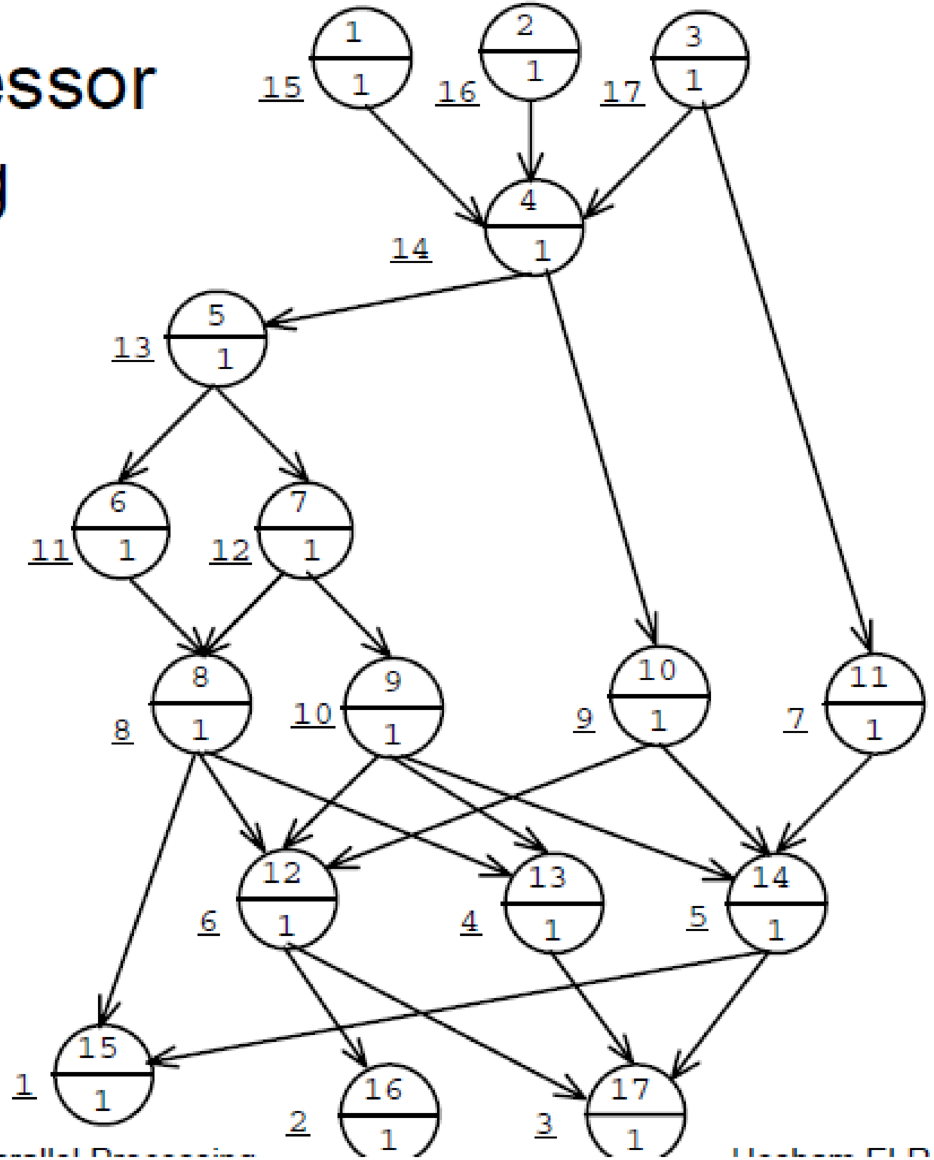


Time	P1	P2	P3
0	3	2	1
1	4	5	6
2	7	8	9
3			

Two-processor scheduling

- Use $L(v)$ as the priority of task v and ties are broken arbitrary.
- Whenever a processor becomes available, assign it the unexecuted ready task with the highest priority. Ties are broken arbitrarily.

Two-processor scheduling



time ↓

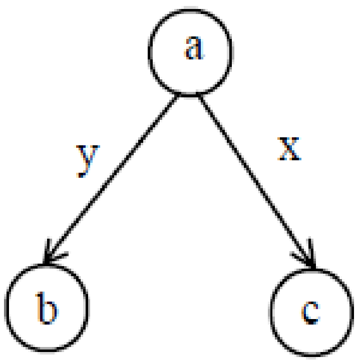
P1	P2
3	2
1	11
4	
5	10
7	6
9	8
12	14
13	16
17	15

Communication Models

Completion time as two components

- Completion Time = Execution Time + Total Communication Delay
- Total Communication Delay = Number of communication messages * delay per message
- Execution time → maximum finishing time of any task

