

1989

The application of the ordered list method and the dynamic programming to the unit commitment

Hoang Uong
Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds



Part of the [Power and Energy Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Uong, Hoang, "The application of the ordered list method and the dynamic programming to the unit commitment" (1989). *Dissertations and Theses*. Paper 3948.

<https://doi.org/10.15760/etd.5832>

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

AN ABSTRACT OF THE THESIS OF Hoang Uong for the
Master of Science in Electrical Engineering May 4th, 1989.

Title : The Application of The Ordered List Method and The
Dynamic Programming to The Unit Commitment.

APPROVED BY THE MEMBERS OF THE THESIS COMMITTEE:

[REDACTED]

Rajinder P. Aggarwal, Chair

[REDACTED]

Jack C. Riley

[REDACTED]

Maria E. Balogh

The electric energy that our society has been using changes daily, weekly, etc. It is important to have some way to know when to turn ON or OFF the power generation units in order to supply the energy demand with the cheapest cost. Proper commitment of the available generating units results in significant cost saving.

The thesis presents a method of committing generating units in a hydro-thermal power system within practical

computer resources such as computer time and data storage.

The presented method applies two different approaches which are the ordered list and the dynamic programming to solve the unit commitment of the hydro and thermal units. The ordered list is used to commit the hydro units because it is simple and uses minimal computer resources. The dynamic programming is employed to commit the thermal units because of its effectiveness in finding an optimal result of a complex problem by breaking it into smaller problems that can be solved independently.

The efficient use of the combination of the above methods makes the unit commitment program run effectively within practical time and the limited main memory of a personal computer. The program using the method described in this thesis ran on a IBM-PCAT 286 12.5 Mhz personal computer with one mega bytes of RAM to solve the unit commitment of seven thermal units and three hydro units for a period of 24 hours. It takes less than one minute of computer time.

THE APPLICATION OF THE ORDERED LIST METHOD
AND THE DYNAMIC PROGRAMMING TO THE
UNIT COMMITMENT

by
HOANG UONG

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
ELECTRICAL ENGINEERING

Portland State University

1989

TO THE OFFICE OF GRADUATE STUDIES:

The members of the Committee approve the thesis of
Hoang Uong presented May 4th, 1989.

[Redacted Signature]

Rajinder P. Aggarwal, Chair

[Redacted Signature]

Jack C. Riley ✓

[Redacted Signature]

Maria E. Balogh

APPROVED:

[Redacted Signature]

Rolf Schaumann, Chair, Department of Electrical Engineering

[Redacted Signature]

Bernard Ross, Vice Provost for Graduate Studies

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank those who have helped and encouraged me to complete this work. My adviser, Dr. R. P. Aggarwal, who encouraged me to complete this thesis which I attempted to give up many times. My family, who always supported me in working on my graduate study even though so much of my family life has been sacrificed. S. C. Clark who is so nice to review my writing. Without their help, this thesis would never been finished.

TABLE OF CONTENTS

	PAGE
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	v
LIST OF FIGURES	vii
CHAPTER	
I INTRODUCTION	1
Unit Commitment	3
Constraints in Unit Commitment	4
II UNIT COMMITMENT SOLUTION METHODS	9
Strictly Priority List Method	11
Dynamic Programming Method	15
III COMPLETE SOLUTION FOR HYDRO-THERMAL SYSTEMS	27
Introduction	27
Solution for The Hydro Units	28
Solution for the Thermal Units	39
Description of The Unit Commitment Procedures	56
IV EXAMPLE	78
V CONCLUSION	84
REFERENCES	90

LIST OF TABLES

TABLE		PAGE
I	The Number of Combinations of N Units in 24 Hours	10
II	Full Load Average Production Cost of Each unit	12
III	The Order of The Units Based on The Full Load average Production Cost	13
IV	All Possible Combinations of The Units . . .	13
V	The Optimal Path from Initial to Final Stage (Bold Letters Indicate Optimal Path) . .	23
VI	Energy Demand Schedule	29
VII	The Dispatching of The Hydro Energy	30
VIII	Unit Data	32
IX	Unit Schedule with The Cheapest Unit Turned on First	33
X	Unit Schedule with The Hydro Unit Turned on When The Energy Demand Is Highest . . .	33
XI	Unit Data	35
XII	Unit Data	38
XIII	Unit Data	42
XIV	The Search Table	43
XV	Off Time Information of Each Unit	48

