I. Minimizing Cmax (Nonpreemptive)

a. P2 || Cmax is NP-hard.

Partition is reducible to P2 || Cmax

b. P | Pj = 1, intree | Cmax

P | Pj = 1, outtree | Cmax are both solvable in polynomial time.

c. P2 | Pj = 1, prec | Cmax is solvable in polynomial time

P | Pj = 1, prec | Cmax is NP-hard

Pm | Pj = 1, prec | Cmax OPEN for fixed m ≥ 3

PARTITION problem

Given: A list A = (a1, a2, …, an) of n integers

Question: Can A be partitioned into A1 and A2 such that

Σ (aj ∈ A1) aj = Σ (aj ∈ A2) aj = ½ Σ (i = 1 to n) ai ?

b. P | Pj = 1, intree | Cmax & P | Pj = 1, outtree | Cmax are the same problem.

Critical-Path Rule (Hu’s Algorithm)

Whenever a machine becomes free for assignment, assign that unexecuted, ready job (if any) which is at the highest level among those unexecuted jobs. Ties may be resolved arbitrarily.
Notes
1. The root of the tree is the only job executed in the last time unit (i.e. \([\text{Cmax-1, Cmax}]\))

2. If the schedule is such that there is no idle time before \(\text{Cmax - 2}\), then the schedule must be optimal.

**Thm:** The Critical-Path rule solves the problem \(P | P_j = 1, \text{intree} | \text{Cmax}\)

**Proof:** By induction on the height \(h\) of the tree \((T)\)

**Basis:** Obvious when \(h = 1\) or \(h = 2\)

**Inductive Step:** Assume the theorem is true for all trees of height \(h = k\), we want to show that it is true for all trees of height \(h = k + 1\), the inductive step is proved by contradiction.

1. Suppose there is a tree \(T\) of height \(h = k + 1\) such that a Critical-Path schedule \(S\) is not of minimal makespan. Let \(\text{Cmax of } S\) be \(w\). Let the optimal makespan be \(w_o\).

   We have \(w_o < w\).
2. Form a new tree T’ from T by deleting all level 2 jobs and joining all level 3 jobs to the root.

3. Since S is not optimal and T is an intree, we can show that Cmax of the Critical-Path schedule S’ for T’ is w-1 and the optimal makespan for T’ is w₀ – 1.

   Complete proof in p. 3-3 of the textbook.

4. Since w₀ < w, w₀ – 1 < w – 1. Therefore, critical path rule is not optimal for T’ which has height h = k. Contradiction!! (to our assumption that it is optimal for all trees of height h = k).

c. P₂ | Pj = 1, prec | Cmax


Coffman-Graham Algorithm

1. Each job is assigned an integer which represents the priority of the job.

2. Whenever a machine becomes free for assignment, assign that job all of whose predecessors have already been executed and which has the largest label (priority) among those jobs not yet assigned.

Let N = (n₁, n₂, … , nₖ) and N’ = (n’₁, n’₂, … , n’ₖ’) be 2 decreasing sequences of integers, i.e. nᵢ > nᵢ₊₁ (1 ≤ i < t) and n’ⱼ > n’ⱼ₊₁ (1 ≤ j < t’). We say that N < N’ if either
1. For some $i$, $1 \leq i \leq t$, we have $n_j = n'_j$, for all $j$ satisfying $1 \leq i \leq i - 1$ and $n_i < n'_i$,

or

2. $n_j = n'_j$ for all $1 \leq j \leq t$ and $t < t'$

e.g. $(7, 5, 3, 2) < (7, 5, 4, 1)$

$(4, 3, 1) < (5, 1)$

$(7, 2) < (7, 2, 1)$

Coffman-Graham Labeling Algorithm

$\alpha = \{J_1, J_2, \ldots, J_n\} \rightarrow \{1, 2, \ldots, n\}$

For each job $J_i$, $S(J_i)$ denote the set of immediate successors of $J_i$.