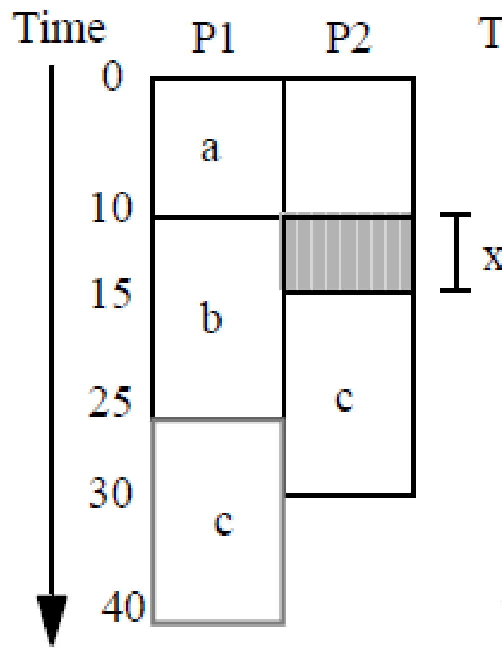
Parallelism versus communication delay



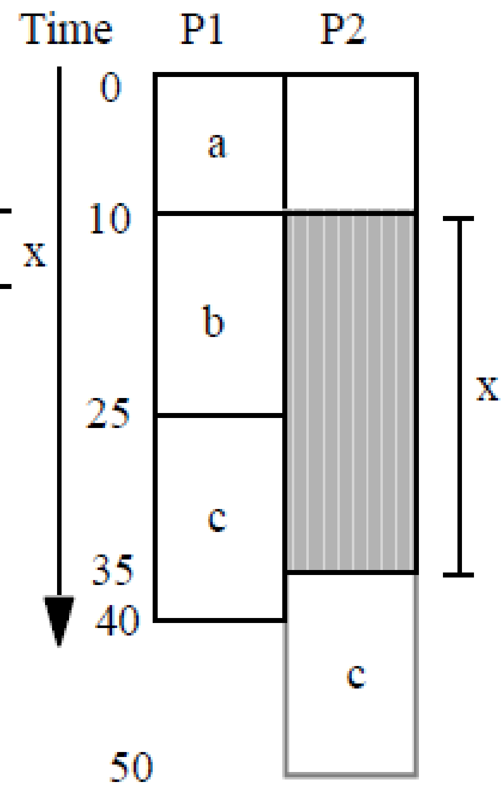
Task Graph

Task	Exceution time
a	10
b	15
c	15

Arc	Communication
(a,b)	y
(a,c)	x < y



Gantt Chart-1
x = 5

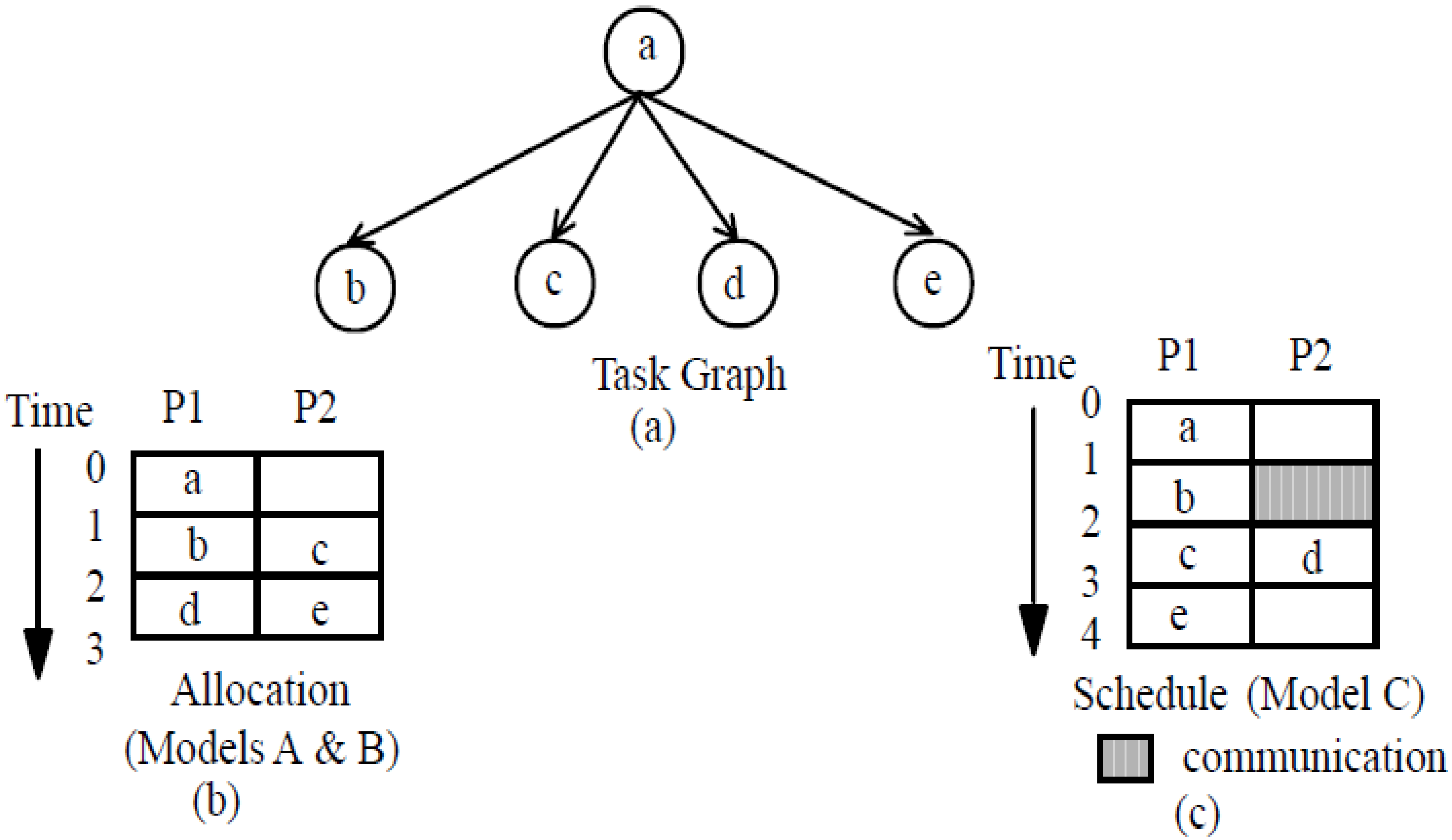


Gantt Chart-2
x = 25

Completion time from the Gantt chart

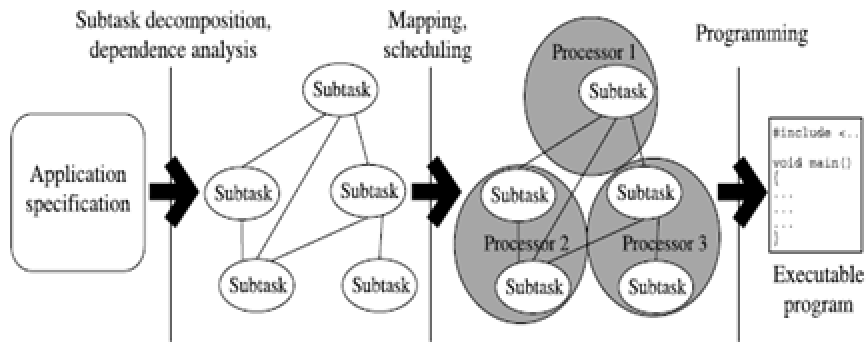
- Completion Time = Schedule Length
- This model assumes the existence of an I/O processor with every processor in the system.
- Communication delay between two tasks allocated to the same processor is negligible.
- Communication delay is counted only between two tasks assigned to different processors.

Completion time from the Gantt chart



The Three models of communication a) task graph; b) allocation and c) schedule communication.

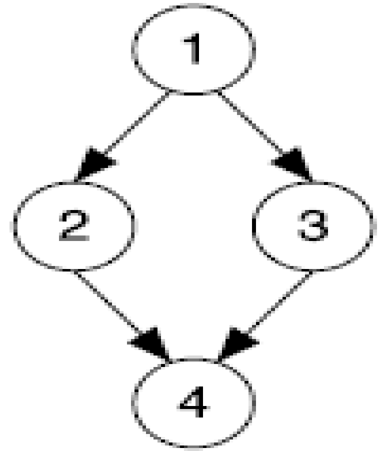
Parallel programming—process of parallelization



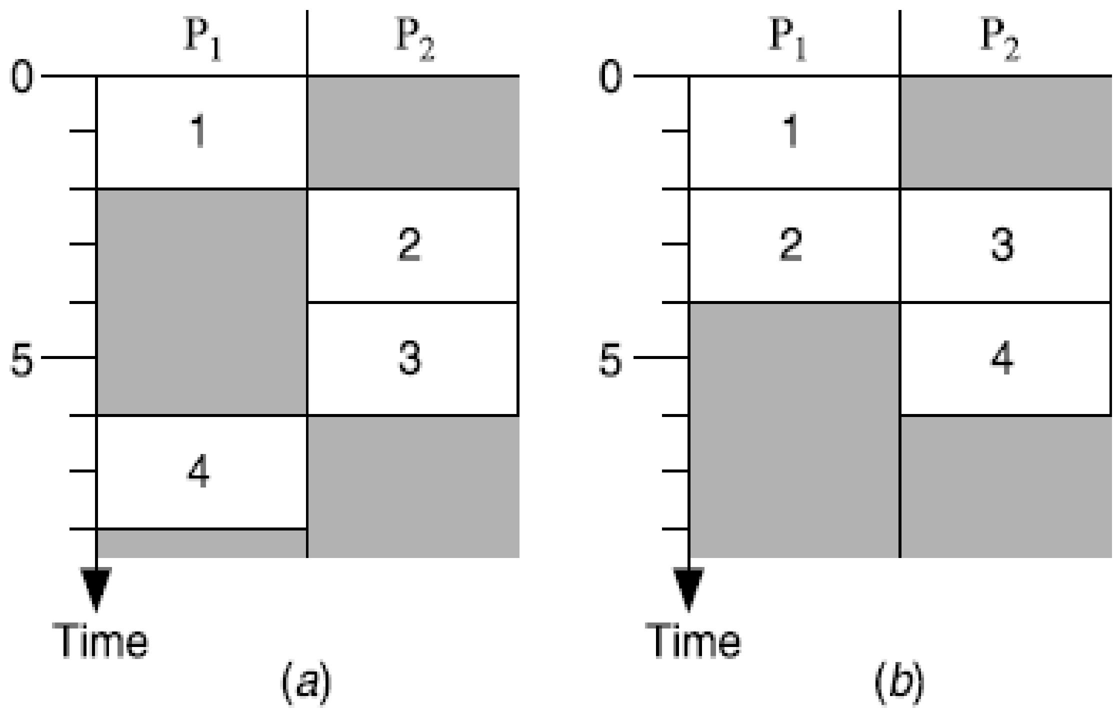
- They are represented as directed acyclic graphs (DAGs), called task graphs,
- where a node reflects a task and a directed edge a communication between the incident

nodes. Weights associated with the nodes and edges represent the computation and communication costs, respectively.

```
1: a = 2
2: u = a + 2
3: v = a * 7
4: x = u + v
```



Example of task graph representing a small program segment.



- yet schedule (b) is shorter than schedule (a).
- The reason is the precedence constraints among the nodes: in the schedule (a), the

two nodes that can be executed in parallel, nodes 2 and 3, are allocated to the same processor. In schedule (b), they are allocated to different processors and executed concurrently.

Data Dependence

- The best way to build an understanding for dependence is to start with a simple example. Consider the following equation:
    - $x = a * 7 + (a + 2)$.
  - Example 1 Program for $x = a * 7 + (a + 2)$
    - 1: $a = 2$
    - 2: $u = a + 2$
    - 3: $v = a * 7$
    - 4: $x = u + v$

- Example 2 Program for $x = a * 7 + (a * 5 + 2)$
- 1: $a = 2$
- 2: $v = a * 5$
- 3: $u = v + 2$
- 4: $v = a * 7$
- 5: $x = u + v$

- BASIC GRAPH CONCEPTS
- 1- (Graph) A graph G is a pair (V, E), where V and E are finite sets.
- An element v of V is called vertex and an element e of E is called edge.

