PROJECT MANAGEMENT

Topic 5

Resource Management

Contents

- Resource loading
- Creating Resource Loading Charts
- Constraints for Resource
- Assignment

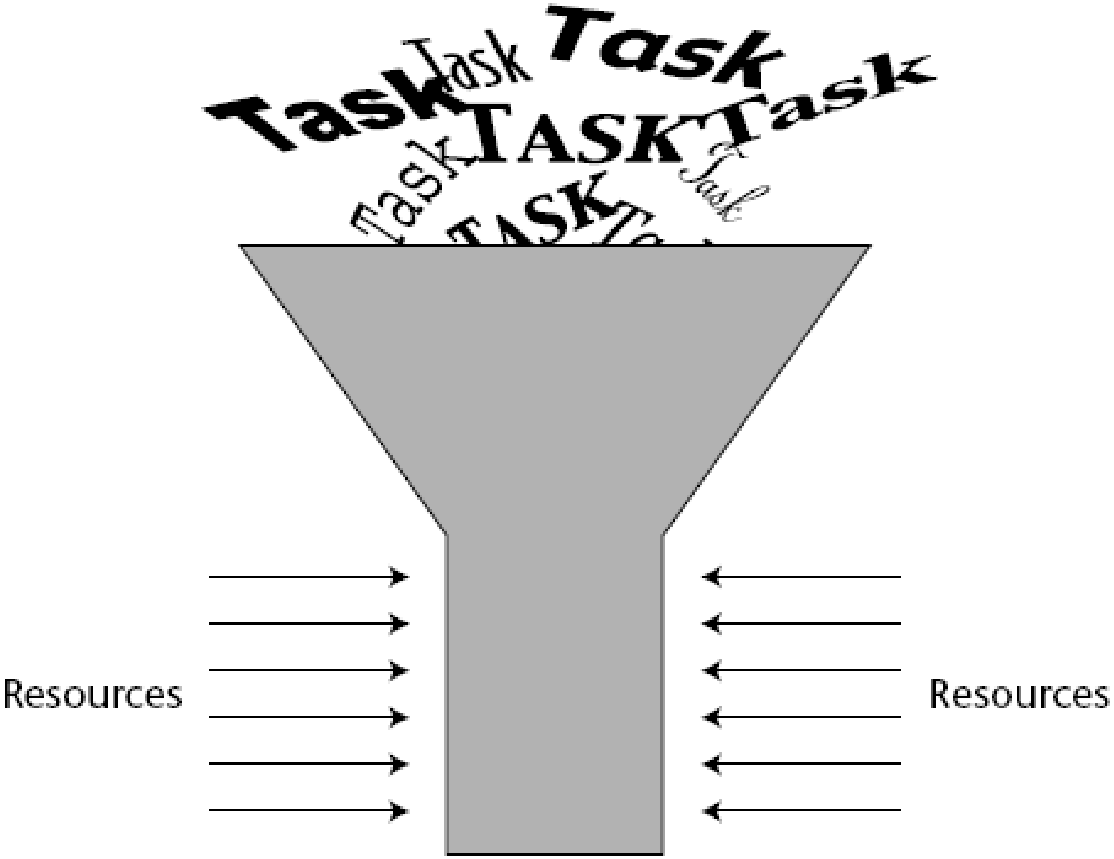
Resource Loading

The amounts of individual resources that a schedule requires during specific time periods.

Resource loading table

Resource Name	Work	Details	5/5	5/12	5/19	5/26
Tom	40 hrs	Work	8h	32h		
Assign Bids	*40 hrs*	Work	8h	32h		
Jeff	40 hrs	Work		8h	32h	
Calculate Cost	*40 hrs*	Work		8h	32h	
Sue	40 hrs	Work		8h	32h	
Select Bid	*40 hrs*	Work		8h	32h	
Carol	8 hrs	Work				8h
PR Campaign	*8 hrs*	Work				8h

Resource Loading



Resource leveling is a process that the project manager follows to schedule how each resource is allocated to tasks in order to accomplish the work within the scheduled start and finish dates of the task.

Resource Loading

Resource schedule needs to be **leveled**

- To ensure that no resource is **over-allocated**. That is, you do not schedule a resource to more than 100 percent of its available time.

- The project manager wants the number of resources (people, in most cases) to follow a logical pattern throughout the life of the project. (the number of resources working on a project at any time is fairly **constant**).

Resource Loading

Resource-Leveling Strategies

1. **Utilizing available slack**: one or more of the project tasks are postponed to a date that is later than their early start date but no later than their late finish date.

2. **Shifting the project finish date**

3. **Smoothing**

   - Occasionally, limited overtime is required to accomplish the work within the scheduled start and finish dates of the task.
   - Overtime can help alleviate some resource over-allocation because it allows more work to be done within the same scheduled start and finish dates.

Resource Loading

Shifting the Project Finish Date

- The critical path may have to be **extended** to achieve an acceptably resource-leveled schedule.
- This case could very well mean that the parallel scheduling on the task network diagram that moved the original finish date to an earlier one needs to be reversed.
- The start-to-start and finish-to-finish dependencies might need to be set back to the linear finish-to-start type.
- If you find yourself caught between over-allocated resources on a schedule that cannot be acceptably leveled and a firm, fixed completion date, you may have to consider reducing the scope of the project.

Creating Resource Loading Charts

Resource Leveling (smoothing)

A ***multivariate***, ***combinatorial*** problem

Objectives

- To determine the resource requirements so that they will be available at the **right time**.
- To allow each activity to be scheduled with the **smoothest possible transition** across resource usage levels.

Resource Loading

Alternative Methods of Scheduling Tasks

Rather than treating the task list as fixed and within that constraint leveling resources, you could resolve the leveling problem by considering further decomposition of one or more tasks.

- ❑ Further Decomposition of Tasks
- ❑ Stretching Tasks
- ❑ Assigning Substitute Resources.

Resource Loading

Further Decomposition of Tasks

- Suppose that a task requires one person for three days within a five-day window. There are two days of slack in the schedule for that task.
- The unavailability of the resource for three consecutive days beginning on the ES date will require scheduling the task work to a longer period of time.
- One solution would be to have the resource work for three nonconsecutive days as early as possible in the five-day window.
- Suppose that the resource is available for the first two days in the five-day window and for the last day in the five-day window.
- The project manager could **decompose** the five-day task into two tasks — one two-day task and one one-day task. The two-day task would then have an FS dependency on the one-day task.

Resource Loading

Stretching Tasks

- Another alternative that preserves the continuity of the task work is to stretch the work over a **longer period of time** by having the resource work on the task at a percent per day lower than was originally planned.
- In previous example: Suppose the resource is available 80 percent of each day in the five-day window, and you need four days of work.

Resource Loading

Stretching Tasks…

- In this simple example, the percentage was constant over the five days, but it might also follow some profile.
- For example, suppose you needed the resource for three days and the resource was available full-time for the first and second days but only half-time for the remaining three days of the five-day window.
- You could first split the task into two tasks—a two-day task and a one-day task.
- The two-day task would fully use the resource and get two days of work completed.
- The second task would be stretched to two days, and the resource would be assigned half-time for two days to complete the remaining day of work on the task.

Resource Loading

Assigning Substitute Resources

- One approach would be to **use less-skilled resources** and add to the total number of hours requested. Here, the thinking is that a less-skilled resource would require a longer period of time to complete the task work.
- This strategy works only for **noncritical path** tasks. Using it for a critical path task would extend the completion date of the project.

Resource Loading

Cost Impact of Resource Leveling

- If the resources are billable based on the labor expended, project costs do not increase.
- If there are resources that are charged on a calendar basis, project costs will increase.
- If there are incentives for early completion and penalties for late completion of a project, a cost impact will be felt as well.

