

## Dynamic Programming Method

- In dynamic programming method, the unit commitment table is to be arrived at for the complete load cycle.

### Advantages

- Reductions in the dimensionality of the problem i.e number of combinations to be tried are reduced in number.
- If a strict priority order is imposed, the numbers of combinations for a 4 unit case are:
  - Priority 1 unit
  - Priority 1 unit + Priority 2 unit
  - Priority 1 unit + Priority 2 unit + Priority 3 unit
  - Priority 1 unit + Priority 2 unit + Priority 3 unit + Priority 4 unit

### The priority listing can be used only if:

- No load costs are zero.
- Unit input-output characteristics are linear between 0 output and full load
- Phase shift transformer tap position
- Switched capacitor settings
- Reactive injection for static VAR compensator
- Load shedding
- DC line flow.

### Assumptions:

- Total number of units available, their individual cost characteristics and the load cycle on the station are assumed priori (previously)
- A state consists of an array of units with specified units operating and the rest off- line.
- The start-up cost of a unit is independent of the time it has been off-line (i.e., fixed amount).
- There are no costs for shutting down a unit.
- There is a strict priority order and in each interval a specified minimum amount of capacity must be operating.

### Forward Dynamic programming method Advantages

- Algorithm to run forward in time from the initial hour to the final hour.

- Forward dynamic programming is suitable if the start-up cost of a unit is a function of the time it has been off-line (i.e., fixed amount)
- Previous history of the unit can be computed at each stage.
- Initial conditions are easily specified.

### Algorithm

- One could set up a dynamic-programming algorithm to run backward in time starting from the final hour to be studied, back to the initial hour.
- Conversely, one could set up the algorithm to run forward in time from the initial hour to the final hour.
- The forward approach has distinct advantages in solving generator unit commitment. For example, if the start-up cost of a unit is a function of the time it has been off-line (i.e., its temperature), then a forward dynamic-program approach is more suitable since the previous history of the unit can be computed at each stage.
- There are other practical reasons for going forward.
- The initial conditions are easily specified and the computations can go forward in time as long as required.
- A forward dynamic-programming algorithm is shown by the flowchart
- The recursive algorithm to compute the minimum cost in hour  $K$  with combination  $F_{\text{cost}}(K,I) = \min \{ P_{\text{cost}}(K,I) + S_{\text{cost}}(K-1,L;K,I) \}$

Where

$$F_{\text{cost}}(K,I) = R(K,I)$$

$F_{\text{cost}}(K, I)$  = least total cost to arrive at state  $(K, I)$   $P_{\text{cost}}(K,I)$  = production cost for state  $(K, I)$

$S_{\text{cost}}(K-1, L: K, I)$  = transition cost from state  $(K-1, L)$  to state  $(K, I)$

State  $(K, 1)$  is the  $Z$ th combination in hour  $K$ . For the forward dynamic programming approach, we define a strategy as the transition, or path, from one state at a given hour to a state at the next hour.

Note that two new variables,  $X$  and  $N$ , have been introduced in Figure.  $X$

= number of states to search each period

N = number of strategies, or paths, to save at each step

These variables allow control of the computational effort (see below Figure). For complete enumeration, the maximum number of the value of X or N is  $2^N - 1$

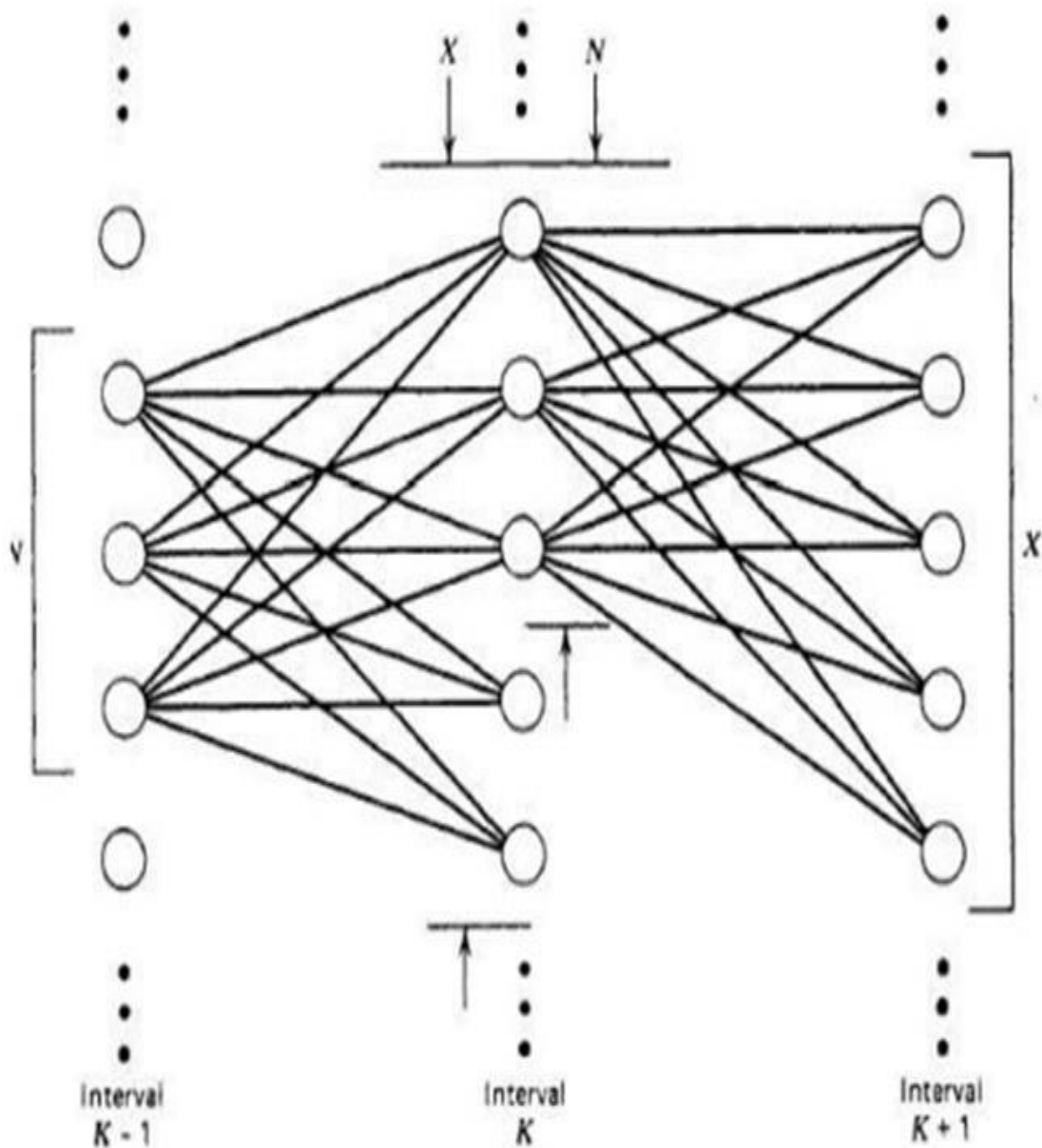


Fig.1 Dynamic programming algorithm