- **Elementary Graph Algorithms**

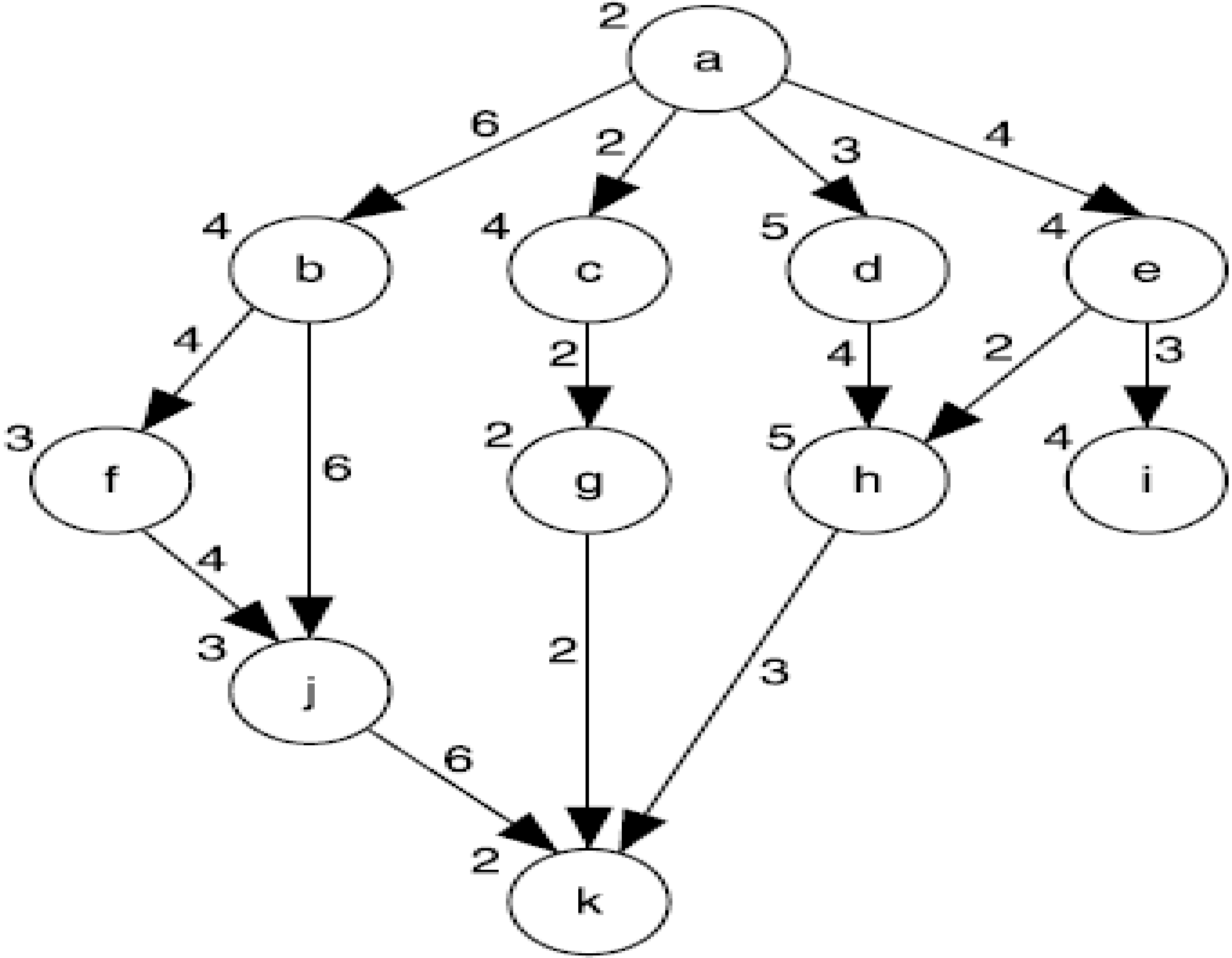
1. BFS (Breadth First Search)
2. DFS (Depth First Search)
3. Task Duplication Based Algorithms
4. Clustering Heuristic Algorithms

TASK GRAPH PROPERTIES

1 Critical Path

- An important concept for scheduling is the critical path—the longest path in the task graph.

(Critical Path (CP)) A critical path cp of a task graph $G = (V, E, w, c)$ is a longest path in G $len(cp) = \max_{p \in G}\{len(p)\}$.