Resource Loading

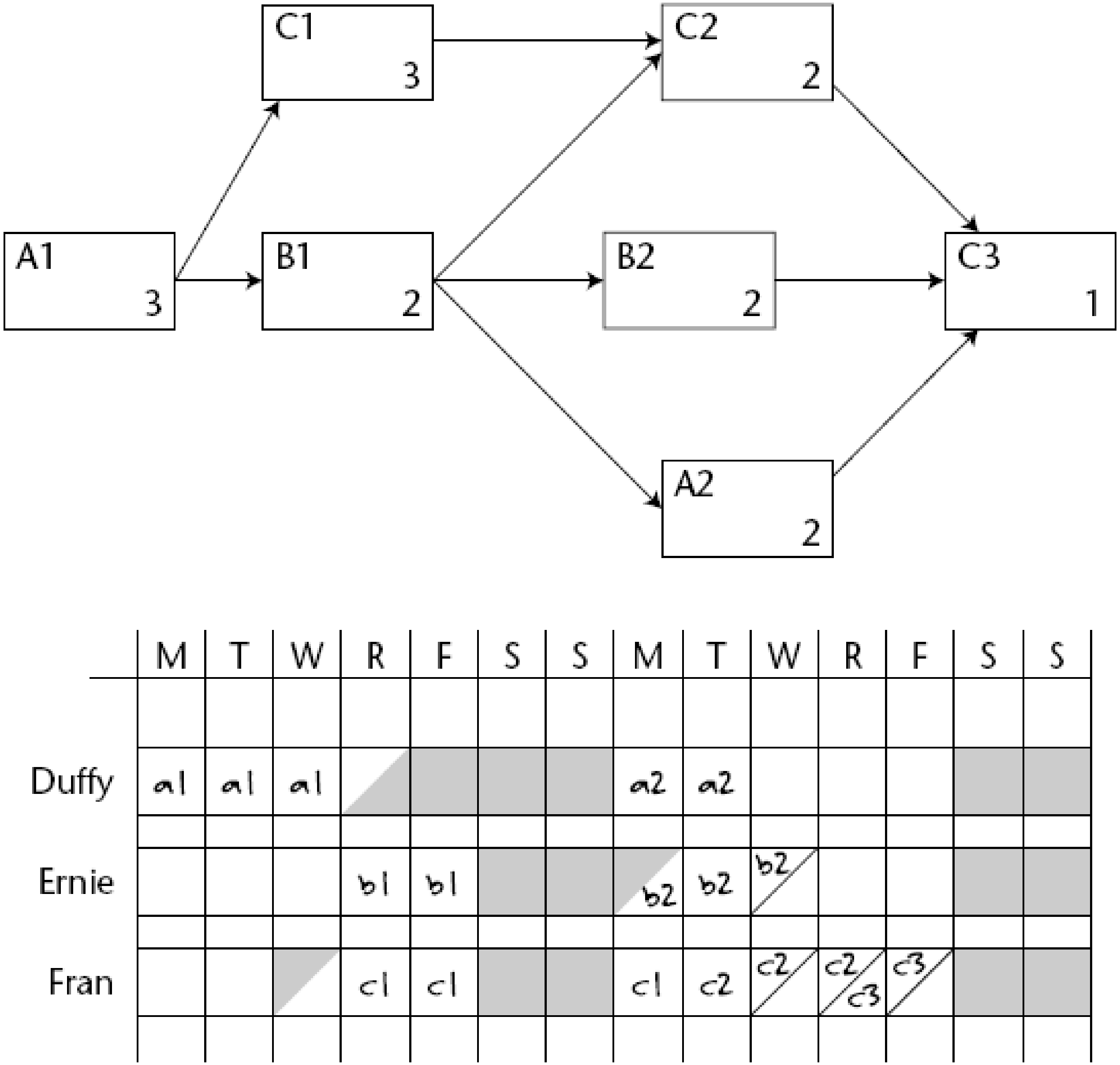
Implementing Micro-Level Project Planning

- Micro-level planning is another step in the decomposition of the tasks that are assigned to an individual. It involves a decomposition to *subtasks*.
- Micro-level project planning begins with the lowest-level task defined in the WBS.
- Because it appears in the WBS, it will have management oversight by the project manager.
- The responsibility for completing this task within a defined window of time will be assigned to a task manager.

Resource Loading

Implementing Micro-Level Project Planning…

- These tasks will each be less than two weeks' duration, so the subtasks that make them up will be of shorter duration. The decomposition should be fairly simple and result in tasks of one to three days' duration.
- In figure 2.7: The shaded areas of the schedule are non-workdays and days when a resource is not available.
- There is no need for software support, which simply adds management overhead with little return on the investment of time expended to capture and manage it.



	M	T	W	R	F	S	S	M	T	W	R	F	S	S
Duffy	a1	a1	a1	▨	▨	▨	▨	a2	a2				▨	▨
Ernie				b1	b1	▨	▨	b2	b2	b2			▨	▨
Fran			▨	c1	c1	▨	▨	c1	c2	c2	c2 c3	c3	▨	▨

Figure 7-2 Example of a task dependency diagram and time-scaled resource schedule

Resource Loading

Work Packages

- The work to be done within a task is called a *work package*.
- The work package is a statement by each task manager as to how he or she plans to complete the task within the scheduled start and finish dates.
- If the task manager or anyone working on the task were not available, someone else could use the work package to figure out how to continue the work of the task with minimal lost time.

Resource Loading

Purpose of a Work Package

- In addition to the task descriptions, the package includes start and end dates for the task.
- The work package also can be adapted to status reporting (measure what percent of the task is complete).

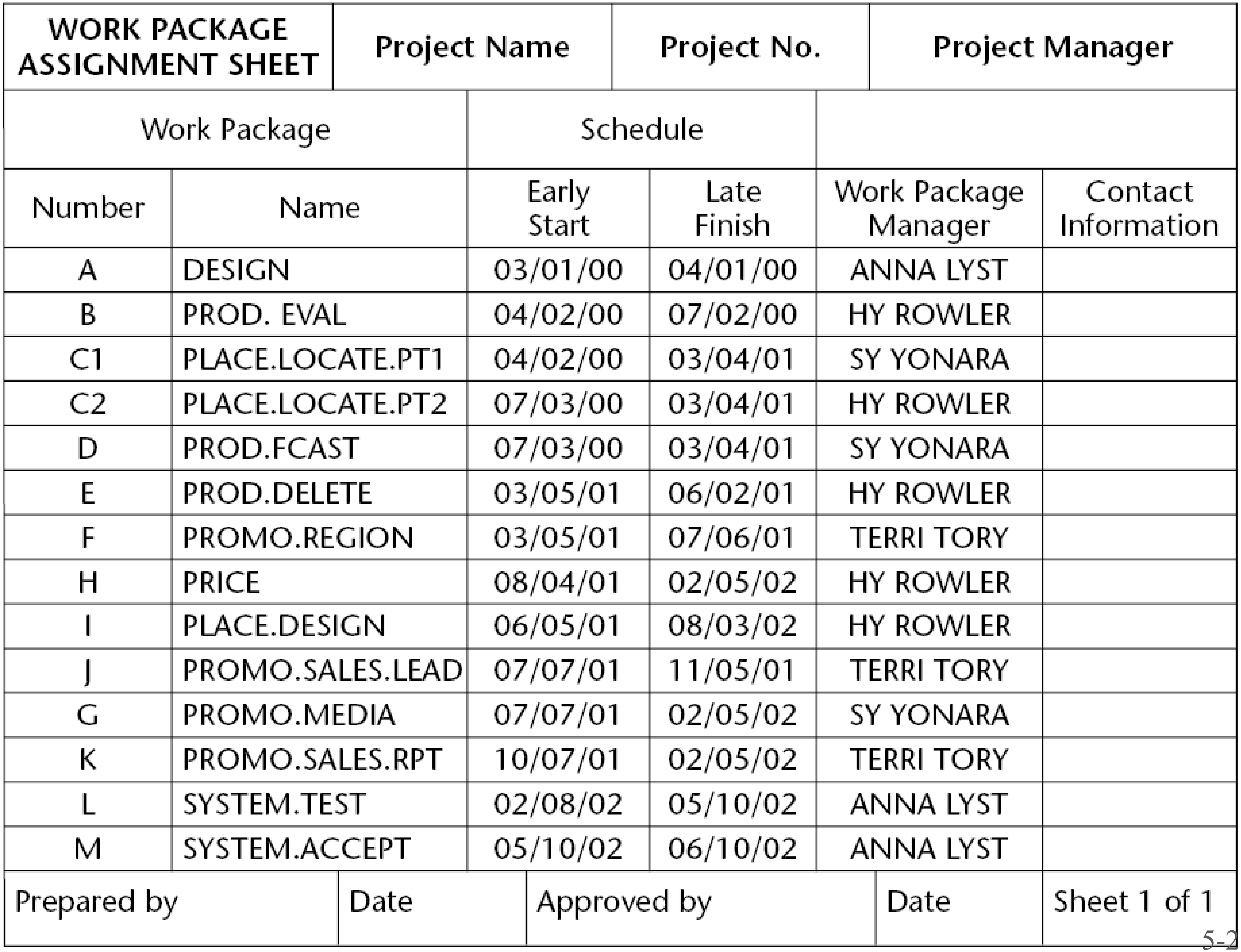
Format of a Work Package

Work package assignment sheet

- It contains some basic information about each work package and its manager. Task managers should be given only the scheduled start and end dates for their tasks.
- It is a report for the project manager only.

