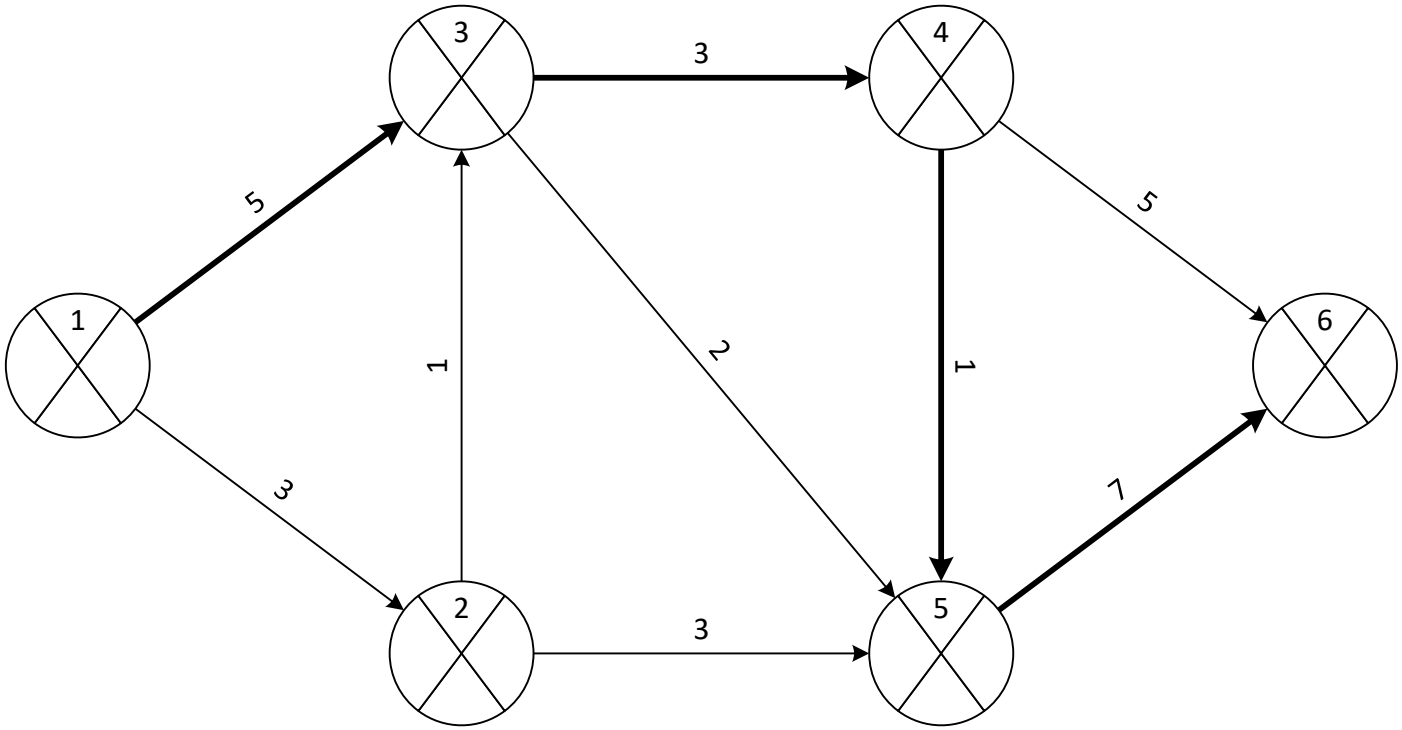
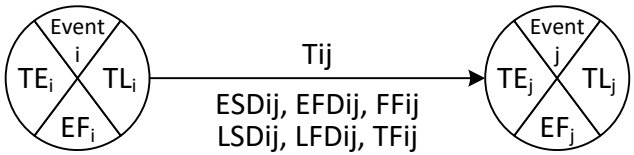
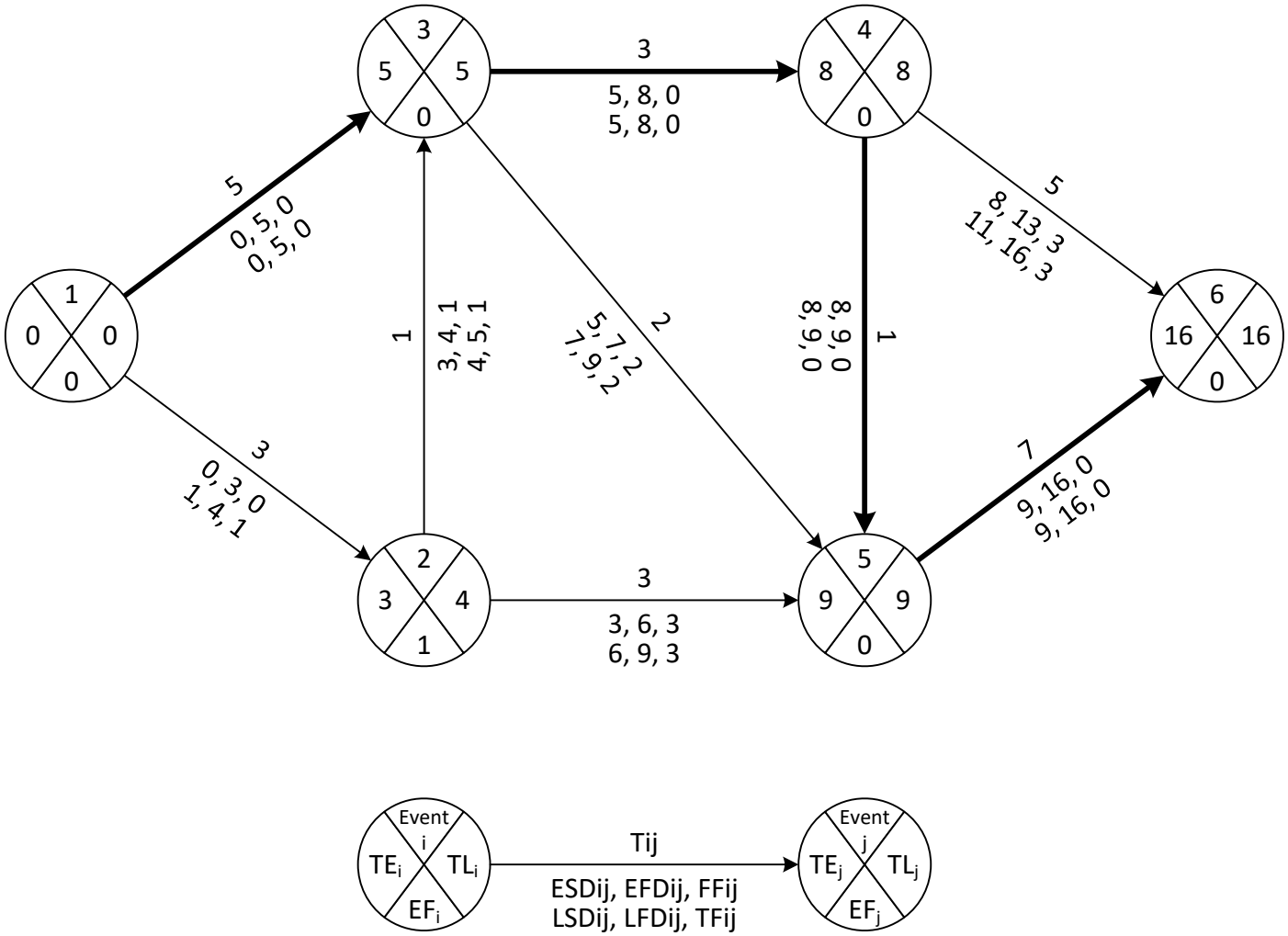
CEE 536