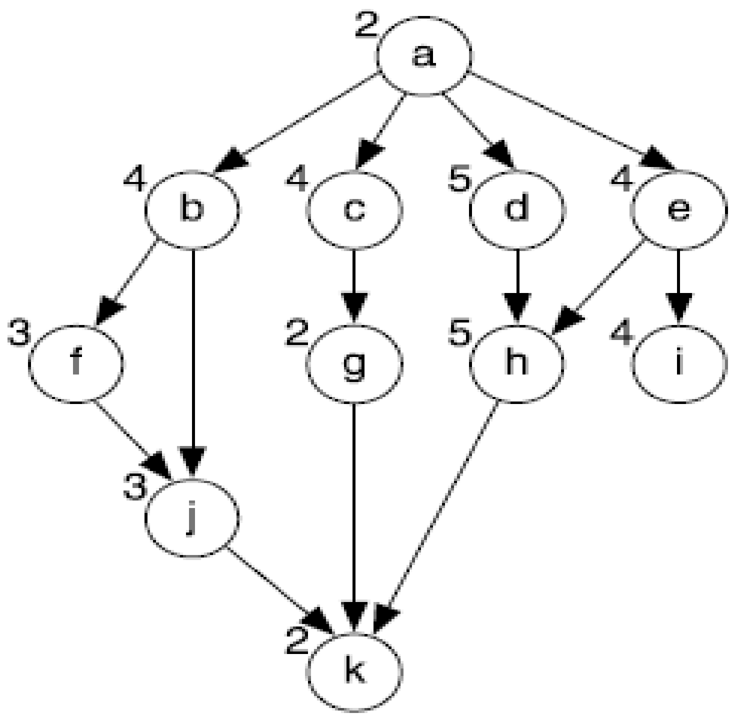


2
a
6
2
3
4
4
b
4
c
5
d
4
e
4
2
4
2
3
3
f
6
2
g
5
h
4
i
4
3
j
2
3
6
2
k

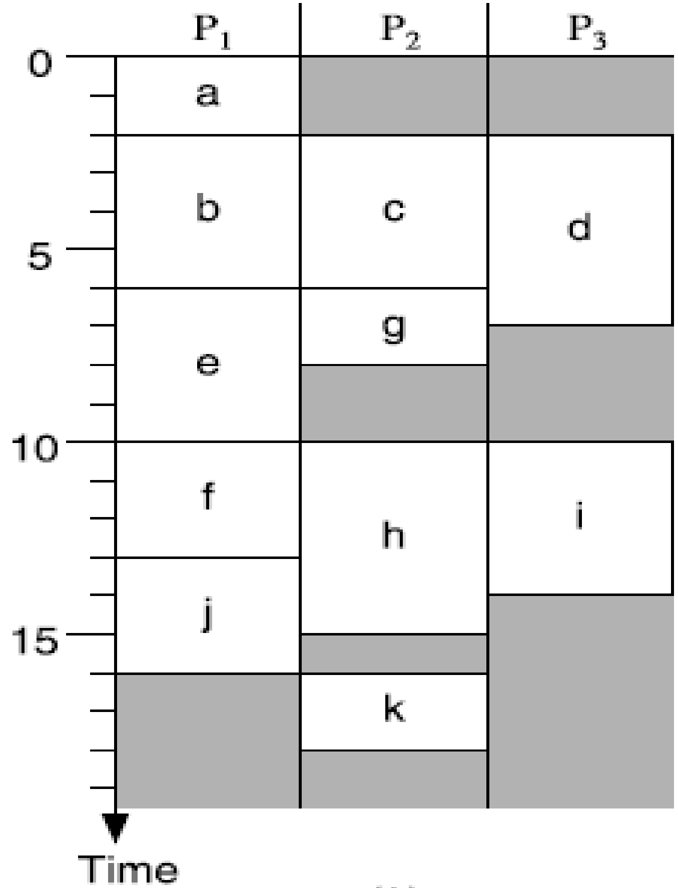
FUNDAMENTAL TAXONOMY

- static task scheduling as opposed to dynamic
- Dependent vs. independent tasks
- Allocation tasks vs scheduling tasks
- Optimal vs sub optimal

TASK SCHEDULING WITHOUT COMMUNICATION COSTS



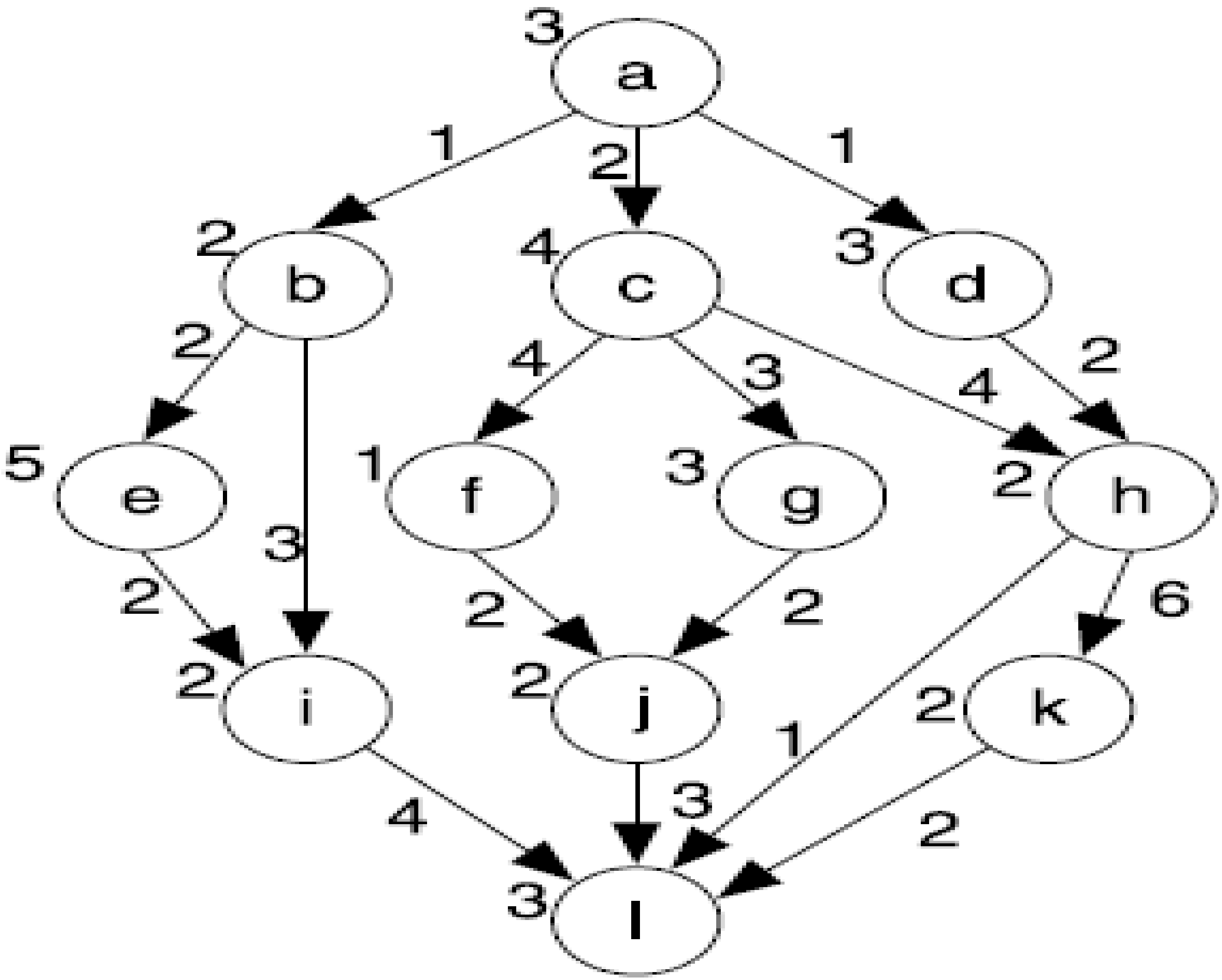
(a)