Work package description report

- A detailed description of the task plan.

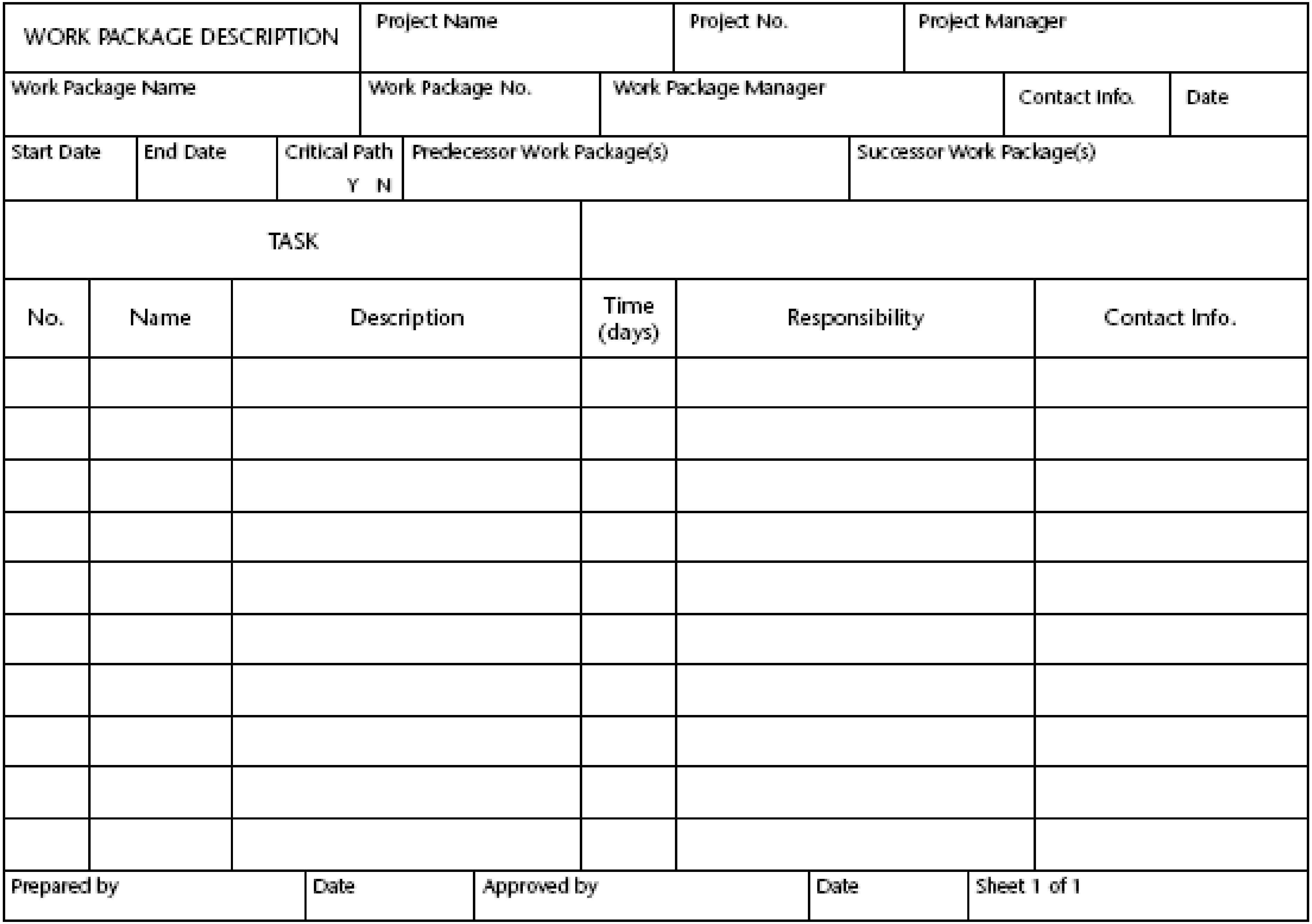


WORK PACKAGE ASSIGNMENT SHEET		**Project Name**	**Project No.**		**Project Manager**
Work Package		Schedule			
Number	Name	Early Start	Late Finish	Work Package Manager	Contact Information
A	DESIGN	03/01/00	04/01/00	ANNA LYST	
B	PROD. EVAL	04/02/00	07/02/00	HY ROWLER	
C1	PLACE.LOCATE.PT1	04/02/00	03/04/01	SY YONARA	
C2	PLACE.LOCATE.PT2	07/03/00	03/04/01	HY ROWLER	
D	PROD.FCAST	07/03/00	03/04/01	SY YONARA	
E	PROD.DELETE	03/05/01	06/02/01	HY ROWLER	
F	PROMO.REGION	03/05/01	07/06/01	TERRI TORY	
H	PRICE	08/04/01	02/05/02	HY ROWLER	
I	PLACE.DESIGN	06/05/01	08/03/02	HY ROWLER	
J	PROMO.SALES.LEAD	07/07/01	11/05/01	TERRI TORY	
G	PROMO.MEDIA	07/07/01	02/05/02	SY YONARA	
K	PROMO.SALES.RPT	10/07/01	02/05/02	TERRI TORY	
L	SYSTEM.TEST	02/08/02	05/10/02	ANNA LYST	
M	SYSTEM.ACCEPT	05/10/02	06/10/02	ANNA LYST	
Prepared by	Date	Approved by		Date	Sheet 1 of 1

Resource Loading

Work Package Description Report

- Document prepared by the task manager in which he or she describes the details of how he or she will accomplish the work of the task.
- Not all tasks will require or should require work package documentation.
- The documentation can be limited to critical path tasks, near-critical path tasks, high-risk tasks, and tasks that use very scarce or highly skilled staff.



WORK PACKAGE DESCRIPTION	Project Name	Project No.	Project Manager

Work Package Name	Work Package No.	Work Package Manager	Contact Info.	Date

Start Date	End Date	Critical Path Y N	Predecessor Work Package(s)	Successor Work Package(s)

TASK					
No.	Name	Description	Time (days)	Responsibility	Contact Info.

Prepared by	Date	Approved by	Date	Sheet 1 of 1

Creating Resource Loading Charts

Prioritization Rules for Leveling

- ❑ Smallest amount of slack
- ❑ Smallest duration
- ❑ Lowest ID number (FCFS)
- ❑ Greatest number of successor tasks
- ❑ Requiring the most resources

Creating Resource Loading Charts

Procedure

1. Create a project activity **network diagram**
2. Develop resource **loading table**
3. Draw an initial **loading chart**
4. Rearrange activities within their slack to create a more **level profile.**