Example Calculations for Activity on Arrow Networks



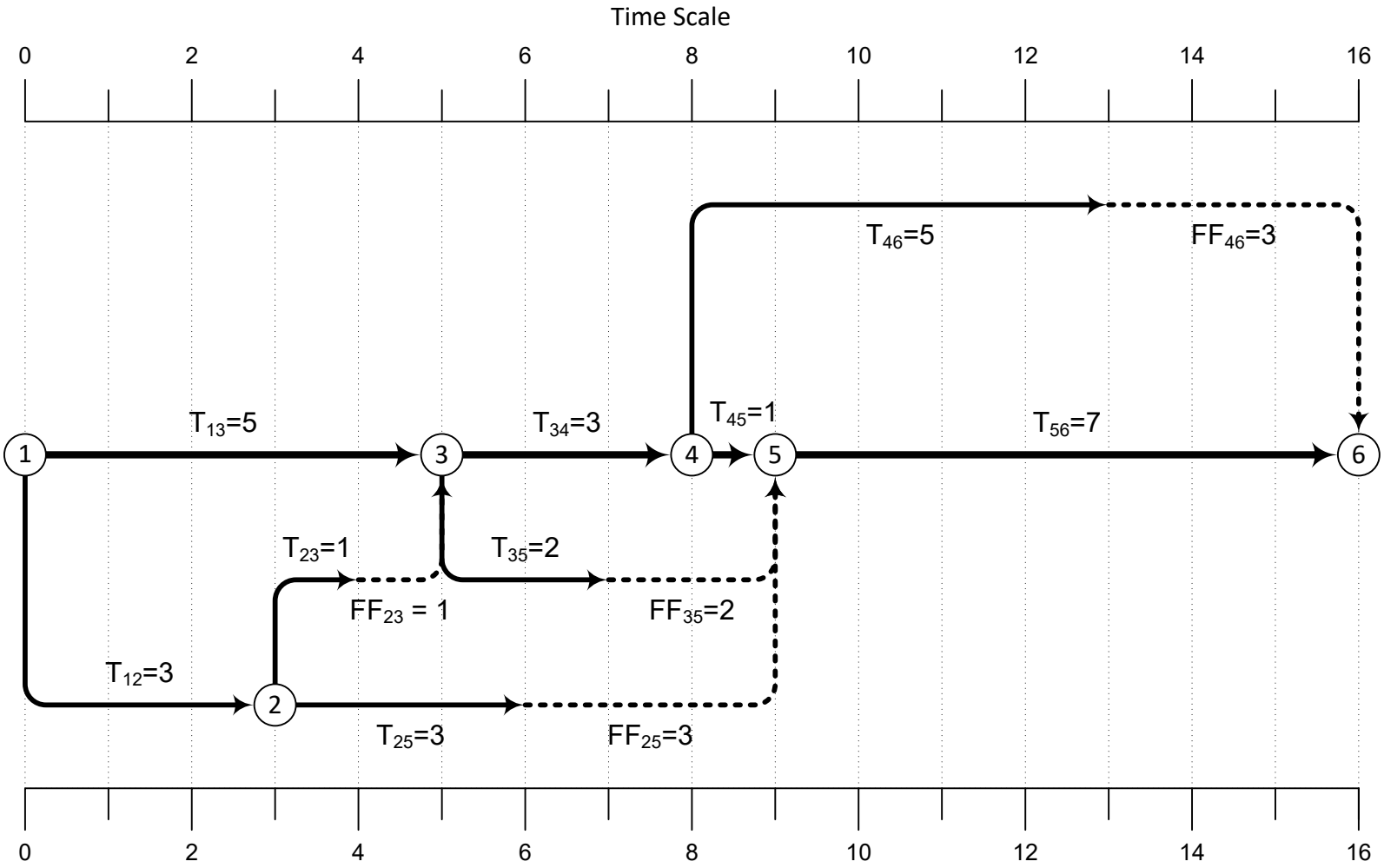


CEE 536
Example Calculations for Activity on Arrow Networks



P.G. Ioannou

Time-Scaled Activity-On-Arrow Network



The horizontal length of each solid arrow shows the duration of the corresponding activity.
The horizontal length of each dashed arrow equals the Free Float of the corresponding activity.
Dashed arrows are needed to show how activities (i,j) connect together to form an activity-on-arrow network.