(b)

example

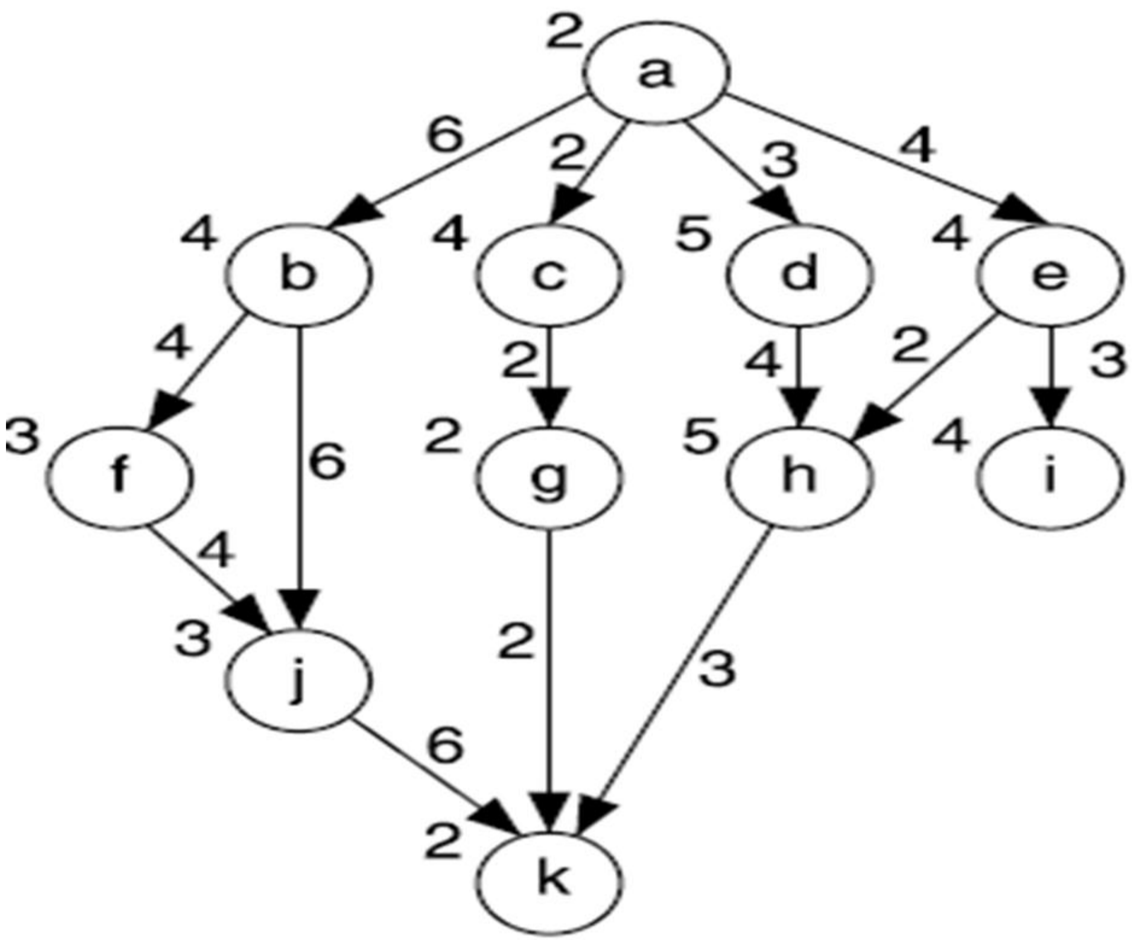
- Use list scheduling with start time minimization to schedule the following task
- graph on four processors:
- (a) The nodes shall be ordered in alphabetical order. What is the resulting schedule length?
- (b) Now order the nodes according BREADTH FIRST and repeat the scheduling. What is the resulting schedule length?