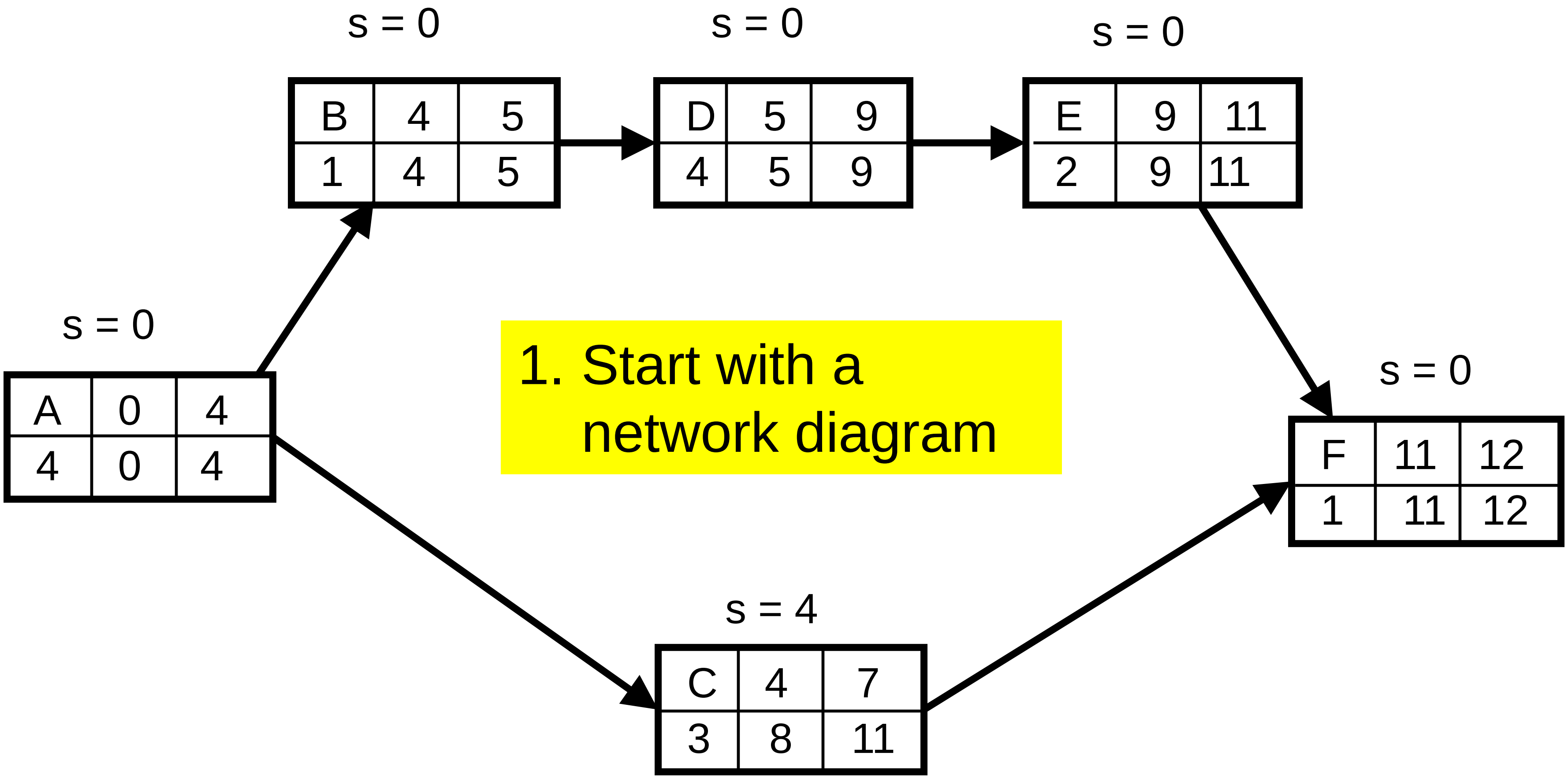
Creating Resource Loading Charts

Example

Activity	Priority	Duration	Resource
A	-	4	6
B	A	1	2
C	A	3	2
D	B	4	7
E	D	2	3
F	C, E	1	6

Creating Resource Loading Charts

AON diagram.



1. Start with a network diagram

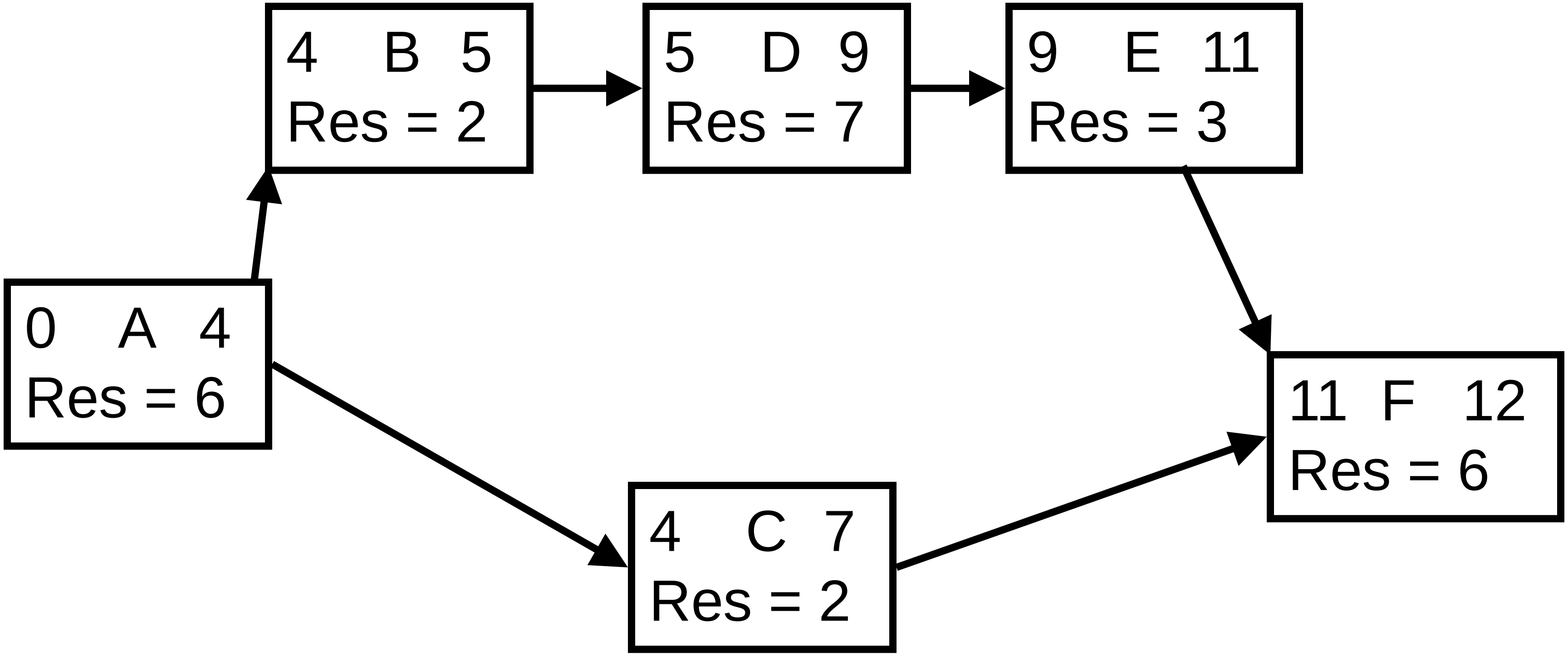
Creating Resource Loading Charts

Activity	Resource	Duration	ES	Slack	LF
A	6	4	0	0	4
B	2	1	4	0	5
~~C~~	~~2~~	~~3~~	~~4~~	~~4~~	~~11~~
D	7	4	5	0	9
E	3	2	9	0	11
F	6	1	11	0	12