XVI	On Time Information of Each Unit	49
XVII	The Total Production Cost from Initial Strategy	51
XVIII	Unit Data	52
XIX	Off Time Information of Each Unit	52
XX	Fuel Cost	53
XXI	Data of The Thermal Units	78
XXII	Start-up Cost	79
XXIII	Must on And off Time	79
XXIV	Maintenance Schedule	79
XXV	Data of The Hydro Units	80
XXVI	The Search Table	80
XXVII	The Unit Commitment Schedule	82
XXVIII	The Unit Commitment Schedule If The Maintenance Schedule of Unit 2 Is Optimized	83
XXIX	The Number of Combinations in The Search range	88

LIST OF FIGURES

FIGURE		PAGE
1	Typical Graph	16
2	The Shortest Path is Bold	21
3	The Failure of The Hydro Energy Allocation in Hour 7	36
4	The Order of The Hydro Energy Allocation . .	36
5	The Calculation of a Current Strategy	54

CHAPTER I

INTRODUCTION

Because human activities follow cycles, most systems supplying services to large population will experience cycles. This includes transportation systems, communication systems, as well as electric power systems. The total load on any power system will generally be higher during the day time and early evening when industrial loads are high, lights are on, and be lower during the late evening and early morning when most of the population is asleep. In addition, the use of electric power has weekly cycles. The load being lower over the weekend than weekdays. Why not just simply commit enough units to cover the maximum system load and leave them running? The problem of committing enough units and leave them on line is one of economics. Note that to commit a generating unit is to turn it ON, that is, to bring the unit up to speed, to synchronize it to the system, and to connect it so the unit can deliver power to the network. It is quite expensive to run too many generating units. A great deal of money can be saved by turning units OFF when they are not needed. To serve an economic purpose, we need some ways to schedule when to start and shutdown units, and still satisfy all the

requirements of the whole system. We call this "unit commitment". The general objective of the unit commitment problem is to minimize system operating costs while simultaneously providing sufficient spinning reserve capacity to satisfy a given security level, proper commitment of the available generating units could result in significant cost savings.

Much research has been done in the past in trying to find the best method for unit commitment and many methods have been reported such as linear programming, integer programming, dynamic programming, etc [4, 6, 7]. To find the solution for the unit commitment, a number of difficulties are encountered because of many constraints that are to be satisfied. The more constraints applied to the system, the more computer resource and time is consumed. The dynamic programming approach has proved to be one of the very effective methods for the unit commitment [1, 4]. This method requires significant computer resources in terms of computer memory to store all the data and time to determine the solution.

The purpose of this thesis is to find an efficient method that will result in good unit commitment solution within practical computer time and resource.

Almost all the electric energy we use is transformed from thermal and hydro energy. Hence, this study will present the combination of dynamic programming and ordered

list methods for the commitment of a power system that includes both thermal and hydro units based upon a specified power schedule. The ordered list method is found to be most suitable for the hydro unit commitment and the dynamic programming to be most suitable to the thermal unit commitment. The commitment is such that the total cost is minimum. The total cost includes both the production cost and costs associated with the start-up which consists of hot-start and cold-start cost and shutdown cost of the units. The spinning reserve requirement is also included. In addition, the operation of the individual units must satisfy the specified minimum up-time, minimum shutdown time and maintenance schedule.

UNIT COMMITMENT

The unit commitment determines when to start and shut down units such that the load plus spinning reserve requirement within the specified period is always satisfied. A substantial saving is possible by a proper commitment of the available generating units.

The difficulty of getting a good solution for unit commitment is that there are many constraints to be satisfied each time we want to commit each unit in a system. The problem becomes more complex because a system may contain both thermal and hydro units that have different type of constraints. Besides the constraints applied to each

individual units, another external factor of spinning reserve requirement is applied to the whole system because of system operating requirements.

CONSTRAINTS IN UNIT COMMITMENT

Many constraints can be placed on the unit commitment problem. The list given below is by no means exhaustive but it is the most usual in the electric power systems. Each individual power system or power pool may impose different restrictions on the scheduling of units, depending on the generation makeup, load curve characteristics, etc.