P.G. Ioannou

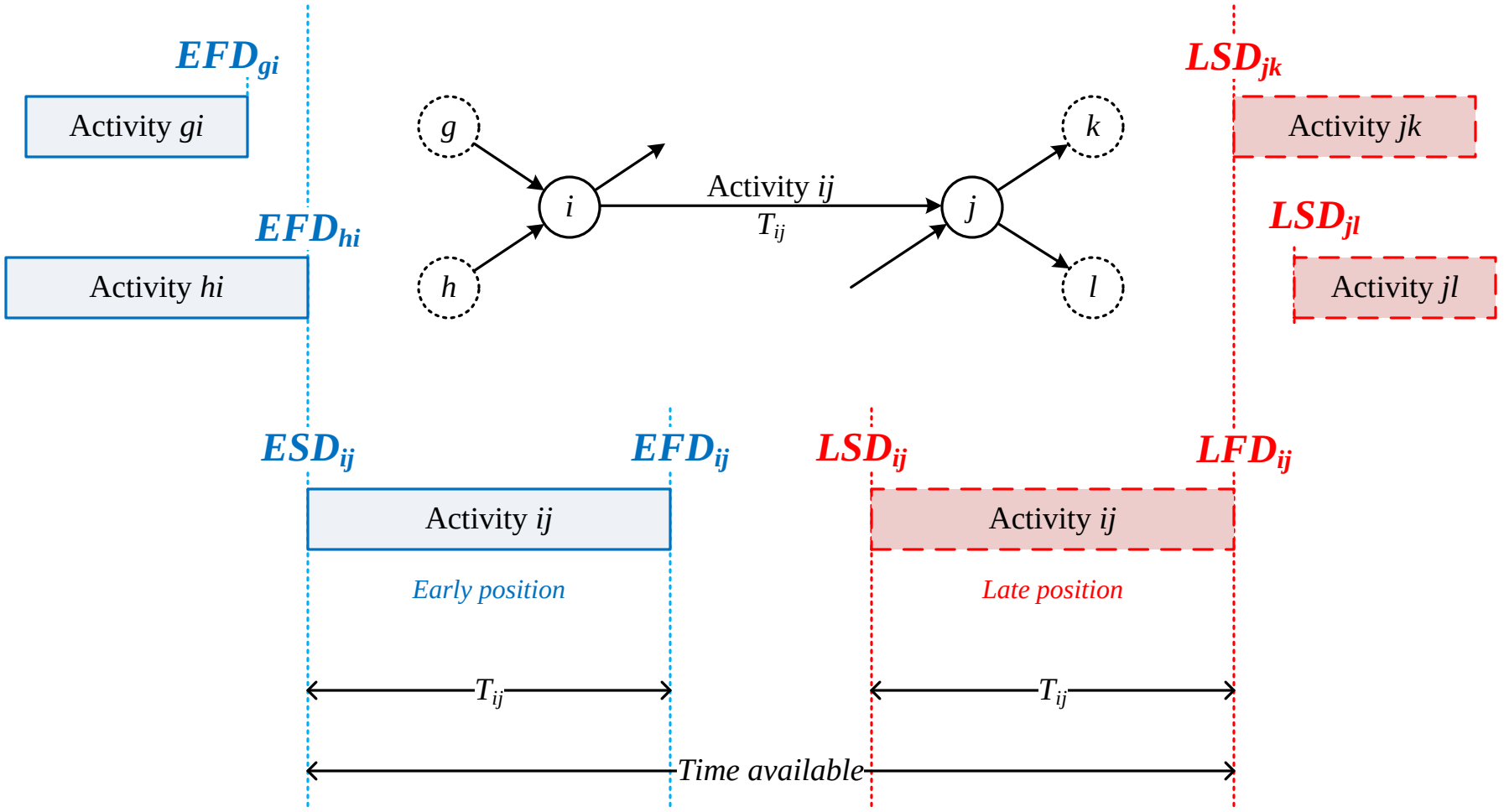


Figure 6.1 The four activity times —arrow diagram notation

Activity-Oriented Approach

Forward Pass Algorithm:

1. Assign $ESD_{st,j}$, $\forall\ j$ (where *st* is the *start* event; typically, $ESD_{st,j} = 0$)
2. Set $EFD_{st,j} = ESD_{st,j} + T_{st,j}$, $\forall\ j$
3. Find an activity (i,j) for which the $EFD_{k,i}$ for all immediate predecessors (k,i) are known.
4. Set $ESD_{i,j} = \max EFD_{k,i}$, $\forall\ k$
5. Set $EFD_{i,j} = ESD_{i,j} + T_{i,j}$
6. If the ESD and EFD for all activities have been computed, then STOP, else GOTO 3.

Minimum Project Duration = $\max EFD_{i,fin}$, $\forall\ i$ (where *fin* is the *finish* event)

Backward Pass Algorithm:

1. Assign $LFD_{i,fin}$, $\forall\ i$ (typically, $LFD_{i,fin} = \max EFD_{i,fin}$)
2. Set $LSD_{i,fin} = LFD_{i,fin} - T_{i,fin}$, $\forall\ i$
3. Find an activity (i,j) for which the $LSD_{j,k}$ for all immediate successors (j,k) are known.
4. Set $LFD_{i,j} = \min LSD_{j,k}$, $\forall\ k$
5. Set $LSD_{i,j} = LFD_{i,j} - T_{i,j}$
6. If the LSD and LFD for all activities have been computed, then STOP, else GOTO 3.

Floats:

Total Float: $TF_{i,j} = LFD_{i,j} - EFD_{i,j} = LSD_{i,j} - ESD_{i,j}$

If $TF_{i,j} = 0$, then activity (i,j) is on Critical Path.

Free Float: $FF_{i,j} = ESD_{j,k} - EFD_{i,j}$

If all acts (j,k) are dummies, use $\min ESD_{k,l}$, $\forall\ (k,l)$, instead of $ESD_{j,k}$

$FF_{i,j} \le TF_{i,j}\ \forall\ i,j$

$FF_{i,j} = TF_{i,j}$ if $\exists\ (j,k)$: $TF_{j,k} = 0$

Interfering Float: $INTF_{i,j} = TF_{i,j} - FF_{i,j} = LFD_{i,j} - ESD_{j,k}$

$INTF_{i,j} \le TF_{i,j}$

Independent Float: $INDF_{i,j} = ESD_{j,k} - (LFD_{h,i} + T_{i,j}) = FF_{i,j} - INTF_{h,i}$

If all acts (h,i) are dummies, use $\max LFD_{m,h}$, $\forall\ (m,h)$, instead of $LFD_{h,i}$

$INDF_{i,j} \le FF_{i,j} \le TF_{i,j}$. If $INDF_{i,j} < 0$, then set $INDF_{i,j} = 0$.