2. Produce a table that shows the duration, Early Start, Late Finish, Slack, and Resource(s) required for each Activity.

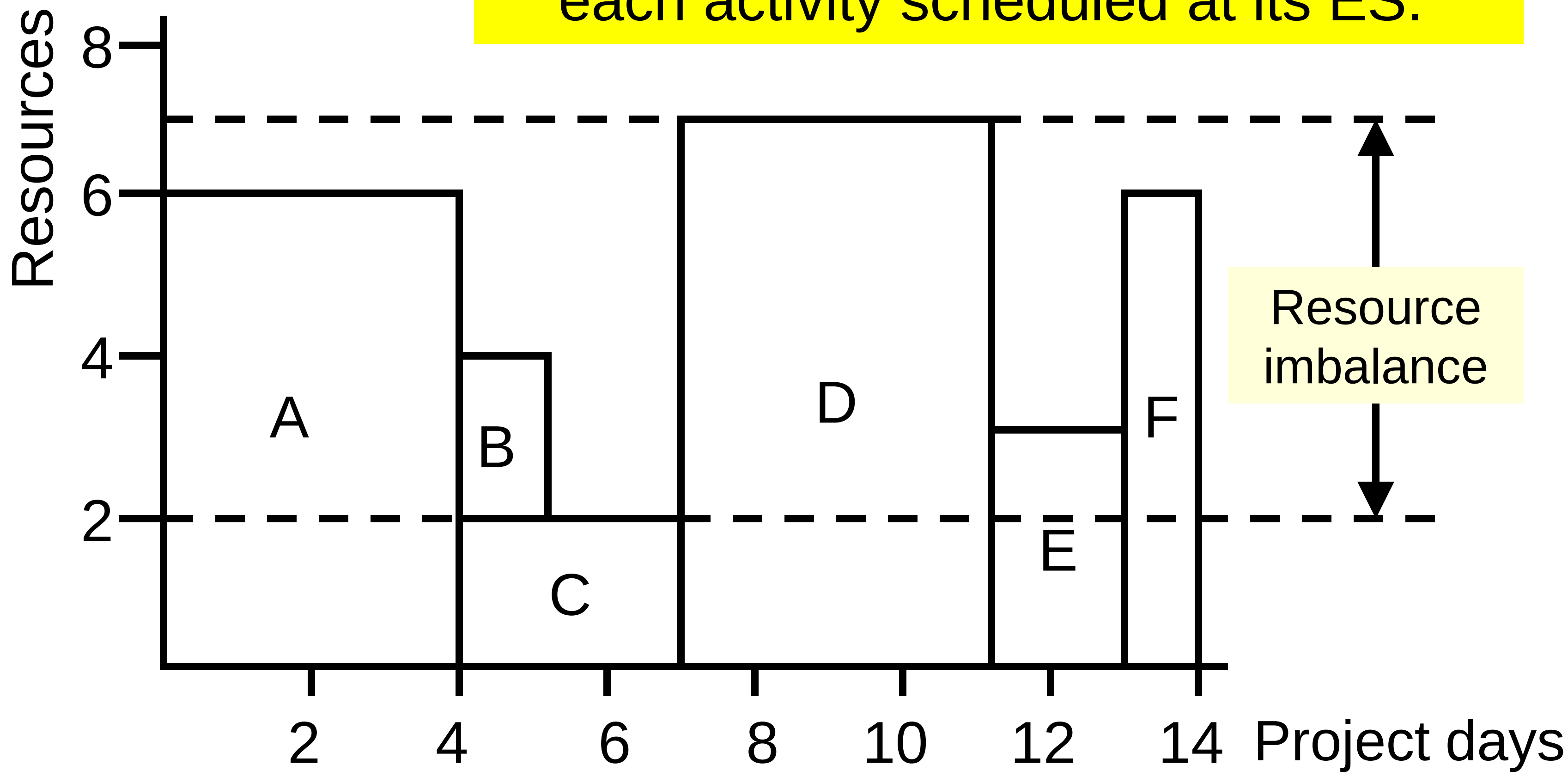
Creating Resource Loading Charts

Display the amount of resources required as a function of time.