Unit Commitment Constraints

Spinning reserve is the term used to describe the total generation available from all units synchronized on the system minus the present load plus losses being supplied. Spinning reserve must be carried so that the loss of one or more units does not cause too much drop in system frequency. Quite simply, if one unit is lost, there must be large enough reserve on the other units to make up for the loss in a specific time period.

Spinning reserve must be allocated to obey certain rules, usually set by the power pool that specify how reserve is to be distributed to various units. Typical rules specify that reserve must be a given percentage of the forecast of peak demand, or reserve must be capable of

making up the loss of the most heavily loaded unit within a given period, or reserve requirements might be calculated as a function of the probability of not having enough generation to meet the load.

Following are the different types of spinning reserve requirements that may be observed in the commitment of the units:

Fixed Total Spinning Reserve : This is the minimum total effective spinning reserve in the system. It is fixed for the time period of interest. Normally the amount is equal to some percentage of the peak load responsibility of the system.

Minimum Spinning Reserve for Each Area : Each area is assured of a definite additional available capacity within the area to specify the minimum reserve. Such requirement is needed particularly when transmission system capacity is considered.

Maximum Spinning Reserve from any Plant : Such a constraint prevents a concentration of spinning reserve, particularly, in any single large plant and serves to distribute the spinning reserve to all the plants.

Maximum Spinning Reserve from each Unit : This requirement avoids the allocation of a large amount of spinning reserve to a few units. The specified maximum amount for each unit may be reflected by the unit's capacity and its response rate in picking up generation.

Thermal Unit Constraints

Thermal units require a crew to operate them, especially when turned ON and turned OFF. A thermal unit can undergo only gradual temperature changes. This translates into a time period of hours required to bring the unit on line. As a result of such restrictions in the operation of a thermal plant, various constraints arise such as

Minimum Up Time : Once the unit is running, it should not be turned OFF too soon.

Minimum Down Time : Once the unit is decommitted, there is a minimum time before it can be recommitted.

Crew Constraints : If a plant consists of two or more units, they may not be turned ON both at the same time.

Start Up Cost : Because the temperature and pressure of the thermal unit must be moved slowly, a certain amount of energy must be expended to bring the unit on line. This energy does not result in any MW generation from the unit but counts as a major part of start up cost of the unit commitment problem. The start-up cost can vary from a maximum cold-start value to a smaller value if the unit has only been turned OFF recently. There are two methods of treating a thermal unit during its down period. The first allows the unit's boiler to cool down and then heat back up to operating temperature in time for scheduled turn ON. The second is called "banking" and requires that enough energy

be supplied to the boiler to maintain the operating temperature. The costs for the two can be compared so that the best approach may be chosen.

$$\text{Start-up cost when cooling} = C_c(1 - e^{-t/a})F + C_f$$

Where :

C_c : Cold start cost (MBtu).

F : Fuel cost

C_f : Fixed cost (includes crew expense, maintenance expenses)

a : Thermal time constant for the unit

t : Time in hours the unit was cooled

$$\text{Start-up cost when banking} = C_t.t.F + C_f$$

Where :

C_t : cost of maintaining unit at operating temperature. Up to a certain number of hours the cost of banking will be less than the cost of cooling.

Other constraints

a. Must Run : Some units are given a must-run status during certain times of the year for reason of voltage support on the transmission network or for such purposes as supply of steam for uses outside the plant itself.

b. Must Off : Units which have forced or unscheduled outages are unavailable for commitment.

c. Maintenance : The capacity limits of thermal units may change often due to maintenance or scheduled

outages of various equipment in the plant.

CHAPTER II

UNIT COMMITMENT SOLUTION METHODS

In the last decade several different unit commitment programming algorithms have been developed. Many of these programs use techniques such as dynamic, linear, or integer programming to select the most economical schedule of units for a study period [7, 6, 4, 5]. These programs also consider power system requirements such as unit maintenance schedules, minimum up and down time requirements, spinning reserve etc.

Each execution of the unit commitment program requires a large amount of data setup and initialization. Some of the data involved are heuristic and a small change in the heuristic data often causes large change in the results. On the other hand, despite the use of sophisticated programs, it is not always possible to obtain a schedule which is the most economical.