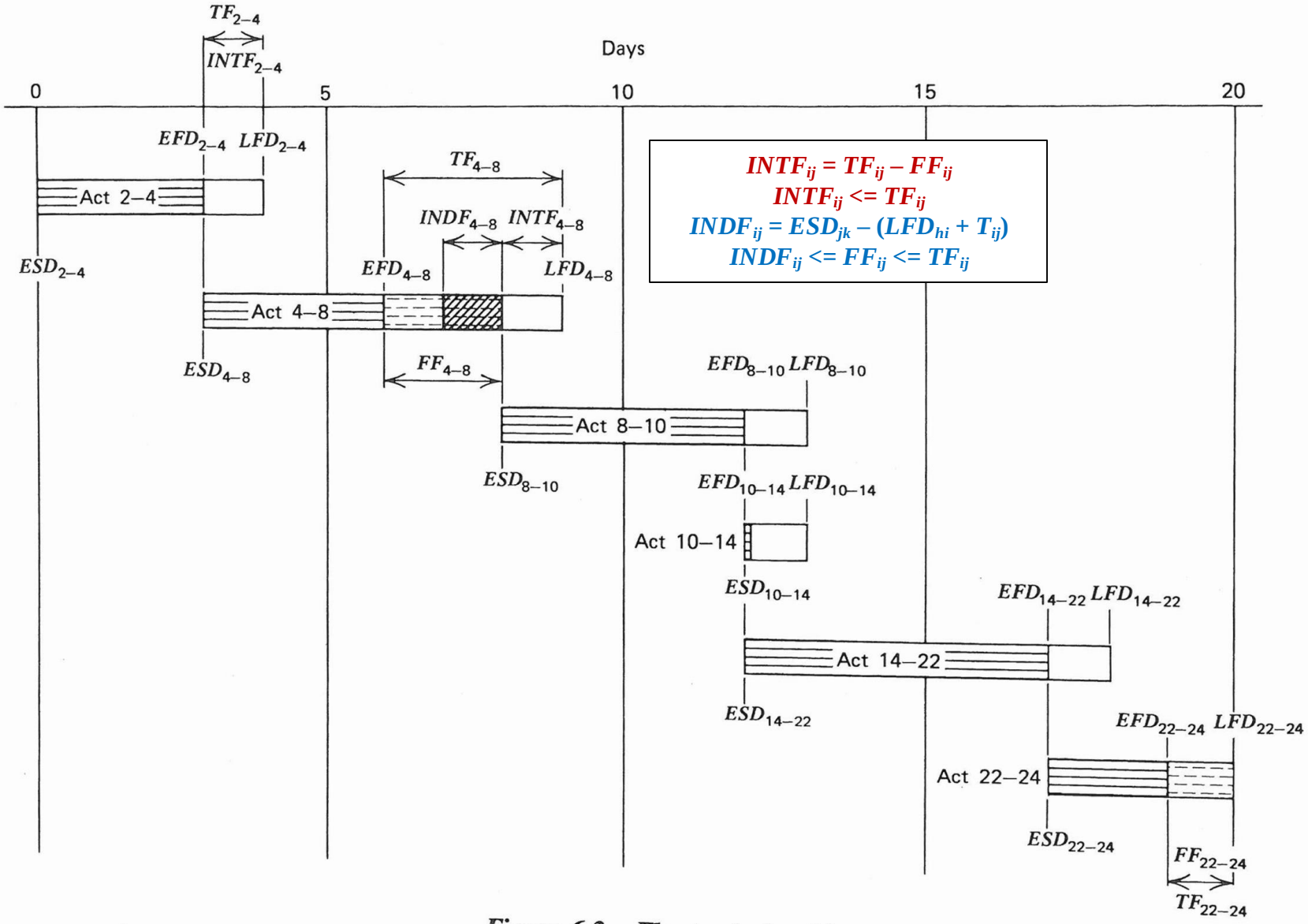


Figure 6.3 Float relationships.

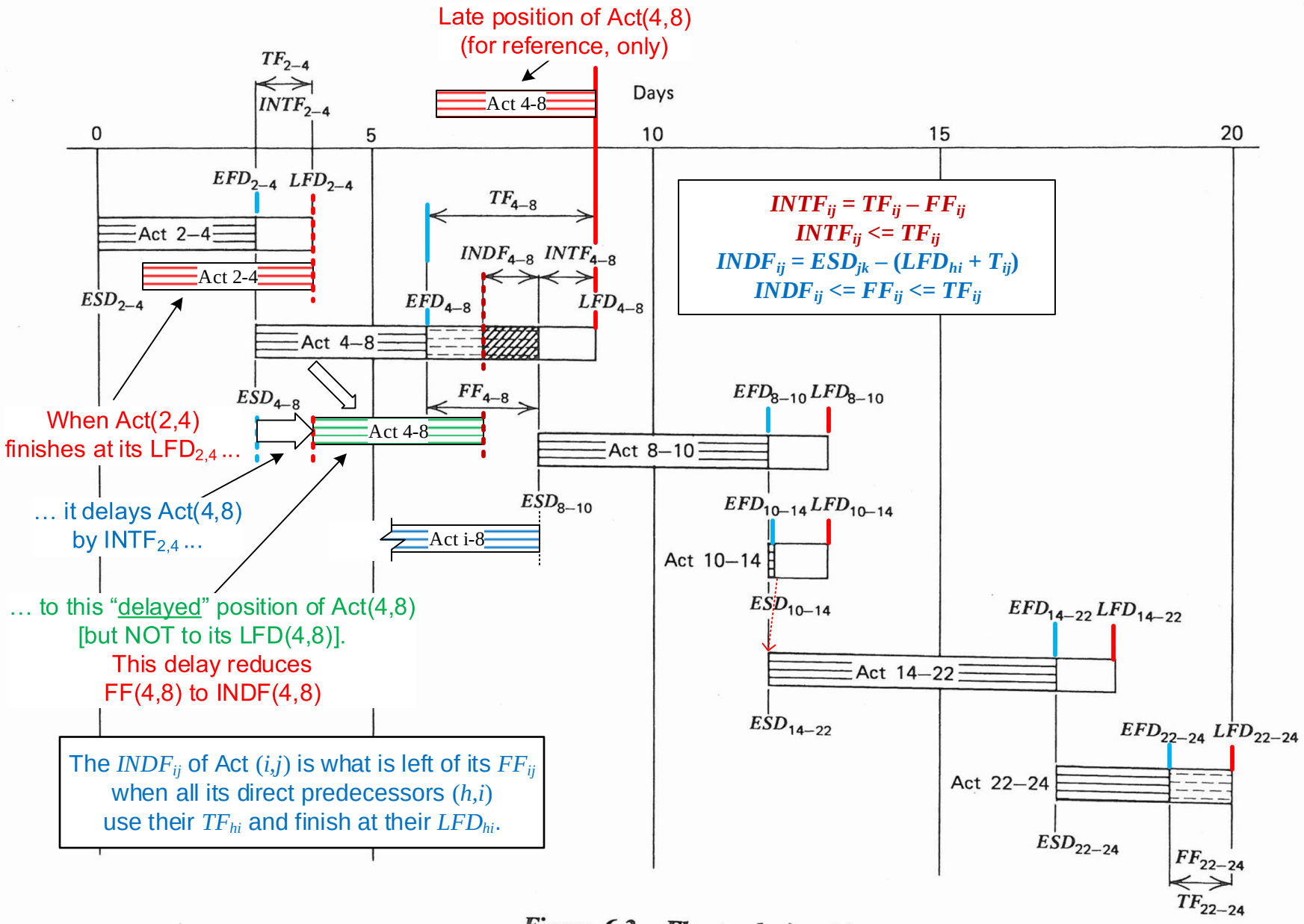


Figure 6.3 Float relationships.

Float Computations—Remodeling Chemical Laboratory

i	*j*	Activity Description	*T* (Days)	TEi *ESD*	TEi+TEij *EFD*	TLj-Tij *LSD*	TLj *LFD*	*TF*	*FF*	*INTF*	*INDF*
2	4	Strip room	3	0	3	1	4	1	0	1	—
2	12	Obtain fume hood	10	0	10	4	14	4	2	2	2
2	14	Obtain cabinets	10	0	10	3	13	3	2	1	2
2	20	Obtain chemical sink	10	0	10	7	17	7	4	3	4
* 2	24	Painter availability	20	0	20	0	20	0	—	—	—
2	28	Obtain vinyl floor covering	5	0	5	21	26	21	21	0	21
4	6	Rough-in plumbing and electrical	5	3	8	4	9	1	0	1	—
4	8	Replace existing fume duct	3	3	6	6	9	3	2	1	1
6	8	Dummy	0	8	8	9	9	1	0	1	—
6	10	Repair floor	1	8	9	12	13	4	3	1	2
8	10	Repair walls and ceiling	4	8	12	9	13	1	0	1	—
10	12	Dummy	0	12	12	14	14	2	0	2	—
10	14	Dummy	0	12	12	13	13	1	0	1	—
12	16	Install new fume hood	1	12	13	14	15	2	0	2	—
14	16	Dummy	0	12	12	15	15	3	1	2	0
14	22	Install wall cabinets	5	12	17	13	18	1	0	1	—
16	18	Install 1/3 base cabinets	1	13	14	15	16	2	0	2	—
18	20	Dummy	0	14	14	17	17	3	0	3	—
18	22	Install 2/3 base cabinets	2	14	16	16	18	2	1	1	0
20	22	Install chemical sink	1	14	15	17	18	3	2	1	0
22	24	Finish plumbing and electrical	2	17	19	18	20	1	1	0	0
24	26	Dummy	0	20	20	23	23	3	0	3	—
*24	28	Paint cabinets	6	20	26	20	26	0	—	—	—
26	28	Paint walls and ceiling	3	20	23	23	26	3	3	0	3
*28	30	Lay vinyl floor	1	26	27	26	27	0	—	—	—