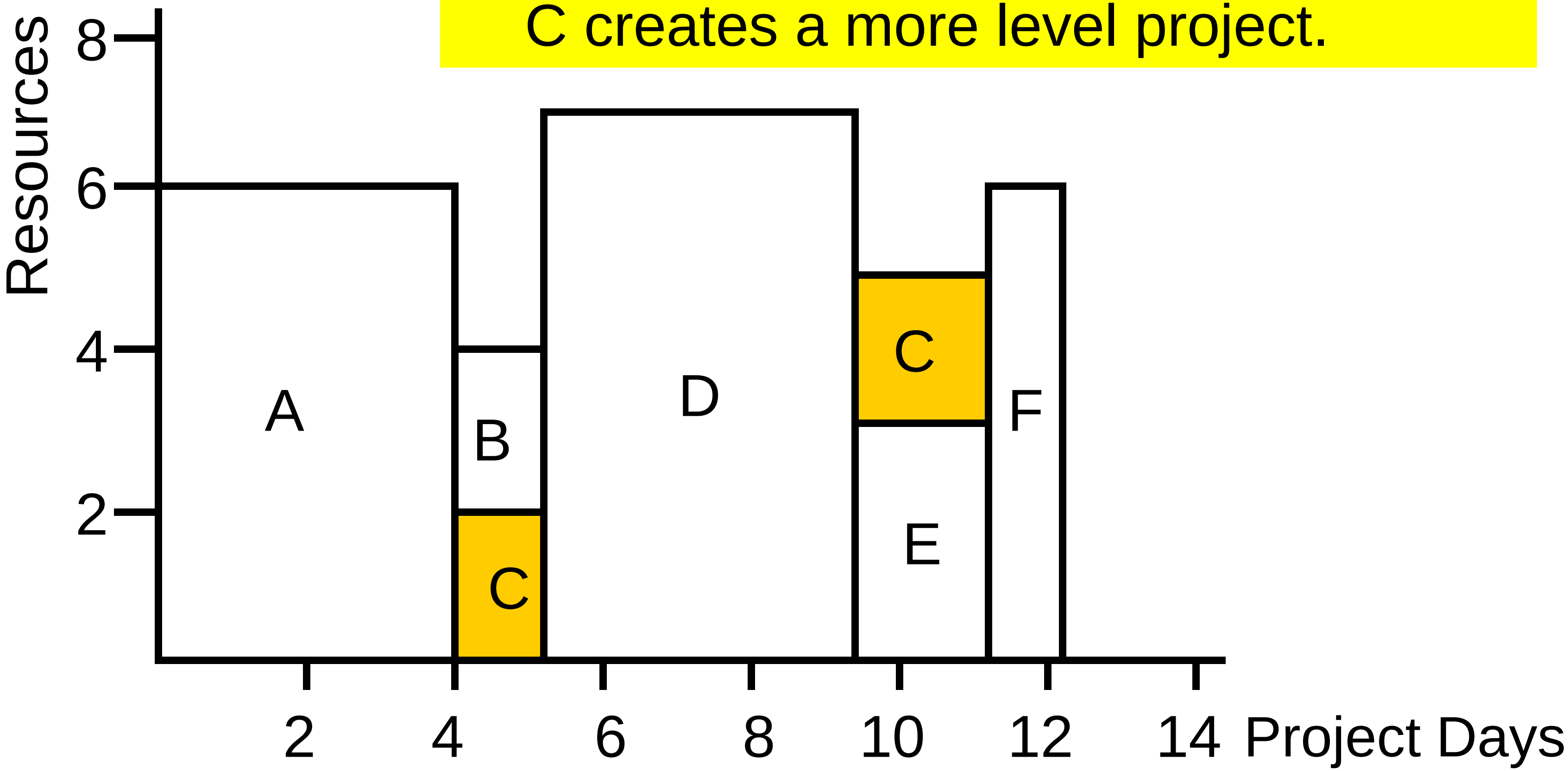
Creating Resource Loading Charts

3. Draw an initial loading chart with each activity scheduled at its ES.



Creating Resource Loading Charts

4. Rearrange activities within their slack to create a more level profile. Splitting C creates a more level project.



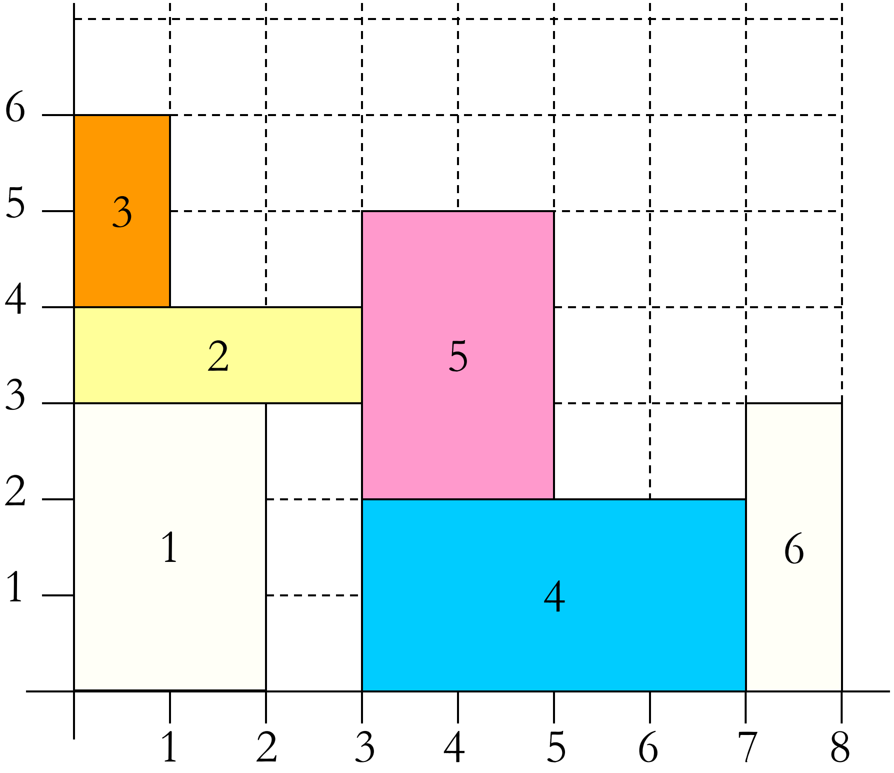
Creating Resource Loading Charts

Suppose jobs require a resource

Activity	p(j)	Priority	ES	LF	R(1,j)
1	2	-	0	3	3
2	3	-	0	3	1
3	1	-	0	6	2
4	4	1,2	3	7	2
5	2	2,3	3	5	3
6	1	4	7	8	3

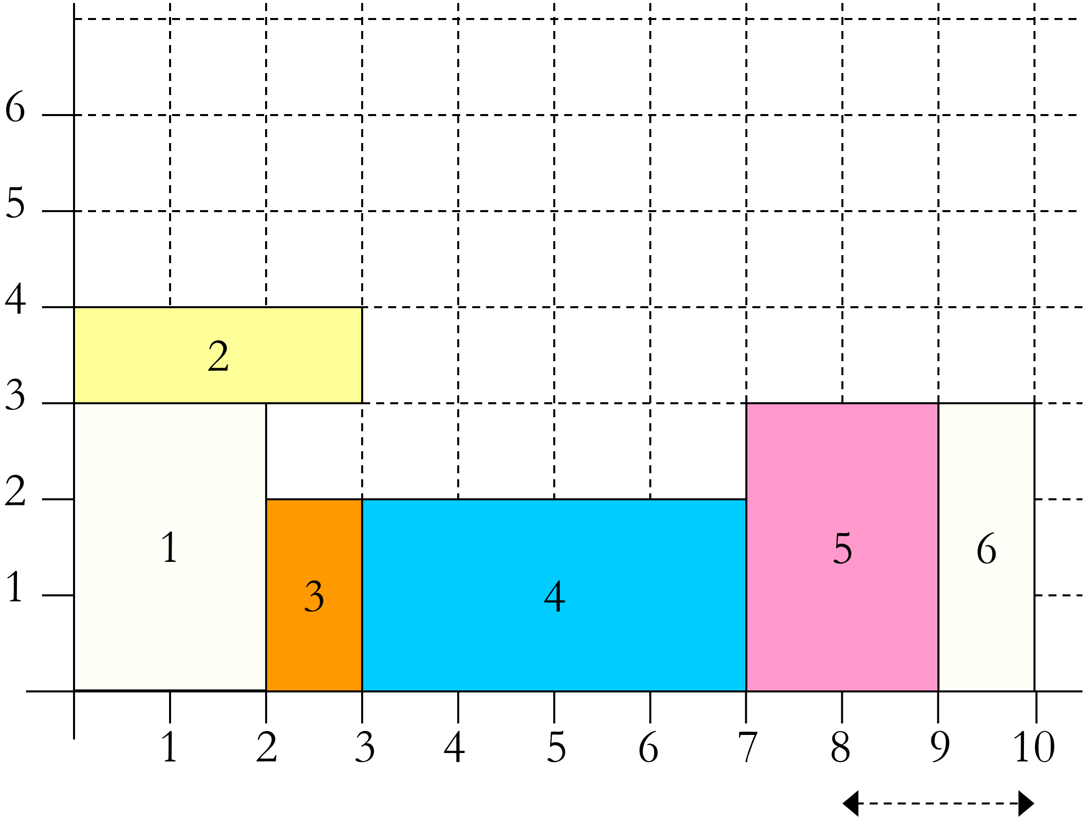
↑

Resource required



Creating Resource Loading Charts

Reponse $R_1 = 4$

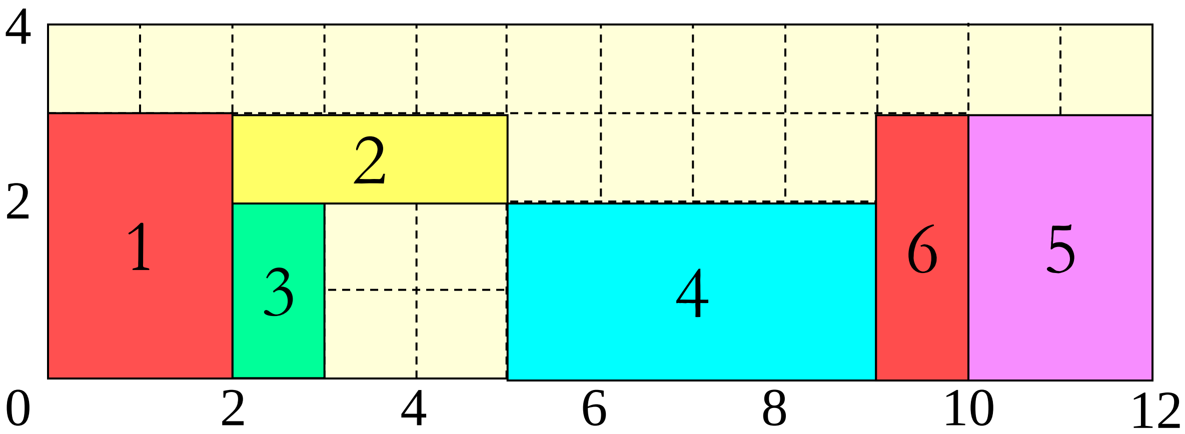


Cmax increases by 2

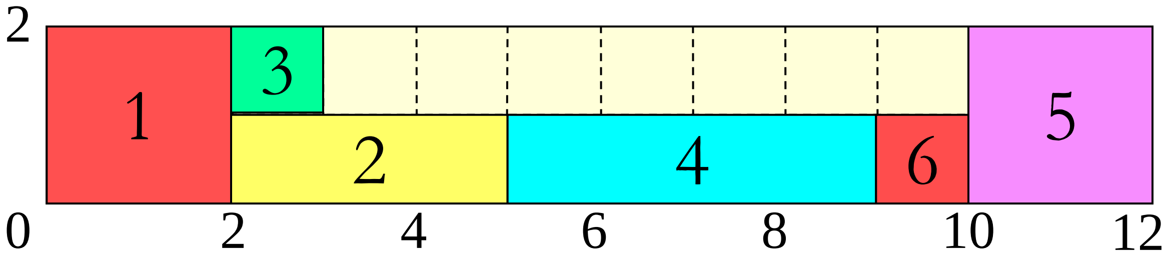
Creating Resource Loading Charts

Act.	p(j)	P(j)	ES	LF	R(1,j)	R(2,j)
1	2	-	0	2	3	2
2	3	-	0	3	1	1
3	1	-	0	1	2	1
4	4	1,2	3	7	2	1
5	2	2,3	3	5	3	2
6	1	4	7	8	3	1

R_1 =3



R_1 =2



Constraints for Resource

General

- **Serial** ←
- Parallel

Priority rule

- latest finish time
- minimum slack

Sampling procedure.

