Lecture Notes on

Managing Info-gap Duration-Uncertainties in Projects

Yakov Ben-Haim
Yitzhak Moda'i Chair in
Technology and Economics
Faculty of Mechanical Engineering
Technion — Israel Institute of Technology
Haifa 32000 Israel
http://www.technion.ac.il/yakov
yakov@technion.ac.il

¶ Source material:

• Yakov Ben-Haim, 2001, *Information-Gap Decision Theory: Decisions Under Severe Uncertainty,* Academic Press, section 3.3.4, chapter 10.

• Yakov Ben-Haim and Alexander Laufer, 1998, Robust reliability of projects with activity-duration uncertainty, *ASCE Journal of Construction Engineering and Management.* 124: 125–132.

• Alexander Laufer and Yakov Ben-Haim, 1998, Robust reliability in project scheduling with time buffering, TME 469.

A Note to the Student: These lecture notes are not a substitute for the thorough study of books. These notes are no more than an aid in following the lectures.

Contents

[0]lectures prj_mgt.tex 8.3.2005

1 Basic Problem

¶ A project is characterized by:

- A flow-chart of tasks.
- Uncertainty in the duration of each task.
  (Alternatively: cost uncertainty.)
- Global requirement: complete project on time.

¶ Questions:

- How robust is the project to task-duration uncertainty?
- How risky is the project?
- How can the robustness be increased (and the risk reduced)?
  - Re-structuring the project.
  - On-line monitoring.
  - Gathering information.
- How opportune is the project?
  Can windfalls be exploited?

2 Project Reliability with a Global Time Buffer: Theory

¶ Consider a project whose task flow chart is:

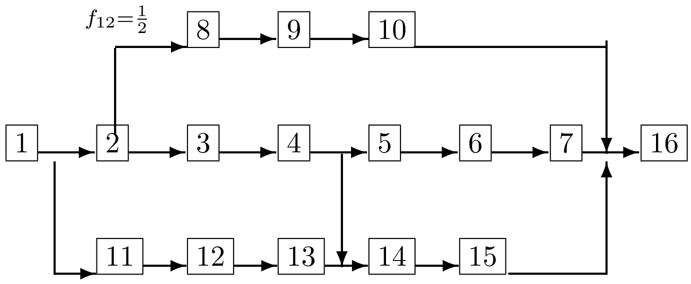

Figure 1: A 16-activity project schedule. Trans. p.blue11

This project has 4 task paths (Trans. p.blue11):
Path 1: $1 \rightarrow 2 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 16$.
Path 2: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 16$.
Path 3: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 14 \rightarrow 15 \rightarrow 16$.
Path 4: $1 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16$.

¶ In order to answer the questions in section 1 on page 3 we need:

- Dynamic model: describing the task-path structure and its relation to total project duration.
- Failure criterion.
- Uncertainty model.

¶ We first consider the **dynamic model.**
t_n = unknown duration of nth task, $n = 1, \ldots, N$.
$t = (t_1, \ldots, t_N)^T$
There are M paths.

f_{mn} = fractional participation of task n in path m.
m: path.
n: task.
In path m, the task following task n
begins when task n is fraction f_{mn} complete.

¶ E.g., in path 1 of fig. 1:
task 8 begins when task 2 is 1/2 complete:
$f_{12} = 0.5$.

¶ The duration of the m path, c_m,
equals the sum of the durations of **all tasks**
weighted by their fractional participations in path m:

$$c_m = \sum_{n=1}^{N} f_{mn} t_n, \quad m = 1, \ldots, M \tag{1}$$

For instance, the duration of the 1st path is:

$$c_1 = 1 \cdot t_1 + \frac{1}{2} \cdot t_2 + 1 \cdot t_8 + 1 \cdot t_9 + 1 \cdot t_{10} + 1 \cdot t_{16} \tag{2}$$

Define F = matrix of participation factors $f_{mn} \in \Re^{M \times N}$.
For instance, for fig. 1 (Trans. p.blue12):

$$F = \begin{pmatrix} 1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{3}$$

¶ Now the relation between task- and path-durations is:

$$c = Ft \tag{4}$$

The **dynamic model** is the duration of the longest path:

$$T = \|c\| = \max_{1 \le m \le M} |c_m| = \max_{1 \le m \le M} \sum_{n=1}^{N} f_{mn} t_n \tag{5}$$

Note that $\|c\|$ is in fact a vector norm, sometimes called the "zero norm".