Calculation of FF_{ij} and $INDF_{ij}$ when **all successors** or **all predecessors** of activity **(*i,j*)** are **dummies**

For both FF_{ij} and $INDF_{ij}$: If activity (*i,j*) is succeeded only by dummies, then it must not delay the min(*ESD*) of the successors of those dummies.

Incorrect: $FF_{6,10} = ESD_{10,12} - EFD_{6,10} = 12 - 9 = 3$

Correct: $FF_{6,10} = \min(ESD_{12,16}, ESD_{14,22}) - EFD_{6,10} = \min(12,12) - 9 = 3$

(It is a coincidence that for this example both results are the same, $FF = 3$.)

For $INDF_{ij}$: If activity (*i,j*) is preceded only by dummies, then its start is delayed to the max(*LFD*) of the predecessors of those dummies.

Incorrect: $INDF_{26,28} = ESD_{28,30} - (LFD_{24,26} + T_{26,28}) = 26-(23+3) = 0$

Correct: $INDF_{26,28} = ESD_{28,30} - (LFD_{22,24} + T_{26,28}) = 26-(20+3) = 3$

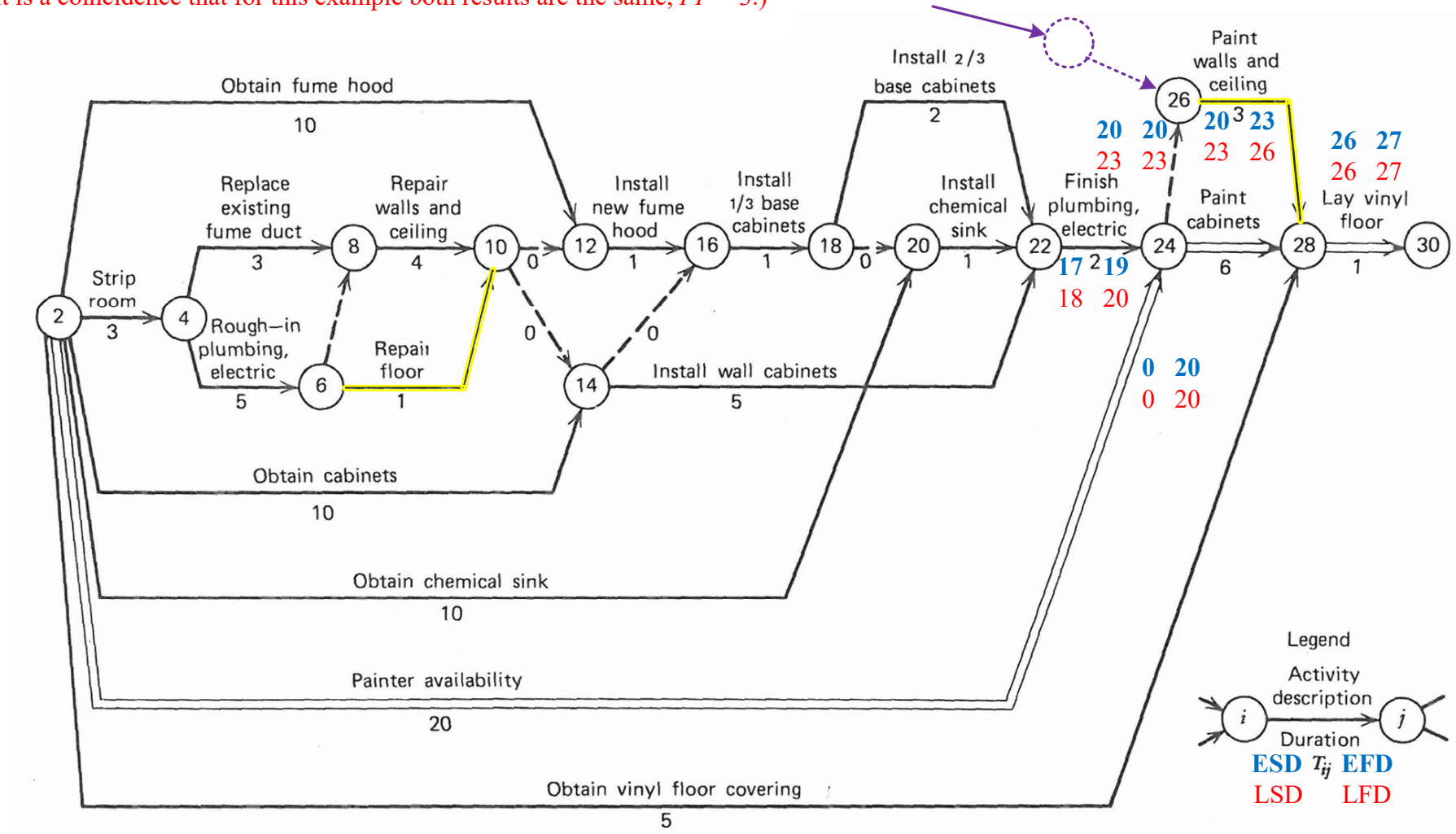


Figure 6.2 Arrow diagram—remodeling chemical laboratory.

Event-Oriented Approach

Forward Pass Algorithm:

1. Assign TE_{st} (where *st* is the *start* event; typically $TE_{st} = 0$)
2. Find a node *j* for which the TE_i for all immediately preceding nodes *i* are known.
3. Set $TE_j = \max\{TE_i + T_{i,j}\}, \ \forall\ i$
4. If the TE for all nodes have been computed, then STOP, else GOTO 2.

Backward Pass Algorithm:

1. Assign TL_{fin} (where *fin* is the *finish* event; typically, $TL_{fin} = TE_{fin}$)
2. Find a node *i* for which the TL_j for all immediately succeeding nodes *j* are known.
3. Set $TL_i = \min\{TL_j - T_{i,j}\}, \ \forall\ j$
4. If the TL for all nodes have been computed, then STOP, else GOTO 2.