The commitment problem can be very difficult. As a theoretical exercise, let us postulate the following situation.

- a. We must establish a loading pattern for M periods.
- b. We have N units to commit and dispatch.
- c. The M load levels and operating limits on the N

units are such that any unit can supply the individual loads and that any combination of units can also supply the loads.

Next, assume we are going to establish the commitment by enumeration. The total number of combinations we need to try each hour is

$$C_{(N, 1)} + C_{(N, 2)} + \dots + C_{(N, N-1)} + C_{(N, N)} = 2^N - 1$$

Where $C_{(N, j)}$ is the combination of N items taken j at a time.

That is :

$$C_{(N, j)} = N! / [(N - j)! * j!]$$

$$j! = 1 * 2 * 3 * \dots * j$$

For the total of M intervals, the maximum number of possible combinations is $(2^N - 1)^M$, which can become a very large number.

For example, take a 24 hour period and consider systems with 5, and 10 units. The value of $(2^N - 1)^{24}$ becomes the following :

TABLE I
THE NUMBER OF COMBINATIONS OF N
UNITS IN 24 HOURS

N	$(2^N - 1)^{24}$
5	$6.20 * 10^{35}$
10	$1.73 * 10^{72}$

These very large numbers are the upper bounds for the

number of required enumerations. Fortunately, the relationships of the constraints on the units and the load capacity of typical utility systems are such that we do not approach these large numbers. Nevertheless, the real practical barrier in the optimized unit commitment problem is the high dimension of the possible solution space.

The two most discussed techniques for the solution of the unit commitment problems are [1, 7]:

- a. Priority list schemes.
- b. Dynamic programming.

STRICTLY PRIORITY LIST METHOD

The simplest unit commitment solution methods consist of creating a priority list of units. An example below is the best explanation for this method. The priority list can be obtained in a simple manner by noting the full load average production cost of each unit, which is the net heat rate at full load times the fuel cost.

Example : Suppose we have three units given below.

Unit 1 :

Minimum generation $P_{\min} = 150$ MW

Maximum generation $P_{\max} = 600$ MW

Heat rate $H_1 = 510.0 + 7.2P_1 + 0.00142P_1^2$ MBtu/h

Unit 2 :

$P_{\min} = 100$ MW

$P_{\max} = 400$ MW

$$H_2 = 310.0 + 7.85P_2 + 0.00194P_2^2 \text{ MBtu/h}$$

Unit 3 :

$$P_{\min} = 50 \text{ MW}$$

$$P_{\max} = 200 \text{ MW}$$

$$H_3 = 78.0 + 7.97P_3 + 0.00482P_3^2 \text{ MBtu/h}$$

With fuel costs in dollars/MBtu:

$$F_1 = 1.1$$

$$F_2 = 1.0$$

$$F_3 = 1.2$$

First, the full load average production cost will be calculated.

TABLE II

FULL LOAD AVERAGE PRODUCTION COST OF EACH UNIT

Unit	Full Load Average Production Cost (Dollars/MWh)
1	9.79
2	9.4
3	11.188

Second, a strict priority order for these units based on the average production cost would order them as follows:

TABLE III

THE ORDER OF THE UNITS BASED ON THE FULL LOAD
AVERAGE PRODUCTION COST

Unit	Full Load Average		
	Production cost (Dollars/MWh)	P_{\min} (MW)	P_{\max} (MW)
2	9.4	100	400
1	9.79	150	600
3	11.4	50	200

The commitment scheme would (ignoring min up/down time, start up costs, etc.) simply use the following combinations.

TABLE IV

ALL POSSIBLE COMBINATIONS OF THE UNITS

Unit Combination	P_{\min} (MW)	P_{\max} (MW)
2 , 1 , 3	300	1200
2 , 1	250	1000
2	100	400

A shut down rule is simple : When load is above 1000 MW, run all the three units, between 1000 MW and 400 MW, run units 1 and 2; below 400 MW run only unit 2.