¶ The **failure criterion:**
the project fails if the duration of the longest path
exceeds a critical value:

$$T > t_{\mathrm{c}} \tag{6}$$

¶ **Uncertainty model:** weighted fractional variations of task times.

$$\mathcal{U}(\alpha, \tilde{t}) = \left\{ t : \ \frac{|t_n - \tilde{t}_n|}{\tilde{t}_n} \le w_n \alpha, \quad n = 1, \ldots, N \right\}, \quad \alpha \ge 0 \tag{7}$$

¶ This is a family of nested sets.
Two levels of uncertainty:

- At fixed α: t_n, $n = 1, \ldots, N$ are uncertain.

$$\tilde{t}_n - w_n \tilde{t}_n \alpha \le t_n \le \tilde{t}_n + w_n \tilde{t}_n \alpha \tag{8}$$

- α, the **uncertainty parameter,** is unknown:
  unknown horizon of uncertainty.

¶ Robustness function:

$$
\begin{aligned}
\widehat{\alpha} &= \max\ \alpha \text{ which precludes failure} && (9)\\
&= \max\left\{\alpha:\ \text{failure is not possible}\right\} && (10)\\
&= \max\left\{\alpha:\ T \le t_{\rm c} \text{ for all } t \in \mathcal{U}(\alpha, \widetilde{t})\right\} && (11)\\
&= \max\left\{\alpha:\ \max_{1\le m\le M} \underbrace{\sum_{n=1}^{N} f_{mn} t_n}_{c_m} \le t_{\rm c} \text{ for all } t \in \mathcal{U}(\alpha, \widetilde{t})\right\} && (12)\\
&= \max\left\{\alpha:\ \max_{1\le m\le M} \max_{t\in\mathcal{U}(\alpha,\widetilde{t})} \sum_{n=1}^{N} f_{mn} t_n \le t_{\rm c}\right\} && (13)
\end{aligned}
$$

Recall that, for $t \in \mathcal{U}(\alpha, \widetilde{t})$:

$$
\widetilde{t}_n - w_n \widetilde{t}_n \alpha \le t_n \le \widetilde{t}_n + w_n \widetilde{t}_n \alpha \qquad (14)
$$

Thus:

$$
\begin{aligned}
\max_{t\in\mathcal{U}(\alpha,\widetilde{t})} c_m &= \max_{t\in\mathcal{U}(\alpha,\widetilde{t})} \sum_{n=1}^{N} f_{mn} t_n && (15)\\
&= \sum_{n=1}^{N} f_{mn} \left(\widetilde{t}_n + w_n \widetilde{t}_n \alpha\right) && (16)\\
&= \underbrace{\sum_{n=1}^{N} f_{mn} \widetilde{t}_n}_{\overline{c}_m} + \alpha \underbrace{\sum_{n=1}^{N} f_{mn} w_n \widetilde{t}_n}_{f_m} && (17)\\
&= \overline{c}_m + \alpha f_m && (18)
\end{aligned}
$$

The robustness is obtained by solving for α:

$$\max_{1\le m\le M} (\overline{c}_m + \alpha f_m) = t_{\rm c} \tag{19}$$

We can decompose this according to the separate paths:

$$\widehat{\alpha}_m = \text{robustness of path } m \tag{20}$$

which is the solution for α of:

$$(\overline{c}_m + \alpha f_m) = t_{\rm c} \tag{21}$$

which is:

$$\widehat{\alpha}_m = \frac{t_{\rm c} - \overline{c}_m}{f_m} \tag{22}$$

for each $m = 1, \ldots, M$.
One now sees that the project robustness is the lowest path-robustness:

$$\widehat{\alpha} = \min_{1\le m\le M} \widehat{\alpha}_m \tag{23}$$

$$= \min_{1\le m\le M} \frac{t_{\rm c} - \overline{c}_m}{f_m} \tag{24}$$

3 Example: Reliability as a Function of Global Time Buffer

¶ Consider the following data for $\tilde{t}$ and w:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\tilde{t}_n$	1	1	2	3	3	3	2	1	2	3	3	3	1	3	2	1
w_n	1	1	1	1	1	1	1	1	1	1	3	2	2	3	2	1

Table 1: Nominal durations and uncertainty-weights. (Trans. p.blue12)

