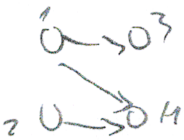
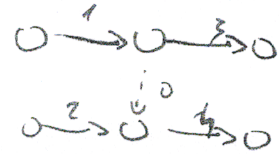
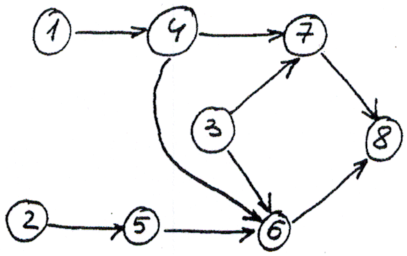
#4.2





#4.3

(a)



Jobs	1	2	3	4	5	6	7	8
p_j	4	6	10	12	10	2	4	2

Compute the earliest completion times C_j' using the forward procedure

Jobs	1	2	3	4	5	6	7	8
C_j'	4	6	10	16	16	18	20	22

This implies that the makespan is 22. Assuming the makespan is 22, compute the latest possible completion times C_j'' using the backward procedure.

Jobs	1	2	3	4	5	6	7	8
C_j''	4	8	16	16	18	20	20	22

The critical path is $1 \to 4 \to 7 \to 8$. In this case it is unique.

(b) Suppose $p_7' = 1$. Applying the forward procedure to the new data gives the following earliest completion times C_j':

Jobs	1	2	3	4	5	6	7	8
C_j'	4	6	10	16	16	18	17	20

The new makespan is 20.

Jobs	1	2	3	4	5	6	7	8
C_j''	4	6	16	16	16	18	18	20

In this case there are two critical paths:

$1 \to 4 \to 6 \to 8$

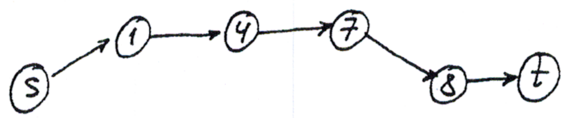
and

$2 \to 5 \to 6 \to 8$

#4.6

(a) Apply the Time/Cost Trade-Off Heuristic.

(i) Setting the processing times at their maxima, the $1 \to 4 \to 7 \to 8$ critical path is obtained. The subgraph G_{cp} of the critical paths is:



The initial makespan is 22.
[s, t are the source and sink nodes]

The minimal cuts are {1}, {4}, {7} and {8}

Cut {4} has the lowest cost of processing time reduction: $c_4 = 2$.
Reducing p_4 from 12 to 11 costs 2 and saves 6 in overhead, resulting in net savings of 4.

Jobs	1	2	3	4	5	6	7	8
C'_j	4	6	10	15	16	18	19	21
C''_j	4	7	15	15	17	19	19	21

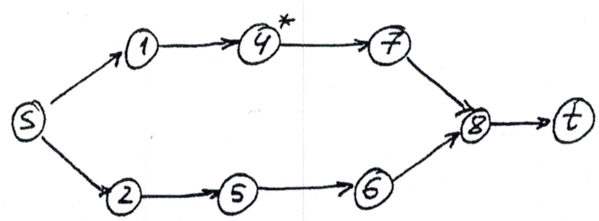
The new makespan is 21.

(ii) The critical path is still $s \to 1 \to 4 \to 7 \to 8$ and {4} has the lowest cost of processing time reduction. Reduce p_4 again by 1:

Jobs	1	2	3	4	5	6	7	8
C'_j	4	6	10	14	16	18	18	20
C''_j	4	6	14	14	16	18	18	20

The new makespan is 20.

G_{cp}:



Note: $\bigcirc^*$ means that the job is at its minimum processing time

(iii) Cut {8} has the lowest cost of processing time reduction: $c_8 = 4$.
Reducing p_8 from 2 to 1 costs 4 and saves 6 in overhead, resulting in net savings of 2.

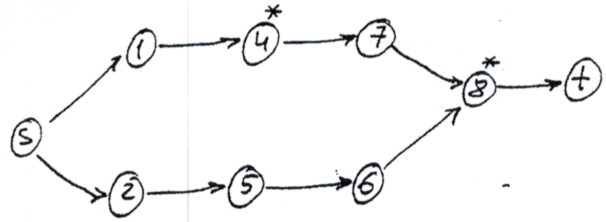
#2.6 (p.2)

Reduce p_8 by 1:

Jobs	1	2	3	4	5	6	7	8
C'_j	4	6	10	14	16	18	18	19
C''_j	4	6	14	14	16	18	18	19

The new makespan is 19.

G_{cp}:



The above graph doesnot have minimal cuts with a cost less than the overhead. Therefore, the current solution is <u>optimal</u>.

However, the optimal solution is <u>not unique</u>. For example, we could decrease both p_1 and p_6 by 1 without affecting the total cost:

Jobs	1	2	3	4	5	6	7	8
C'_j	3	6	10	13	16	17	17	18
C''_j	3	6	13	13	16	17	17	18

The makespan is 18.

#4.8

(a) and (b) Based on the given data, we have the following table:

Job	1	2	3	4	5	6	7	8	9	10	11
p_i^a	2	1	5	2	1	4	1	1	6	1	8
p_i^m	8	5	6	3	2	5	2	4	7	4	9
p_i^b	14	9	7	4	3	6	3	7	8	7	10
μ_i	8	5	6	3	2	5	2	4	7	4	9
σ_i	2	1.33	0.33	0.33	0.33	0.33	0.33	1	0.33	1	0.33
σ_i^2	4	1.78	0.11	0.11	0.11	0.11	0.11	1	0.11	1	0.11

There are 8 possible paths (look at Figure 2.5):

Path1: S-1-2-10-T; mean length (ML)=17, variance (VAR) = 6.78
Path2: S-1-9-11-T; ML = 24; VAR = 4.22
Path3: S-1-8-11-T; ML = 21; VAR = 5.11
Path4: S-3-5-8-11-T; ML = 21; VAR = 1.33
Path5: S-3-5-9-11-T; ML = 24; VAR = 0.44
Path6: S-4-6-9-11-T; ML = 24; VAR = 0.44
Path7: S-4-6-8-11-T; ML = 21; VAR = 1.33
Path8: S-4-7-11-T; ML = 14; VAR = 0.33

In decreasing order of ML we have: Path2, Path5, Path6, Path3, Path4, Path7, Path1, Path8.
In decreasing order of VAR we have: Path1, Path3, Path2, Path4, Path7, Path5, Path6, Path8.

(c) Following the PERT procedure:

$P(C_{max} > 27) = 1 - P(path1 < 27) * P(path2 < 27) * P(path3 < 27) * P(path4 < 27)$
$*P(path5 < 27) * P(path6 < 27) * P(path7 < 27) * P(path8 < 27)$=0.076

(d) Path 1 has the highest variance, therefore:

$P(Path1 > 27) = 0.0001$.