The algorithm for the priority list scheme is as

follows:

1. At each hour when a load is dropping, determine whether the dropping of the next unit on the priority list will leave sufficient generation to supply the load plus spinning reserve requirements. If no, continue as is, if yes, go to step 2. When load is increased, determine whether the current unit combination still leaves sufficient generation to supply the load plus spinning reserve requirements. If yes, continue as is, if no, turn on the next unit combination on the list.

2. Determine the number of hours, H , before the unit will be needed again. That is, assuming that the load is dropping and will go back up some hours later.

3. If H is less than the minimum shutdown time for the unit, keep the commitment as is. If no, go to step 4.

4. Calculate two costs. The first is the sum of the hourly production costs from the next H hours with the unit up. Then recalculate the same sum for the unit down and add in the start up cost for either cooling the unit or banking it, whichever is less expensive. If there is sufficient savings from shutting down the unit, it should be shut down, otherwise keep it on.

5. Repeat this entire procedure for the next unit in the priority list. If it drops, go to the next one.

DYNAMIC PROGRAMMING METHOD

Introduction

Many problems are formulated for objects and connections between them. For example, given an air line route map of the United States of America, we might be interested in questions like: "What is the fastest way to get from Portland, Oregon to New York?". Or we might be more interested in money than in time, and look for the cheapest way to get from Portland to New York. In this study, we have the same type of question to save money for the unit commitment. We want to find the cheapest way to commit the power units from the beginning to the end of a period of time. To answer such question we need only information about connections (air line routes, transition from one hour to another) between objects (places, combinations of units).

A graph is a mathematical object which accurately models such situations. Dynamic programming approach utilizes the graph mechanism as means to achieve its goal.

Graphs

A typical graph as shown in figure 1 is a collection of vertices and edges. Vertices are simple objects which can have names and other properties and edge is a connection between two vertices. A graph G consists of two sets called the vertices V and edges E . V is a finite non-empty set of vertices, and E is a finite set of pairs of vertices. Each

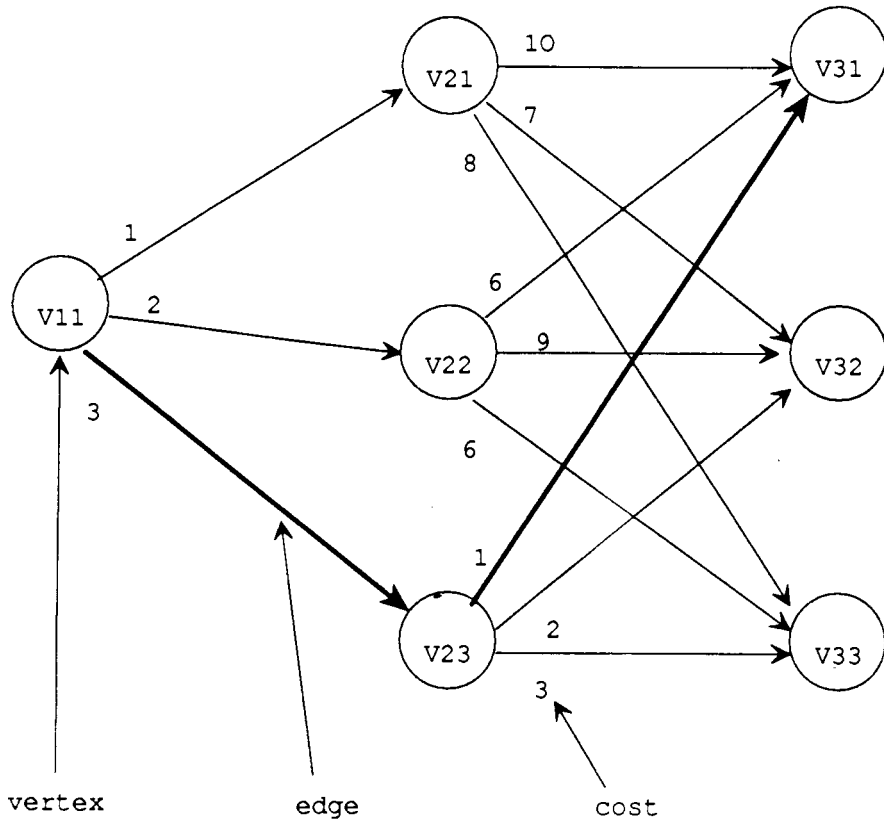


Figure 1. Typical graph.

pair in E is an edge in G . If the pair order i.e. the pair $\langle \text{node } i, \text{node } j \rangle$ is different from the pair order $\langle \text{node } j, \text{node } i \rangle$ then we call the graph directed, otherwise we call it undirected. For many applications there is often a positive real number, called a cost, which is attached to each edge. Such graph is called a network (in our case of unit commitment).