With this data we can calculate:
$\hat{\alpha}_m$ = path robustnesses.
$\hat{\alpha}$ = overall project robustness:

t_c	α_1	α_2	α_3	α_4
16	0.88	**0.00**	0.14	0.063
18	1.12	**0.13**	0.24	**0.13**
20	1.35	0.25	0.33	**0.19**

Table 2: Path robustnesses with various allotted activity durations. (Trans. p.blue12)

¶ Note the following points:

- At $t_{\rm c} = 16$: $\widehat{\alpha}_2 = 0 \implies c_2 = 16$.
  Thus path 2 is the nominal-critical path.
- At $t_{\rm c} = 18$: $\widehat{\alpha}_2 = \widehat{\alpha}_4$.
  These two paths have the same robustness.
- At $t_{\rm c} = 20$: $\widehat{\alpha}_2 > \widehat{\alpha}_4$. Now:
  the uncertainty-critical path (path 4)

is different from

  the nominal-critical path (path 2).
- $\widehat{\alpha}$ increases monotonically,
  though not linearly,
  with $t_{\rm c}$.

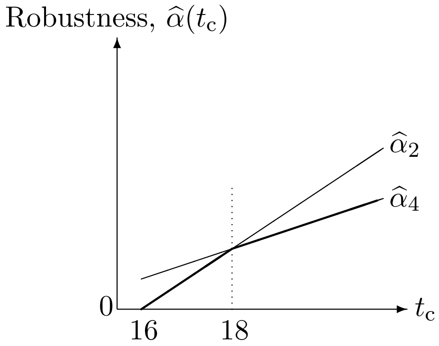

Figure 2: Trade-off of robustness $\widehat{\alpha}_m(t_{\rm c})$ against critical time $t_{\rm c}$, for two task paths.

4 Example: On-line Evaluation of Reliability

¶ We continue with the previous example.

We are 2.5 time units after project initiation:

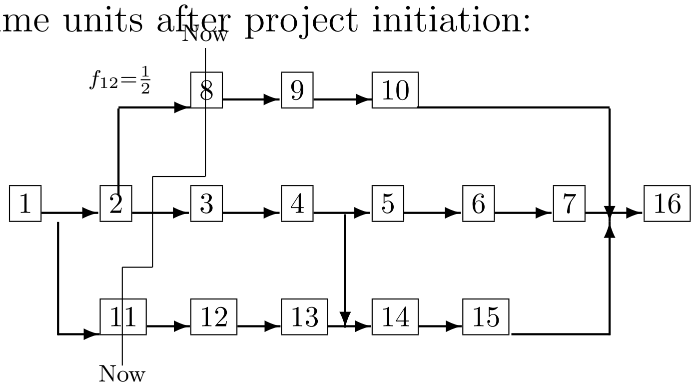

Figure 3: A 16-activity project schedule. The line labeled 'Now' indicates the current status of the project. (Trans. p.blue13)

¶ The situation:

- Task 1 completed after 1.5 time units: 0.5 unit over-run.
- Task 2 completed in 1 time unit as planned.
- Task 8 has been running 0.5 time unit.
- Task 11 has been running 1 time unit.

¶ New information:

- Task 8 will definitely end in 0.5 time unit.
- Uncertainty in task 11 is reduced somewhat.
- Uncertainty in tasks 5,6 & 14 is reduced substantially.

This new information is expressed in table 3:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\tilde{t}_n$	*0*	*0*	2	3	3	3	2	*0.5*	2	3	*2*	3	1	3	2	1
w_n	*0*	*0*	1	1	*0.5*	*0.5*	1	*0*	1	1	*2*	2	2	*1*	2	1

Table 3: Nominal durations and uncertainty-weights. (Trans. p.blue13)

We now obtain the following path robustnesses (table 4):

t_c Remaining Time	$t_c + 2.5$ Total Time	α_1	α_2	α_3	α_4
14	16.5	1.25	**0.00**	0.23	0.10
15.43	17.93	1.49	**0.13**	0.34	0.17
16.09	18.59	1.60	**0.19**	0.39	0.21

Table 4: Path robustnesses with various allotted activity durations, evaluated during project execution. (Trans. p.blue13)

¶ Note:

- Now path 2 is **always** critical.
- At $t_c = 16.5$: greater minimal time needed for completion due to the time over-run.
- At $t_c = 17.93$ (previous $t_c = 18$) and at $t_c = 18.59$ (previous $t_c = 20$) less time is needed than before due to the new information: reduced w_n values.