Constraints for Resource

Serial

- Each stage represents a job ⇒ n stages
- Completed set of jobs: scheduled jobs
- Decision set: jobs of which all predecessors have been scheduled
- Remaining set: other jobs

Process

1. Start with an empty schedule

2. Select job from decision set with highest priority, and schedule it as early as possible

3. Repeat step 2 if the decision set is not empty.

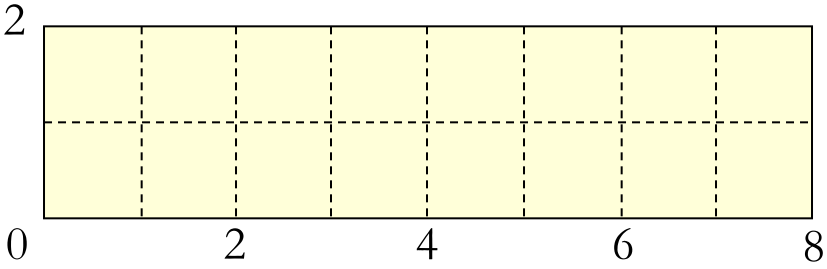
Constraints for Resource

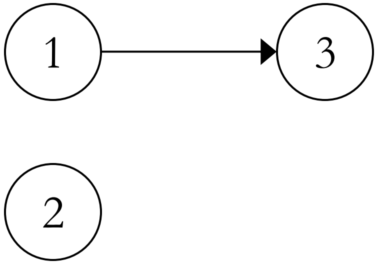
Serials

Act.	p(j)	P(j)	R(1,j)	v(j) (priority)
1	2	-	1	2
2	3	-	1	1
3	3	1	2	2

Decision set

$R_1 = 2$





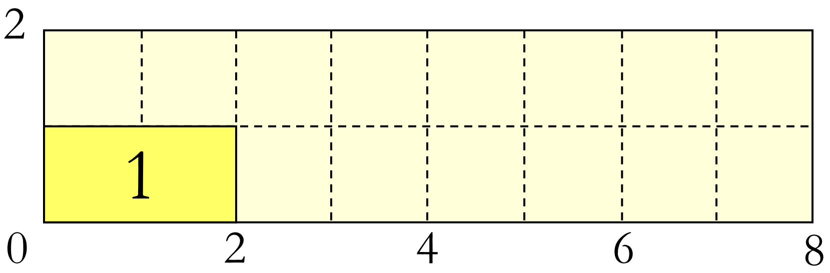
Constraints for Resource

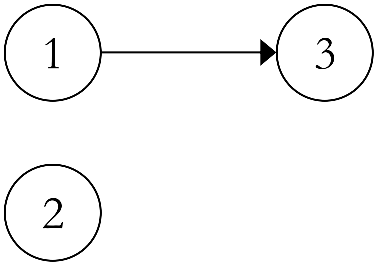
Serial

Job	p(j)	P(j)	R(1,j)	v(j) (priority)
1	2	-	1	2
2	3	-	1	1
3	3	1	2	2

Decision set ←

$R_1 = 2$





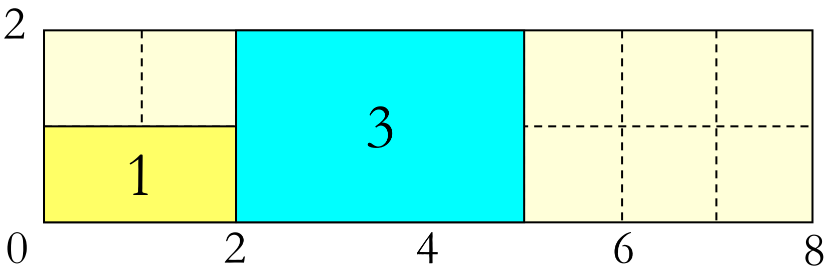
Constraints for Resource

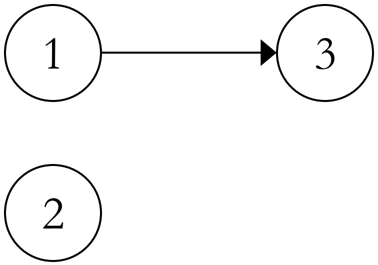
Serial

Job	p(j)	P(j)	R(1,j)	v(j) (priority)
1	2	-	1	2
2	3	-	1	1
3	3	1	2	2

Decision set

R_1 = 2





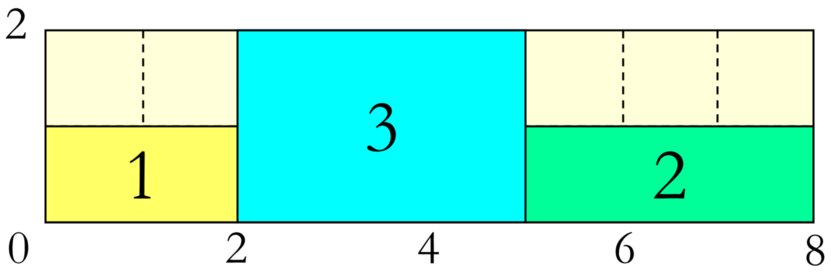
Constraints for Resource

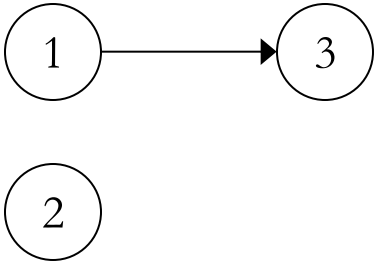
Serial

Job	p(j)	P(j)	R(1,j)	v(j) (priority)
1	2	-	1	2
2	3	-	1	1
3	3	1	2	2

Decision set ←

$R_1 = 2$





Constraints for Resource

Parallel

Each stage n represents:

1. partial schedule
2. schedule time t_n
3. four disjoint sets of jobs:
   - completed set: scheduled jobs, completed at t_n
   - active set: scheduled jobs, not completed yet
   - decision set: all unscheduled jobs, that could be scheduled
   - remaining set: all unscheduled jobs, that cannot be scheduled

Constraints for Resource

Parallel

Process

1. Start with an empty schedule.

2. Let T be the first time in which an unscheduled job may start. Let D be the collection of jobs that may be started on T, and of which all predecessors are scheduled

3. Select the job from D with the highest priority, and schedule it from time T

4. Repeat step 2 if there remain jobs to be scheduled

Constraints for Resource

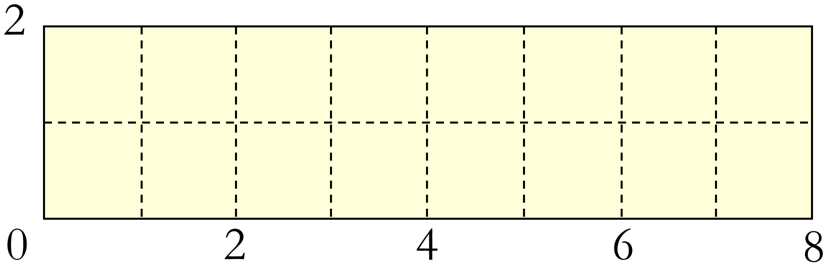
Parallel

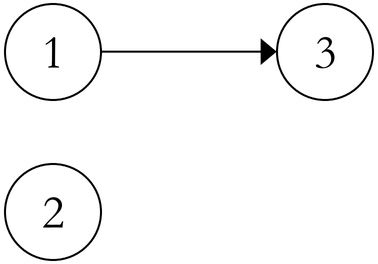
Job	p(j)	P(j)	R(1,j)	v(j) (priority)
1	2	-	1	2
2	3	-	1	1
3	3	1	2	2

Decision set

$T = 0$

$\mathbf{R_1 = 2}$





Constraints for Resource

Parallel

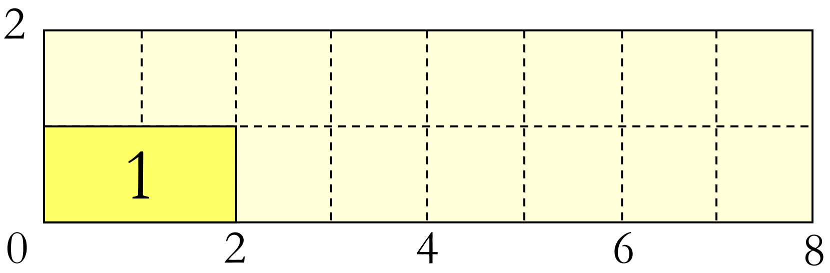
Job	p(j)	P(j)	R(1,j)	v(j) (priority)
1	2	-	1	2
2	3	-	1	1
3	3	1	2	2