Application of Dynamic Programming to Unit Commitment

The principle of divide and conquer has guided the design of many algorithms to solve a large problem, break it up into smaller problems that can be solved independently. Dynamic programming is an algorithm design method that carries this principle to the extreme. It can be used when the solution to a problem may be viewed as the result of a sequence of decisions [2].

For some problems such as the unit commitment, it is not possible to make step wise decisions (based only on local information) in such a manner that the sequence of decisions made is optimal.

Example :

Given a graph as shown in figure 1. Suppose that we wish to find a shortest path from vertex V_{11} to vertex V_{31} . Let V_{21} , V_{22} and V_{23} be the vertices adjacent from vertex V_{11} . Which of the vertices in V_{21} , V_{22} and V_{23} should be the second vertex on the path in order to have a shortest path from the

vertex V_{11} to vertex V_{31} ? There is no way to decide at this moment and guarantee that future decisions may be made leading to an optimal sequence. What if we wish to find a shortest path from V_{11} to all other vertices in V_{31} , V_{32} and V_{33} ?

There are two principal difficulties with the application of this technique. First, it may not be possible to combine the solutions of two problems to form a solution of a larger one. Second, there may be an unacceptably large number of small problems to solve. However, there is a certain category of problems for which the dynamic programming is quite effective. These problems have a general property that any decision involved in finding the best way to do a small subproblem remains an optimal decision even when that subproblem is included as a piece of some larger problem [2, 3].

One way to solve problems for which it is not possible to make a sequence of step wise decisions leading to an optimal decision sequence is to try out all possible decision sequences. We could list all decision sequences and then pick out the best. Dynamic programming often drastically reduces the amount of enumeration by avoiding the enumeration of some sequences of decisions that can not possibly be optimal. In dynamic programming, an optimal sequence of decisions is arrived at by making explicit appeal to the principle of Optimality.

Principle of Optimality : An optimal sequence of decisions has the property that whatever the initial state and decision are, the remaining decisions must form an optimal decision sequence with regard to the state resulting from the first decision [2]. In dynamic programming, many decision sequences may be generated. However, sequences containing suboptimal subsequences can not be optimal and so will not be generated. An optimal policy must contain only optimal subpolicies [1].

Bellman and Dreyfus explain it as follows : a policy is optimal if, at each stage, whatever the preceding decisions may have been, the decisions still to be taken constitute an optimal policy when the result of the previous decisions is included [1].

With this Optimality principle in mind, the dynamic programming can be applied to solve a wide variety of problems. Here one multi-stage graph application introduced for the unit commitment is of our interest. It is best to explain through an example as follows:

Example :

First, it will be well to introduce some of the notions of dynamic programming. Figure 2 presents the cost of transporting a unit shipment from node V_{01} to node V_{41} . The values on the edges are the costs, or values of shipping the unit from the originating to the terminating node of the edge. The problem is to find the minimum cost route from V_{01}

to V_{41} .

Starting at V_{01} the minimum cost path to V_{41} is $(V_{01}, V_{12}, V_{22}, V_{31}, V_{41})$.

Starting at V_{12} , the minimum cost path to V_{41} is $(V_{12}, V_{22}, V_{31}, V_{41})$.

Starting at V_{22} , the minimum cost path to V_{41} is (V_{22}, V_{31}, V_{41}) .

Starting at V_{31} , the minimum cost path to V_{41} is (V_{31}, V_{41}) .

The same type of statements could be made about the maximum cost path from V_{01} to V_{41} .

The choice of route is made in sequence. There are various stages traversed. The optimum sequence is called optimum policy. Any subsequence is a subpolicy. From this it may be seen that the optimal policy (minimum cost route) contains only optimal subpolicies.