5 Enhancing Project Reliability

¶ We now consider enhancing project reliability with two types of strategies:

- Reducing uncertainty.
- Re-structuring the project.

5.1 Formulation

¶ Consider the following project flow chart:

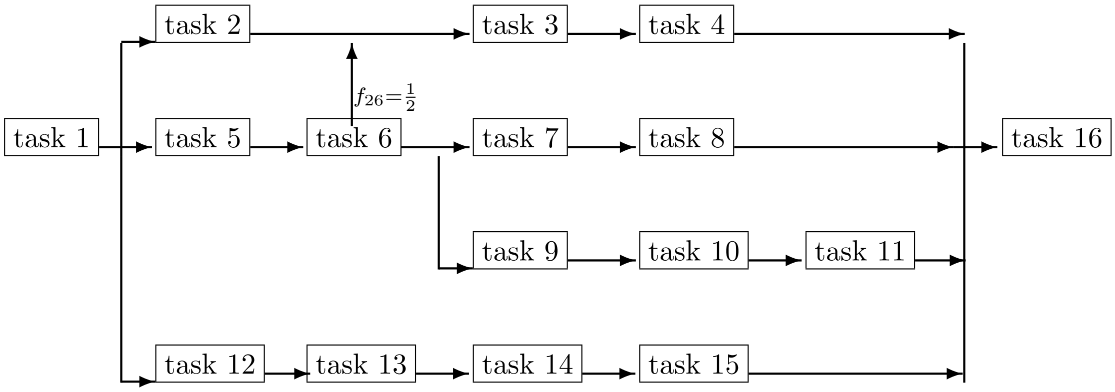

Figure 4: A 16-activity project schedule for section 5. (Trans. p.blue29)

¶ The project has 5 task paths (Trans. p.blue29):
Path 1: $1 \to 2 \to 3 \to 4 \to 16$.
Path 2: $1 \to 5 \to 6 \to 3 \to 4 \to 16$.
Path 3: $1 \to 5 \to 6 \to 7 \to 8 \to 16$.
Path 4: $1 \to 5 \to 6 \to 9 \to 10 \to 11 \to 16$.
Path 5: $1 \to 12 \to 13 \to 14 \to 15 \to 16$.
¶ Following is the participation matrix (Trans. p.blue29):

$$F = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad (25)$$

¶ The dynamical model is the duration of the longest path:

$$T = \max_{1 \le m \le M} \sum_{n=1}^{N} f_{mn} t_n \tag{26}$$

¶ The failure criterion is:

$$T > t_{\mathrm{c}} \tag{27}$$

¶ The uncertainty model is:

$$\mathcal{U}(\alpha, \tilde{t}) = \left\{ t : \ \frac{|t_n - \tilde{t}_n|}{\tilde{t}_n} \le w_n \alpha, \quad n = 1, \ldots, N \right\}, \quad \alpha \ge 0 \tag{28}$$

¶ Robustness of mth path:

$$\hat{\alpha}_m = \text{maximum } \alpha \text{ without failure of } m\text{th path} \tag{29}$$

$$= \max \left\{ \alpha : \ \underbrace{\sum_{n=1}^{N} f_{mn} \tilde{t}_n}_{\tilde{c}_m} + \alpha \underbrace{\sum_{n=1}^{N} f_{mn} w_n \tilde{t}_n}_{f_m} \le t_{\mathrm{c}} \right\} \tag{30}$$

$$= \max \left\{ \alpha : \ \tilde{c}_m + \alpha f_m \le t_{\mathrm{c}} \right\} \tag{31}$$

So:

$$\hat{\alpha}_m = \text{robustness of path } m \tag{32}$$

$$= \frac{t_{\mathrm{c}} - \tilde{c}_m}{f_m} \tag{33}$$

Hence the project robustness is:

$$\hat{\alpha} = \min_{1 \le m \le M} \hat{\alpha}_m \tag{34}$$

¶ The data for this project are:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\widetilde{t}_n$	1	4	6	3	2	3	5	4	4	2	1	2	1	3	1	2
w_n	1	2	2	2	1	2	1	1	0.5	1	1	1	1	1	1	1

Table 5: Nominal durations and uncertainty-weights. (Trans. p.blue30)