Set: $ESD_{i,j} = TE_i$, $EFD_{i,j} = TE_i + T_{i,j}$, $LFD_{i,j} = TL_j$, $LSD_{i,j} = TL_j - T_{i,j}$

Floats:

Event Float: $EF_i = TL_i - TE_i$

Total Float: $TF_{i,j} = LFD_{i,j} - EFD_{i,j} = TL_j - (ESD_{i,j} + T_{i,j}) = TL_j - TE_i - T_{i,j}$

If $TF_{i,j} = 0$ (or $EF_i = 0$), then activity (i,j) (or node *i*) is on the Critical Path.

Free Float: $FF_{i,j} = ESD_{j,k} - EFD_{i,j} = TE_j - (ESD_{i,j} + T_{i,j}) = TE_j - TE_i - T_{i,j}$

If all activities (j,k) are dummies, use $\min TE_k, \ \forall\ k$, instead of TE_j

$FF_{i,j} \le TF_{i,j}$

$FF_{i,j} = TF_{i,j}$ if $EF_j = 0$ (or $TE_j = TL_j$)

Interfering Float: $INTF_{i,j} = TF_{i,j} - FF_{i,j} = LFD_{i,j} - ESD_{j,k} = TL_j - TE_j$

$INTF_{i,j} \le TF_{i,j}$

Independent Float: $INDF_{i,j} = ESD_{j,k} - (LFD_{h,i} + T_{i,j}) = TE_j - (TL_i + T_{i,j}) = FF_{i,j} - INTF_{h,i}$

If all activities (h,i) are dummies, use $\max TL_h, \ \forall\ h$, instead of TL_i

$INDF_{i,j} \le FF_{i,j} \le TF_{i,j}$

If $INDF_{i,j} < 0$, then set $INDF_{i,j} = 0$.

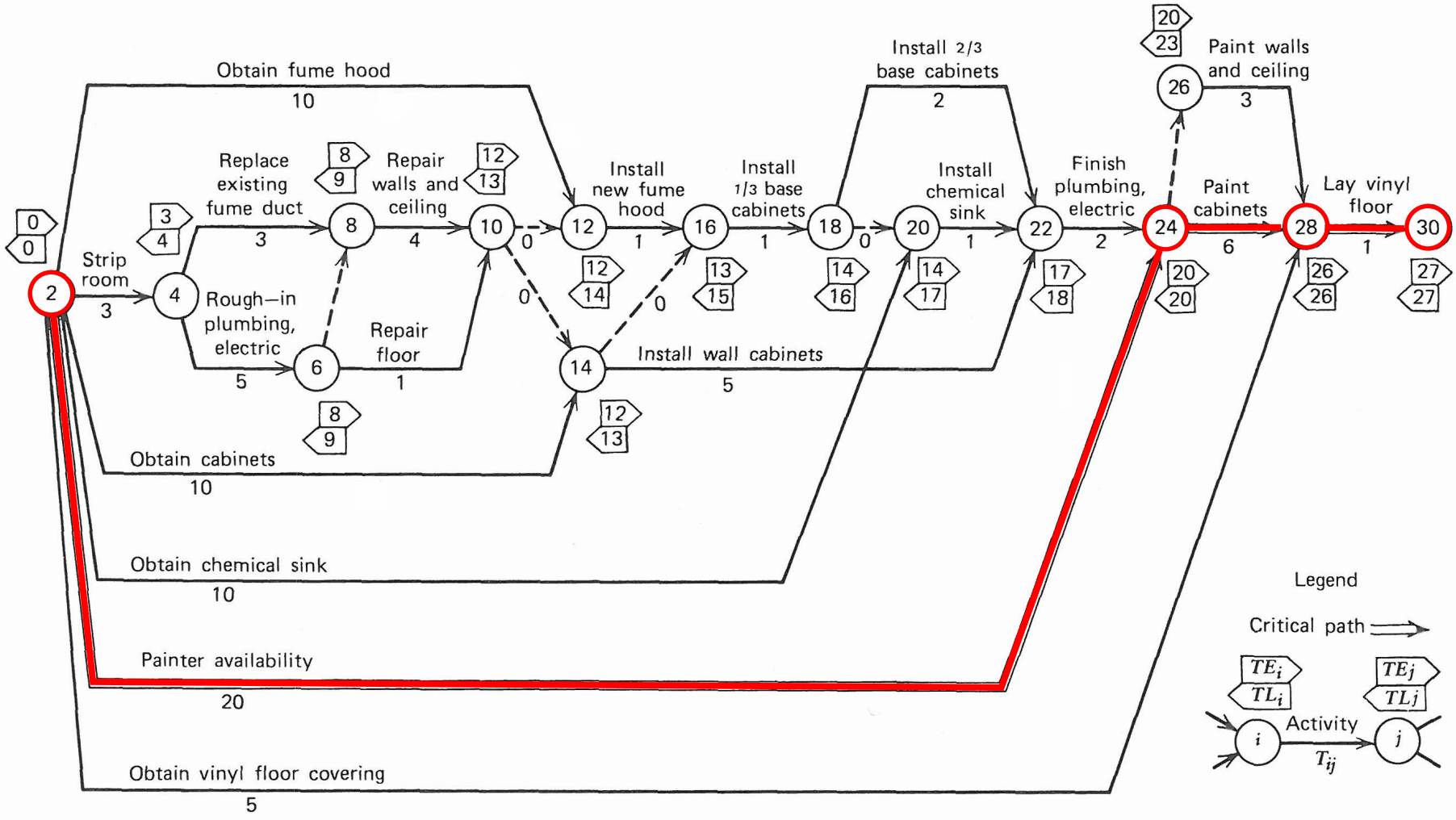


Figure 6.4 ***Critical path computations on an arrow diagram—remodeling chemical laboratory.***