We continue with the same example, let us find the minimum cost path. There are four stages identified as 1, 2, 3 and 4. The numbers in the node circles in Figure 2 represent the minimum costs from initial node V_{11} to those particular nodes.

At each stage, there is a set of nodes to be chosen:

Stage 0 : V_{01}

Stage 1 : V_{11}, V_{12}

Stage 2 : V_{21}, V_{22}, V_{23}

Stage 3 : V_{31}, V_{32}

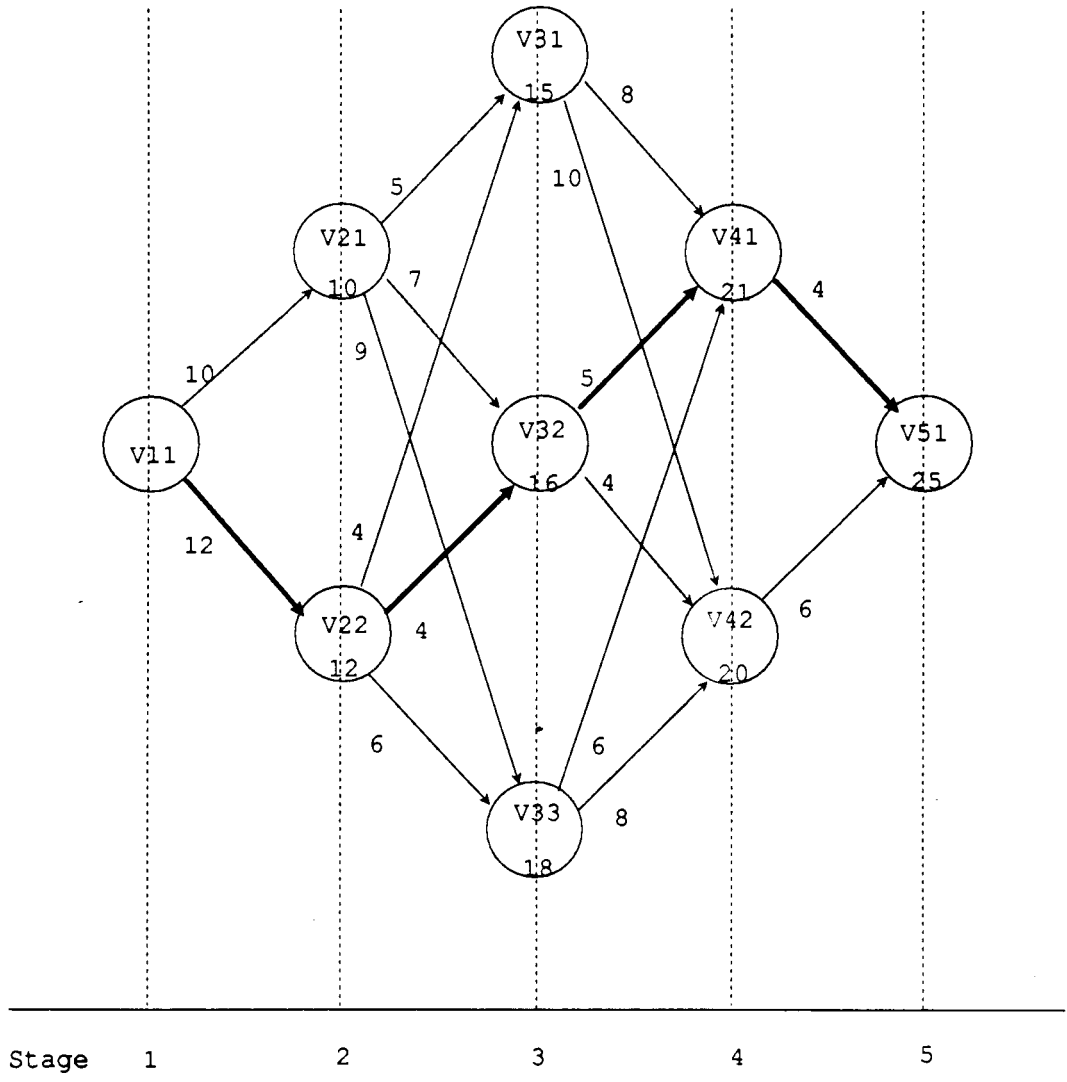


Figure 2. The Shortest Path is Bold.