The resulting path robustnesses are:

t_c	α_1	α_2	α_3	α_4	α_5
17	0.035	0.058	**0.00**	0.13	0.70
19	**0.10**	0.14	**0.10**	0.25	0.90
21	**0.17**	0.21	0.20	0.38	1.10

Table 6: Path robustnesses with various allotted activity durations. (Trans. p.blue30)

¶ Note:

- Path 3 is nominal-critical.
- At $t_c = 19$: $\widehat{\alpha}_1 = \widehat{\alpha}_3$. Other paths more robust.
- At $t_c = 21$: path 1 is uncertainty-critical path.
- Large range of robustnesses. E.g., at $t_c = 21$:
  $\widehat{\alpha}_1 = 0.17$, $\widehat{\alpha}_5 = 1.10$, $\frac{\widehat{\alpha}_1}{\widehat{\alpha}_5} = 6.5$.
  Meaning: some paths much more reliable than others.

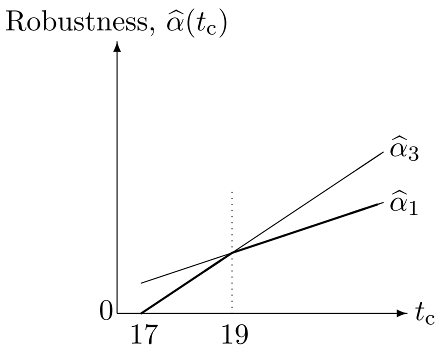

Figure 5: Trade-off of robustness $\widehat{\alpha}_m(t_c)$ against critical time t_c, for two task paths.

5.2 Enhancing Reliability by Reducing Uncertainty

¶ Gathering information reduces uncertainty.
We can express this by reducing the uncertainty weights w_n.
Fig. 6 shows all 5 paths vs w_6 (=2 in table 5).

Figure 6: α_m versus w_6. Symbols for paths 1 to 5: (1) solid; (2) dashed; (3) dot-dash; (4) dotted; (5) dash-dot-dot-dot. (Trans. p.blue30)

¶ Note:

- Only path-robustnesses $\widehat{\alpha}_2$, $\widehat{\alpha}_3$, $\widehat{\alpha}_4$ vary with w_6.
  Reason: only these paths involve task 6,
  as seen in column 6 in F, eq.(25) on p.15.
- The original critical path, #1, remains critical even at $w_6 = 0$.

¶ We can influence path 1 by gathering information about task 2, for which $w_2 = 2$ in table 5 on p.17.
Only path 1 depends on task 2 (See col. 2 of F, eq.(25) on p.15).
Fig. 7 shows $\hat{\alpha}_1$, $\hat{\alpha}_2$ and $\hat{\alpha}_3$ vs w_2.

Figure 7: α_m versus w_2. Symbols for paths 1 to 3: (1) solid; (2) dashed; (3) dot-dash. (Trans. p.blue31)

¶ Note:
- $\hat{\alpha}_1$ grows, but not much, as $w_2 \to 0$.
- Path 3 becomes critical for $w_2 \leq 1$.
  Thus not worth reducing $w_2 < 1$.

¶ Now gather information about path 3.
Explore effect of reducing w_5, w_6, w_7 and w_8.

¶ Suppose we are considering a short-term project,
so that individual task over-runs will be small, about %10.

We ask: How small do these w_n values have to be
in order to achieve the goal of $\widehat{\alpha} \approx \%10$?
We ask: What project duration is required?

t_c	α_1	α_2	α_3	α_4	α_5
$w_5 = w_6 = w_7 = w_8 = 2$					
17	0.035	0.054	**0.00**	0.11	0.70
19	0.10	0.13	**0.065**	0.22	0.90
21	0.17	0.20	**0.13**	0.33	1.10
$w_5 = w_6 = w_7 = w_8 = 1$					
17	0.035	0.061	**0.00**	0.15	0.70
19	**0.10**	0.14	0.12	0.31	0.90
21	**0.17**	0.22	0.24	0.46	1.10
$w_5 = w_6 = w_7 = w_8 = 0.5$					
17	0.035	0.066	**0.00**	0.19	0.70
19	**0.10**	0.15	0.20	0.38	0.90
21	**0.17**	0.24	0.40	0.57	1.10

Table 7: Path robustnesses with various allotted activity durations. (Trans. p.blue32)

¶ Table 7 shows trade-off between:
reducing uncertainty and extending project duration.