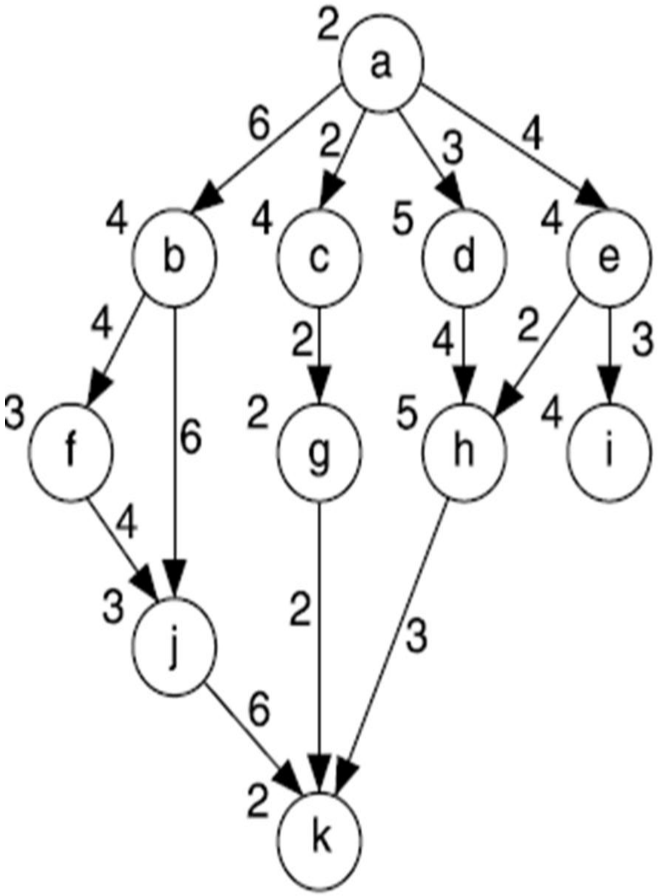

a
b
c
d
e
f
g
h
i
j
k
l

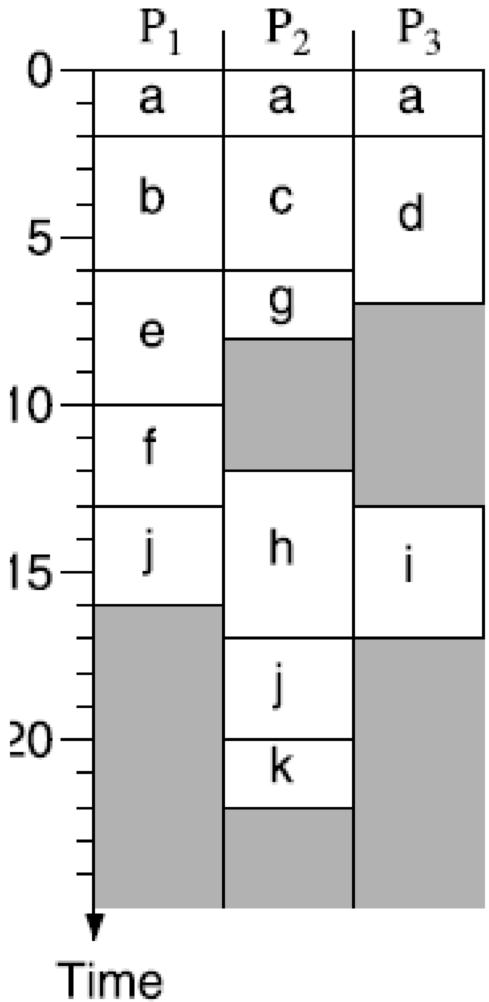
3-NODE DUPLICATION

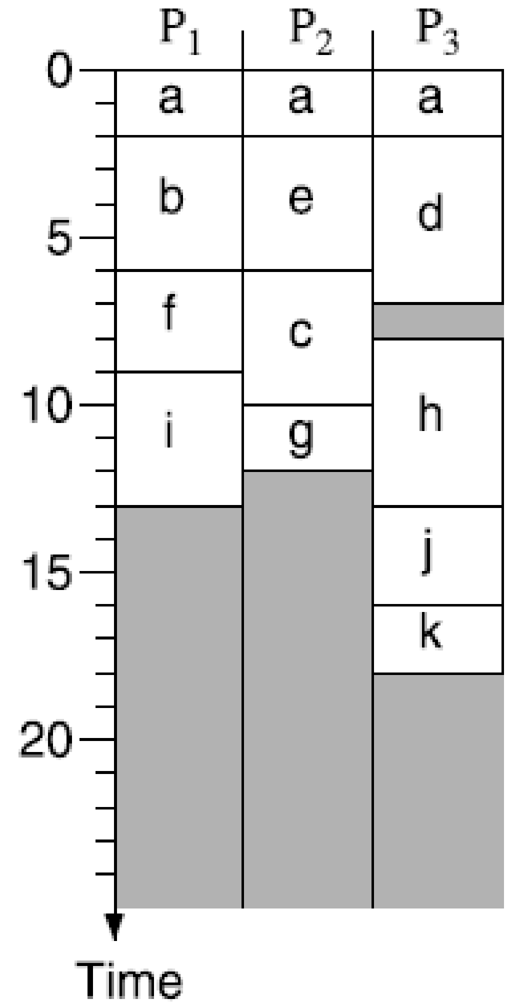
- A solution that has been exploited to reduce communication costs, while avoiding the above described problem, is node duplication. In this approach, some nodes of a task graph are allocated to more than one processor of the target system.