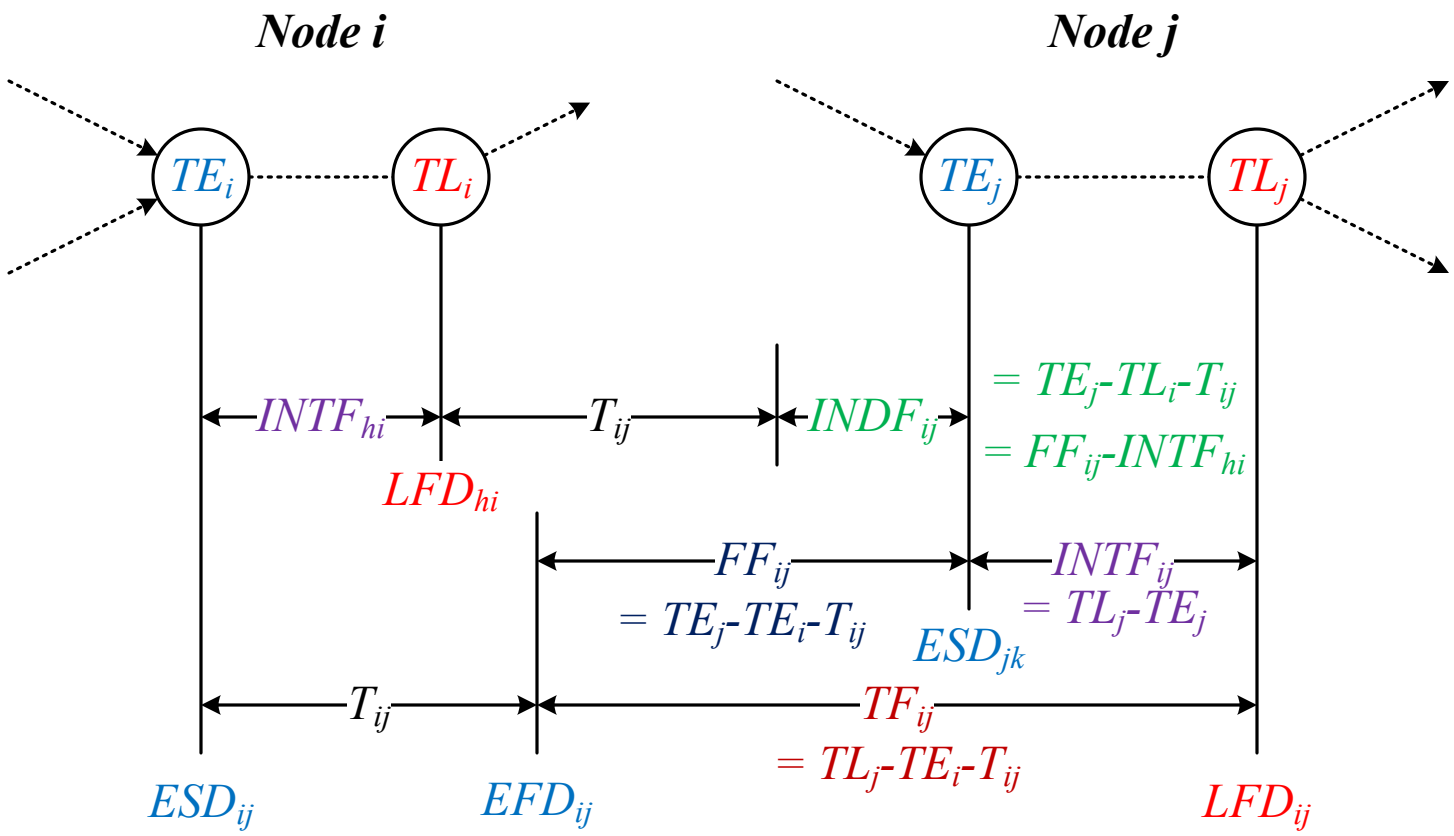


Figure 6.5 Float relationships in event time notation

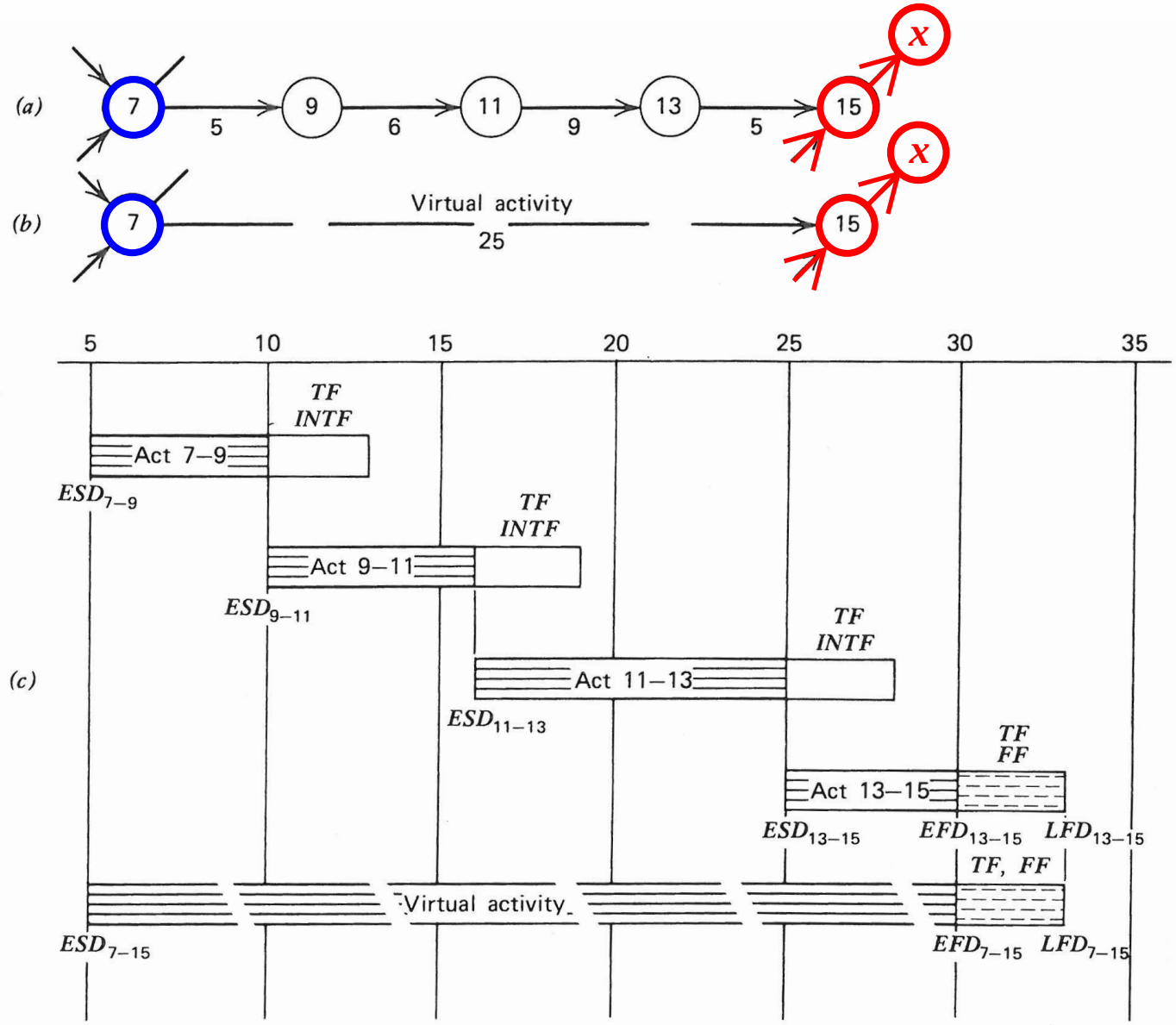


Figure 6.6 Float along a simple activity chain.