¶ **1st block:** $w_5 = \cdots = w_8 = 2$:
We achieve $\widehat{\alpha} = 0.13 (\approx 0.10)$ only at $t_c = 21$.
Path 3 is critical.

¶ **2nd block:** $w_5 = \cdots = w_8 = 1$:
We achieve $\widehat{\alpha} = 0.10$ at $t_c = 19$.
Path 1 is critical.

¶ **3rd block:** $w_5 = \cdots = w_8 = 0.5$:
No further improvement because:
- Path 1 is critical.
- Path 1 is independent of w_5, w_6, w_7 and w_8.

5.3 Enhancing Reliability by Re-structuring

In the original project structure, with $t_c = 21$,
path 1 is uncertainty-critical.

Path 1: $1 \to 2 \to 3 \to 4 \to 16$.

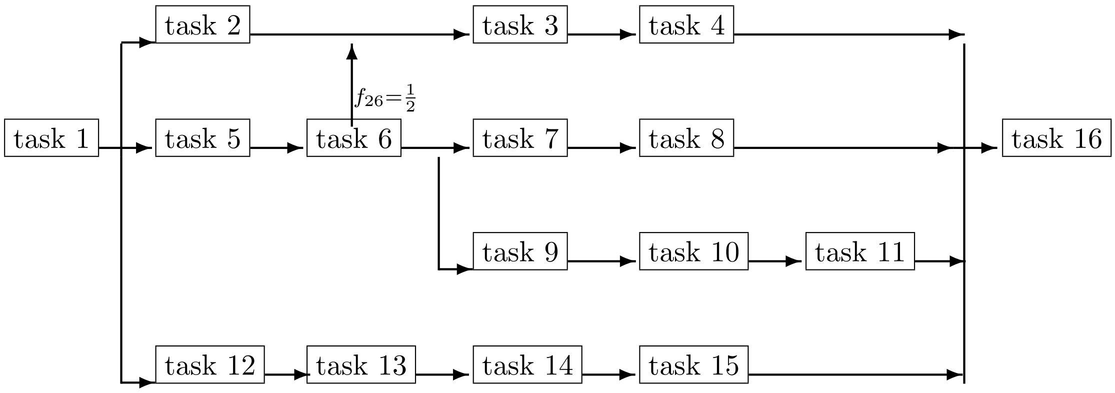

Figure 8: A 16-activity project schedule for section 5. (Trans. p.blue33)

Can we enhance reliability by restructuring this critical path?
Suppose we employ alternative technology to

partially overlap tasks 3 and 4.

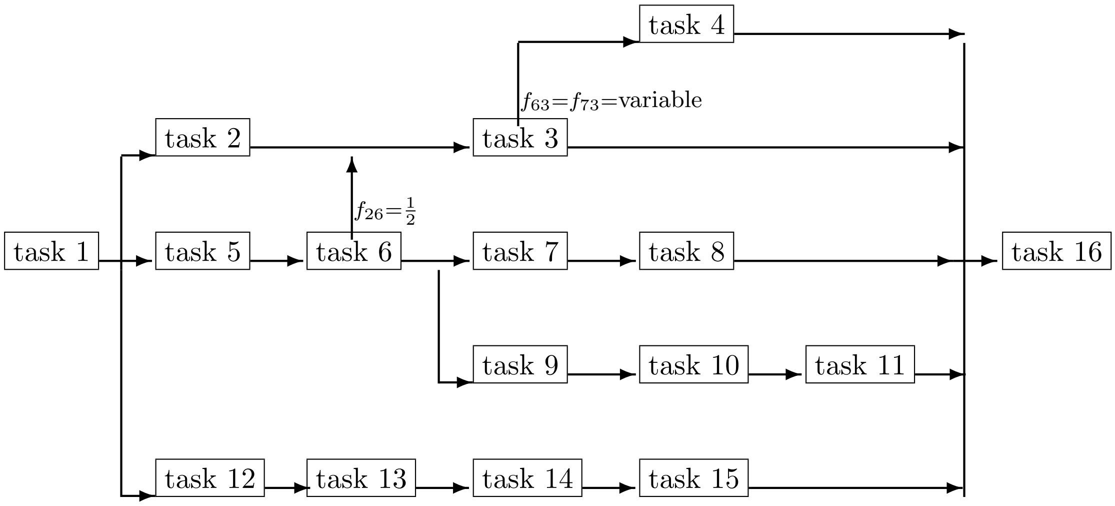

Figure 9: A revised 16-activity project schedule. (Trans. p.blue34)