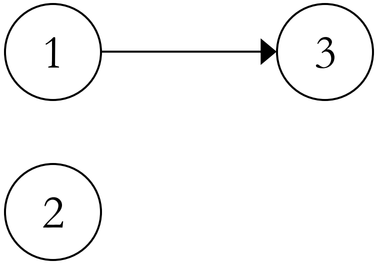
Decision set

←

$T = 0$

$\mathbf{R_1 = 2}$





Constraints for Resource

Parallel

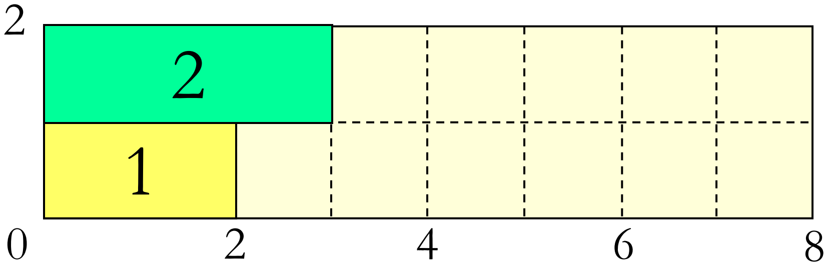
Job	p(j)	P(j)	R(1,j)	v(j) (priority)
1	2	-	1	2
2	3	-	1	1
3	3	1	2	2

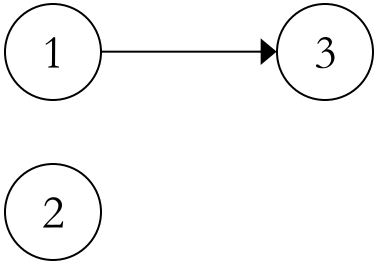
Decision set

←

$T = 3$

$R_1 = 2$



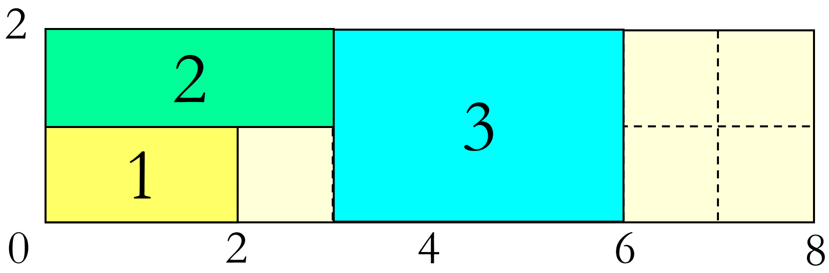


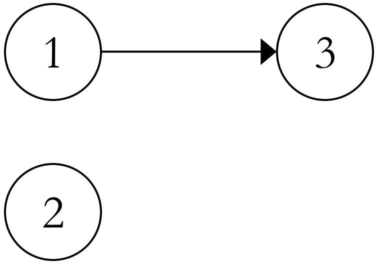
Constraints for Resource

Parallel

Job	p(j)	P(j)	R(1,j)	v(j) (priority)
1	2	-	1	2
2	3	-	1	1
3	3	1	2	2

$R_1 = 2$





Constraints for Resource

Generation scheme

- serial
- parallel

Priority rule ←

- latest finish time
- minimum slack

Sampling procedure

Constraints for Resource

- Latest finish time (LFT) priority rule:

$$v_j = -C_j''$$

- Minimum slack (MS) priority rule

$$v_j = -(C_j'' - p_j - \underset{\substack{\uparrow \\ \text{current earliest starting time}}}{S_j^{'*}})$$

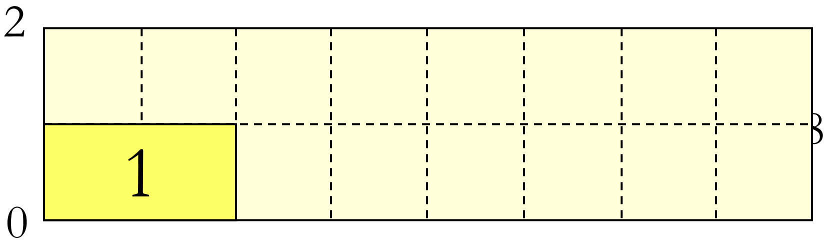
MS priority rule example (1)

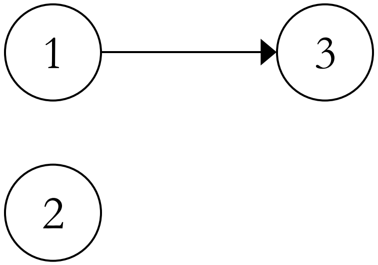
serial scheduling scheme

Job	p(j)	P(j)	R(1,j)	S'(j)	C''(j)	v(j) (priority)
1	2	-	1	0	2	0
2	3	-	1	0	3	-2
3	3	1	2	2	5	0

$$v_j = -(C''_j - p_j - S'^{*}_j)$$

$$R_1 = 2$$



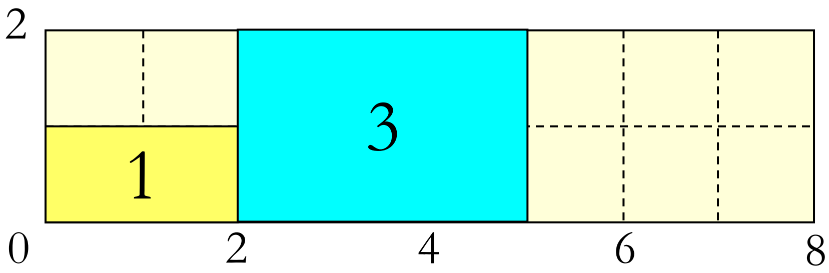


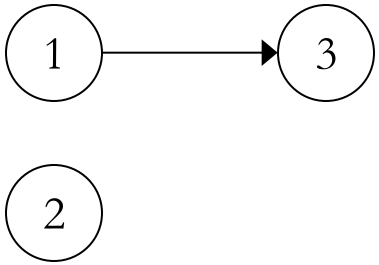
MS priority rule example (2)

Job	p(j)	P(j)	R(1,j)	S'(j)	C''(j)	v(j) (priority)
1	2	-	1	0	2	
2	3	-	1	0	5	-2
3	3	1	2	2	5	0

$$v_j = -(C''_j - p_j - S'^{*}_j)$$

$$R_1 = 2$$



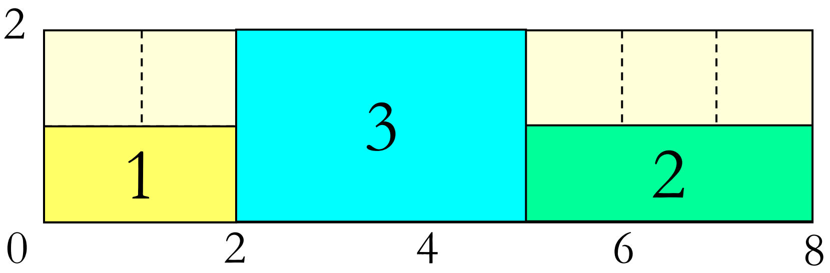


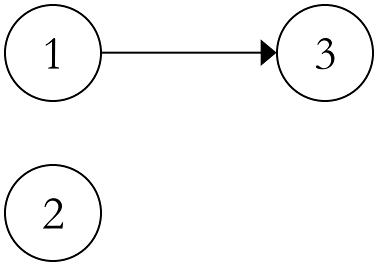
MS priority rule example (3)

Job	p(j)	P(j)	R(1,j)	S'(j)	C''(j)	v(j) (priority)
1	2	-	1	0	2	
2	3	-	1	5	5	3
3	3	1	2	2	5	

$$v_j = -(C_j'' - p_j - S_j'^{*})$$

$$R_1 = 2$$





Key Parameters in Multi-Project Environments

- Schedule slippage
- Resource utilization
- In-process inventory

Prioritizing Resource Allocations in Multi-Project Environments

First come first served

Greatest resource **demand**

Greatest resource **utilization**

Minimum **late finish** time

Mathematical programming

Exercises

House Building Project Data

No.	Activity	Predecessor	Duration (Mo)	Res. Requi.
1.	Design house and obtain financing	-	3	2
2.	Lay foundation	1	2	5
3.	Order Materials	1	1	2
4.	Build house	2, 3	3	10
5.	Select paint	2, 3	1	1
6.	Select carpet	5	1	1
7.	Finish work	4, 6	1	6

Resource Loading chart (4 steps)?