1. Let there be $j$ terminal jobs (i.e. $S(J_i) = \phi$). The terminal jobs are assigned an integer from the set $\{1, 2, \ldots, j\}$ in an arbitrary order.

2. Suppose for some $k \leq n$ that the integers $1, 2, \ldots, k - 1$ have been assigned. For each job $J_i$ for which $\alpha$ has been assigned to all elements of $S(J_i)$, let $N(J_i)$ denote the decreasing sequence of integers formed by ordering the set $\{\alpha(J_j) | J_j \in S(J_i)\}$.

Assign $k$ to the job $J^*$ such that $N(J^*) \leq N(J_i)$ for all such jobs $J_i$.

3. Repeat step 2 until all jobs have been assigned some integers.
Proof that C-G algorithm solves the problem P2 | prec, Pj = 1 | C_{max}

Conventions:

1. If both machines become available simultaneously, machine 1 will always be assigned before machine 2.
2. If a machine is idle in a time interval, we say that it is executing a job with label 0.
Lemma: Let S be a C-G schedule and let t be any integer between 0 and $C_{\text{max}} - 1$ (i.e. $0 \leq t \leq C_{\text{max}} - 1$). If $J$ is the job executing in the interval $[t, t + 1]$ on machine 1, and $J' \neq J$ is another job executing in the interval $[t', t' + 1]$ where $t \leq t' \leq C_{\text{max}} - 1$, on any machine, then $\alpha(J) > \alpha(J')$.

Suppose we can form $k$ segments $(X_0, X_1, X_2, \ldots, X_{k-1})$

Claim 1: The number of jobs in each segment $X_i$ is odd. Let the number of jobs in segment $X_i$ be $2n_i - 1$. The time taken to execute all jobs in $X_i$ in the C-G schedule is $n_i$.

Claim 2: Each job in segment $X_i$ precedes every job in segment $X_{i-1}, \forall k-1 \leq i \leq 1$.

Because of claim 2, all jobs in $X_i$ must be executed before any job in $X_{i-1}$ can start in any schedule whatsoever. Since $X_i$ consists of $2n_i - 1$ jobs, this will require at least $n_i$ units of time. Thus to execute all jobs, we need at least $\sum_{i=0}^{k-1} n_i$ units of time. But C-G schedule has $C_{\text{max}} = \sum_{i=0}^{k-1} n_i$.

Hence C-G algorithm is optimal.

(Complete proof is given in p. 3-5 of the text book.)
C-G algorithm is not optimal for $P_3 | \text{prec}, P_j = 1 | C_{\text{max}}$

**CLIQUE**

**INSTANCE:** A graph $G = (V,E)$ and a positive integer $J \leq |V|$.  

**QUESTION:** Does $G$ contain a clique of size $J$ or more, that is, a subset $V' \subseteq V$ such that $|V'| \geq J$ and every two vertices in $V'$ are joined by an edge in $E$?

$P | \text{prec}, P_j = 1 | C_{\text{max}}$ is NP-hard (CLIQUE $\alpha P | \text{prec}, P_j = 1 | C_{\text{max}}$)

**3-PARTITION**

**INSTANCE:** Set $A$ of $3m$ elements, a bound $B \in \mathbb{Z}^+$, and a size $s(a) \in \mathbb{Z}^+$ for each $a \in A$ such that $B/4 < s(a) < B/2$ and such that $\sum_{a \in A} s(a) = mB$.

**QUESTION:** Can $A$ be partitioned into $m$ disjoint sets $A_1, A_2, \ldots, A_m$ such that, for $1 \leq i \leq m$, $\sum_{a \in A_i} s(a) = B/3$? (note that each $A_i$ must therefore contain exactly three elements from $A$)

$P \parallel C_{\text{max}}$ is NP-Hard (3-PARTITION $\alpha P \parallel C_{\text{max}}$)