TASK SCHEDULING COMMUNICATION COSTS

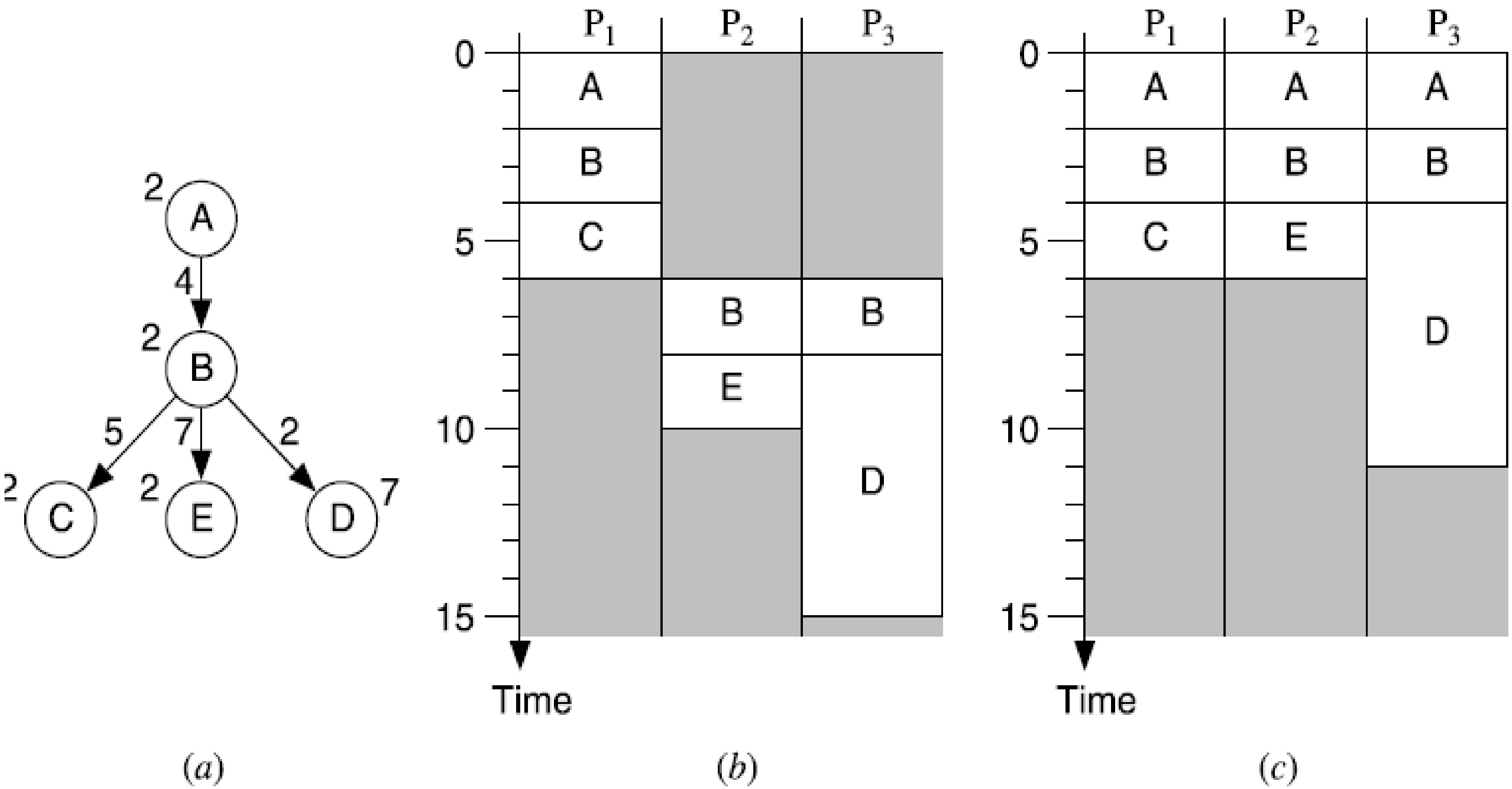



a
b
c
d
e
f
g
h
i
j
k


P_1
P_2
P_3
0
5
10
15
20
Time


P_1
P_2
P_3
0
5
10
15
20
Time

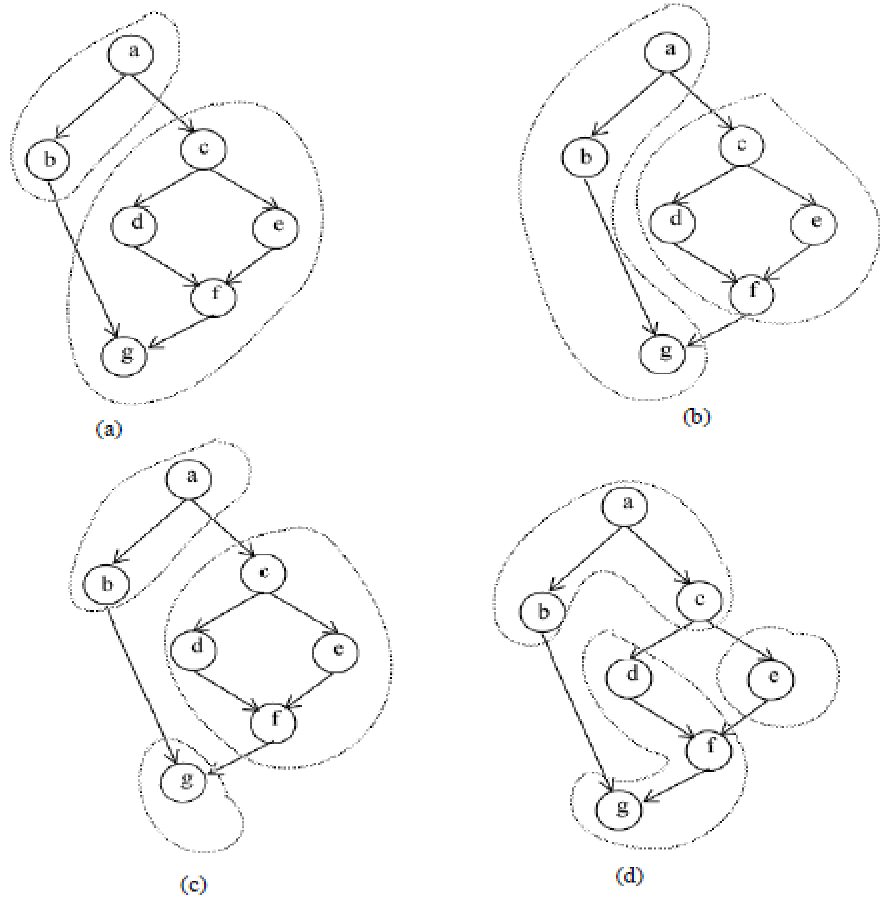
A small example task graph (a) and two schedules with node duplication; (b) only B is duplicated; (c) A and B are duplicated.



4- CLUSTRING

Clustering

- Different ways to cluster a task graph



Computational Models

- Speedup:
  - Time (one CPU): T(1).
  - Time (n CPUs): T(n).
  - Speedup: S
  - S = T(1)/T(n)

- Efficiency

$$E(p) = S(p)/\text{no. of processors}$$