CHAPTER 6: Scheduling Computations for Arrow Networks

Activity-on-Arrow Network Facts

- All activities (i,j) that come out of the same node i have ***the same***:
  - Set of direct predecessors {(h,i)}
  - Early Start Date: $ESD_{ij} = TE_i$
- All activities (i,j) that go into the same node j have ***the same***:
  - Set of direct successors {(j,k)}
  - Late Finish Date: $LFD_{ij} = TL_j$
  - Interfering Float: $INTF_{ij} = EF_j$
- $ESD_{ij} = TE_i$ is the duration of the longest path (the longest sequence of activities) connecting the single *starting* node (event) to node (event) i.
- $LFD_{ij} = TL_j$ is the project duration minus the duration of the longest path (the longest sequence of activities) connecting node (event) j to the single *finishing* node.
- The *critical path* is the longest path (sequence of activities) connecting the single *starting* node (event) to the single *finishing* node (event).
- A network may have multiple critical paths, all with the same duration. Critical activities may belong to multiple critical paths.
- *TF* is by how much an activity can be delayed without delaying the project.
- An activity with *TF*=0 is a *critical activity*. Delaying a critical activity delays the project.
- The *critical path* is a sequence of *critical activities*, all with *TF*=0.
- For a critical activity, all floats are zero, not just the *TF*.
- *TF* consists of two parts; the beginning part is *FF* and the rest is *INTF*.
- *TF* = *FF* + *INTF*. As a result, *FF*<=*TF* and *INTF*<=*TF*.

- *FF* is by how much an activity can be delayed and not delay the *ESD* of a direct successor.
- *INTF* is that part of TF which delays (*interferes* with) the *ESD* of a direct successor.
- All activities (*i,j*) that go into *the same* node *j* have the same $INTF_{ij} = EF_j$
- For an activity to have *FF* it must have *TF*. If *TF*=0 then *FF*=0.
- *TF* and *FF* computations assume that the activity can start at its *ESD*.
- If an activity is succeeded *directly* by a *critical* activity, then *TF*=*FF* and *INTF*=0 (because any *TF*>*FF* would delay the *critical* direct successor and thus would also delay the project).
- Activities on any path *share* their *TF*. If one activity is delayed by *TF*, then the *TF* of succeeding activities on the same path *may* be reduced.
- If an activity is delayed by *TF* then the *FF* of *any* succeeding activity *may* be reduced. (By definition of *INDF*, *FF* is reduced at most to *INDF*).
- At least one of the activities that *go into the same node* must have *FF*=0.
- Of the activities that go into the same node, the activity with *FF*=0 is the one that has max*EFD* and determines the *ESD* of its direct successors.
- If only one activity enters a node, then it must have *FF*=0.
- If two activities enter the same node and one activity has *FF*>0 then the other activity must have *FF*=0.
- For an activity to have *FF*>0 it is necessary (but not sufficient) that at least one *other* activity goes into the same node (so the *other* activity can be the one with *FF*=0).
- *FF* is the only float that can be calculated using only ***early*** start and **early** finish times (i.e., no backward pass is needed).
- *INDF* is by how much an activity can still be delayed and not delay the *ESD* of any direct successor, when its own start has been delayed because its direct predecessors used their *TF* and finished at their *LFD*.

- *INDF* assumes a worst-case scenario: An activity's direct predecessors finish late (at LFD) and its direct successors must start early (at *ESD*).
- *INDF* is the *FF* that remains when the start of the activity is delayed because all direct predecessors used their *TF* and finished at their *LFD*.
- *INDF* is the latter (i.e., ending) part of *FF*: As a result, *INDF* <= *FF*
- For an activity to have *INDF* it must have *FF*. If *FF*=0 then *INDF*=0.
- *INDF* = *FF* – (*INTF* of a direct predecessor).
- An *activity chain* is a sequence of activities from some *starting* node (event) to some *ending* node (event) such that only one activity *enters* and only one activity *exits* any of the *intermediate* nodes (events).
- An activity chain is defined by its *starting* node and its *ending* node.
- By definition, more than one activity must enter or exit the *starting* node and more than one activity must enter or exit the *ending* node of an activity chain.
- All activities on an activity chain can be substituted by one "*virtual activity*" that connects the *starting* and *ending* nodes and whose duration is the sum of the durations of the activities on the activity chain.
- All activities on an activity chain have the same *TF* which equals the *TF* of the "virtual activity".
- By definition, each activity on an activity chain (except the last) is the only activity that enters its succeeding node (event). Hence, all activities on an activity chain must have *FF*=0 except for the last activity which *may* have FF>0.

6.11 Figure Ex. 6.3 represents a portion of an arrow network. Compute the remaining total float, free float, and interfering float values for Activities G, H, K, L, M, N, P, R, S, T, and U.

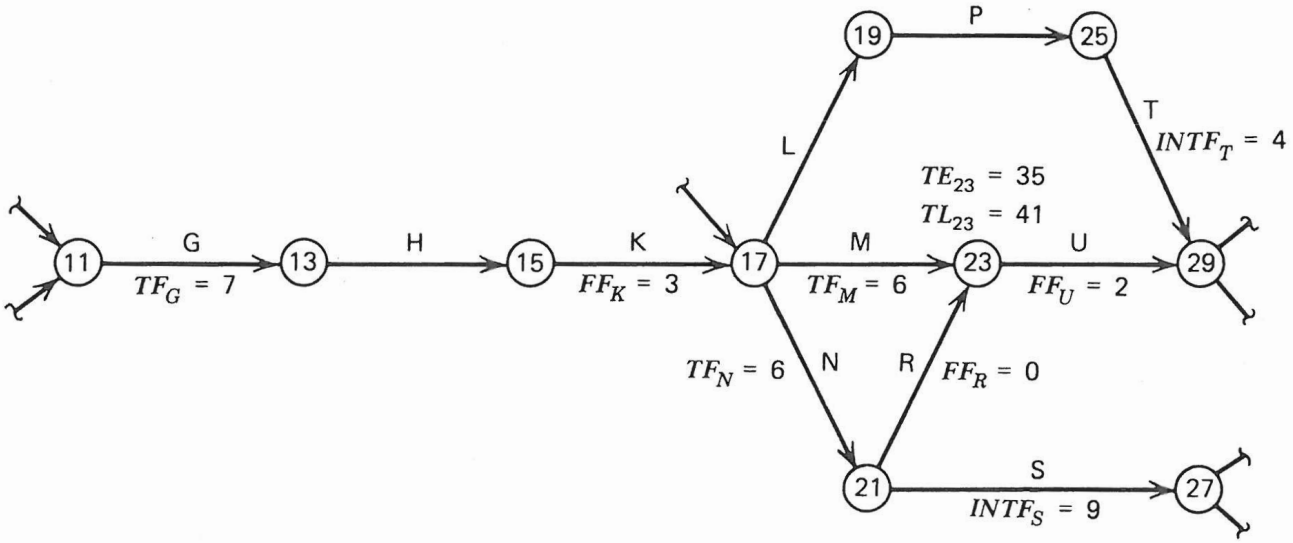


Figure Ex. 6.3