We now have 7 paths (Trans. p.blue34):
Path 1: $1 \rightarrow 2 \rightarrow 3 \rightarrow 16$.
Path 2: $1 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 16$.
Path 3: $1 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 16$.
Path 4: $1 \rightarrow 5 \rightarrow 6 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 16$.
Path 5: $1 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16$.
Path 6: $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 16$.
Path 7: $1 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 4 \rightarrow 16$.
The participation matrix is (Trans. p.blue34):

$$F = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & f_{63} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & f_{73} & 1 & 1 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (35)$$

f_{63} = fractional participation of task 3 in path 6.
f_{73} = fractional participation of task 3 in path 7.

$$f_{63} = f_{73} \quad (36)$$

The robustnesses for these 7 paths are in table 8:

t_c	α_1	α_2	α_3	α_4	α_5	α_6	α_7
17	0.17	0.23	**0.00**	0.13	0.70	0.17	0.23
19	0.26	0.33	**0.10**	0.25	0.90	0.26	0.33
21	0.35	0.43	**0.20**	0.38	1.10	0.35	0.43

Table 8: Path robustnesses with various allotted project durations. $f_{63} = f_{73} = 0.5$. $t_c = 21$. (Trans. p.blue35)

¶ Note:

- Path 3 is critical at all values of t_c.
- Path 3 was unaffected by the restructuring:
  Path 3: $1 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 16$.
  which is the same as before the structural change.
- The restructuring "robustified" the altered paths,
  and transferred criticality to a previous non-critical path.

¶ We now consider the effect on path 6:

Path 6: $1 \to 2 \to 3 \to 4 \to 16$.

and compare with path 3 (critical path for $f_{63} = 0.5$):

Path 3: $1 \to 5 \to 6 \to 7 \to 8 \to 16$.

which is unaffected by the restructuring.

Recall:

$f_{63} = 1 \implies$ no overlap: task 4 starts when task 3 ends.

$f_{63} = 0 \implies$ full overlap: tasks 3 and 4 start together.

Figure 10: $\widehat{\alpha}_3$ and $\widehat{\alpha}_6(f_{63})$. $t_c = 21$. (Trans. p.blue35)

¶ Note:

- $\widehat{\alpha}_6$ increases as overlap increases ($f_{63}: \ 1 \to 0$).
  $\widehat{\alpha}_6(f_{63} = 1) = 0.17$. (No overlap)
  $\widehat{\alpha}_6(f_{63} = 0) = 0.65$. (full overlap)
  Substantial improvement with move from no- to full-overlap.
- $\widehat{\alpha}_3$ is constant since path 3 is unaffected by overlap.
- $\widehat{\alpha}_3 = 0.20$. and $\widehat{\alpha}_3 = \widehat{\alpha}_6$ at $f_{63} = 0.9$
  Hence: no increase in project reliability for overlap $> 10\%$.

¶ Now consider that the uncertainty in task 4
may increase with the degree of overlap.
Why? Because task 4 may depend on results obtained in task 3.

¶ So let w_4 increase with the degree of overlap:

$$\begin{aligned} w_4(f_{63} = 1) &= 2 \\ w_4(f_{63} = 0) &= 5 \end{aligned}$$

$w_4(f_{63})$ varies linearly with f_{63}.

Figure 11: α_3 and $\alpha_6(f_{63}, w_4)$. $t_c = 21$. (Trans. p.blue36)

¶ Note:

- $\widehat{\alpha}_6(f_{63} = 0) \ = \ 0.41$ as opposed to $\widehat{\alpha}_6(f_{63} = 0) \ = \ 0.65$ in fig. 10 on p.25.
  So improvement is still good, but not as good.
- $\widehat{\alpha}_3 = \widehat{\alpha}_6(f_{63})$ at very nearly the same f_{63} (~ 0.9).
  So virtually no impact on the transfer of criticality to path 3.
  Still, greatest useful overlap is $\sim$10%.

6 Enhancing Reliability with Local Time Buffers

¶ We now consider a multi-task project as before,
but now we are concerned with
local stability.
That is, we consider failure as:
time over-run of any individual task.
Of course, we are still concerned with over-all project duration.

¶ The basic idea is to allocate **local time buffers** to each task.

¶ Define:
t_c = duration for completion of project.
$\tilde{c}_m$ = nominal duration of path m.
Hence:
$t_c - \tilde{c}_m$ = amount of "buffer time" which can be allotted among the tasks of path m.
The question: how to distributed this buffer among the tasks?
We will formulate the basic ouline of this problem,
but we will not study its detailed solution.

¶ There are N tasks, for which:

$$t_n = \text{unknown } \textbf{actual} \text{ duration of task } n \tag{37}$$

$$t = (t_1, \ldots, t_N)^T \tag{38}$$

$$\tilde{t}_n = \text{known } \textbf{nominal} \text{ duration of task } n \tag{39}$$

$$\tilde{t} = (\tilde{t}_1, \ldots, \tilde{t}_N)^T \tag{40}$$

¶ The uncertainty model is, as before:

$$\mathcal{U}(\alpha, \tilde{t}) = \left\{ t : \ \frac{|t_n - \tilde{t}_n|}{\tilde{t}_n} \le w_n \alpha, \quad n = 1, \ldots, N \right\}, \quad \alpha \ge 0 \tag{41}$$

¶ Let b_n = **buffer time** following task n.
That is, b_n is the amount of spare time
during which we plan to be idle,
following completion of task n.
No delay results if task n completes during b_n.
Define:

$$b = (b_1, \ldots, b_N)^T \tag{42}$$

¶ The time over-run of task n is:

$$\delta_n(t_n) = \max\left\{t_n, \tilde{t}_n + b_n\right\} - (\tilde{t}_n + b_n) \tag{43}$$

¶ As before, we need 3 components for reliability analysis:

- Dynamic model of the system.
- Failure criterion.
- Uncertainty model: eq.(41) on p.28.

¶ **Failure:** If any single task exceeds it allotted time $\widetilde{t}_n + b_n$
by more than a specified amount $\Delta_{c,n}$.
That is, failure occurs if:

$$\max_{1 \le n \le N} [\delta_n(t_n) - \Delta_{c,n}] \ > \ 0 \tag{44}$$

$\Delta_{c,n}$ can be chosen as any non-negative value.
$\Delta_{c,n}$ can be different for different tasks.

¶ **Dynamic model:**
The failure criterion is applied "locally", at each task.
Hence the path structure does not directly affect success or failure.
The dynamic model is simply the vector t of task durations.

¶ **Robustness** of task n is the greatest tolerable value of α:

$$\widehat{\alpha}_n = \max \left\{ \alpha : \max_{t_n \in \mathcal{U}(\alpha, \widetilde{t})} \delta_n(t_n) \le \Delta_{c,n} \right\} \tag{45}$$

This is obtained by solving the following relation for α:

$$\max_{t_n \in \mathcal{U}(\alpha, \widetilde{t})} \delta_n(t_n) = \Delta_{c,n} \tag{46}$$

¶ Max over-run of task n, up to uncertainty α:

$$\max_{t_n \in \mathcal{U}(\alpha, \widetilde{t})} \delta_n(t_n) = \max \left\{ \widetilde{t}_n(1 + w_n \alpha), \ \widetilde{t}_n + b_n \right\} - (\widetilde{t}_n + b_n) \tag{47}$$

where we understand that:
$\widetilde{t}_n(1 + w_n\alpha)$ = greatest duration of task n allowed by $\mathcal{U}(\alpha, \widetilde{t})$.
$\widetilde{t}_n + b_n$ = greatest nominal duration of task n.

Hence the robustness of task n is:

$$\widehat{\alpha}_n = \frac{b_n + \Delta_{c,n}}{\widetilde{t}_n w_n} \tag{48}$$

The overall robustness of the project is:

$$\widehat{\alpha} = \min_{1\le n\le N} \widehat{\alpha}_n \tag{49}$$

$$= \min_{1\le n\le N} \frac{b_n + \Delta_{\mathrm{c},n}}{\widetilde{t}_n w_n} \tag{50}$$

¶ We would like to choose the buffer times b to maximize $\widehat{\alpha}$.
One approach is to use a 'Robin Hood' principle:

- Take buffer time away from very robust tasks.
- Give buffer time to very vulnerable tasks.
- Continue this until the robustnesses of the tasks are as equal as possible.

We will not pursue this optimization problem.