Stage 4 : V_{41}

$MTC_s(V_i)$: minimum total cost at node V_i in stage s .

$C(V_i, V_j)$: cost from node V_i to V_j .

STAGE 0 :

Minimum total cost at stage 0 :

$$MTC_0(V_{01}) = 0$$

STAGE 1 :

$$MTC_1(V_{11}) = C(V_{01}, V_{11}) = 10$$

$$MTC_1(V_{12}) = C(V_{01}, V_{12}) = 12$$

STAGE 2 :

$$\begin{aligned} MTC_2(V_{21}) &= \min [MTC_1(V_{11}) + C(V_{11}, V_{21}), MTC_1(V_{12}) + C(V_{12}, \\ &V_{21})] \\ &= \min [10 + 5, 12 + 4] \\ &= 15 \end{aligned}$$

Minimum cost path to node $V_{21} = (V_{01}, V_{11}, V_{21})$.

$$\begin{aligned} MTC_2(V_{22}) &= \min [MTC_1(V_{11}) + C(V_{11}, V_{22}), MTC_1(V_{12}) + C(V_{12}, \\ &V_{22})] \\ &= \min [10 + 7, 12 + 4] \\ &= 16 \end{aligned}$$

Minimum cost path to node $V_{22} = (V_{01}, V_{12}, V_{22})$.

$$\begin{aligned} MTC_2(V_{23}) &= \min [MTC_1(V_{11}) + C(V_{11}, V_{23}), MTC_1(V_{12}) + C(V_{12}, \\ &V_{23})] \\ &= \min [10 + 9, 12 + 6] \\ &= 18 \end{aligned}$$

Minimum cost path to node $V_{23} = (V_{01}, V_{12}, V_{23})$.

STAGE 3 :

Repeat the same process as in stage 2, we have :

Minimum cost path to node $V_{31} = (V_{01}, V_{12}, V_{22}, V_{31})$.

Minimum cost path to node $V_{32} = (V_{01}, V_{12}, V_{22}, V_{32})$.

STAGE 4 :

Minimum cost path to node $V_{41} = (V_{01}, V_{12}, V_{22}, V_{31}, V_{41})$.

Tracing back, the path of minimum cost is found as follows :

TABLE V

THE OPTIMAL PATH FROM INITIAL TO FINAL STAGE
(BOLD LETTERS INDICATE OPTIMAL PATH)

Stage	Node	Minimum Cost
0	V_{01}	0
1	V_{11}, V_{12}	10, 12
2	V_{21}, V_{22}, V_{23}	15, 16, 18
3	V_{31}, V_{32}	21, 20
4	V_{41}	25

It would be possible to carry out this procedure in the opposite direction just as easily.

Unit commitment, using dynamic programming methods, tries to find an optimal cost to commit the units in a given period of time (say in unit hour) with a given initial state of all the units. Each hour, a number of most economical and feasible combinations of units (the combinations of units ON or OFF that cost the least and satisfy our requirements) are

selected and a calculation is made to obtain the total cost from initial state up to the moment. Then, a number of unit combinations that result in the most economical approach will be kept. These decision sequences are generated for each hour until the final hour. We select the cheapest solution at the final hour and trace back to get the most economical path.

a. Forward Dynamic Programming Approach

The procedure starts from initial state and goes forward in time to compute the optimal schedule. There are M intervals in the period to be considered. The dynamic programming equations for the computation of the minimum total fuel cost during a time period, K , are given here.

$$F_{\text{cost}}(K, I) = \text{Min} [P_{\text{cost}}(K, I) + S_{\text{cost}}(K - 1, L : K, I) + F_{\text{cost}}(K - 1, L)]$$

{L}

Where

{L} = Set of feasible states in interval $K - 1$.

$F_{\text{cost}}(K, I)$ = 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) where state (K, I) is the I th combination in hour K .

The production cost $P_{\text{cost}}(K, I)$ is obtained by economically dispatching the units on line in state I .

A path is a schedule starting from a state in an

interval K to the final interval M. An optimal path is one for which the total fuel cost is minimum. Given the paths starting from all the individual states in (K + 1)th interval, the optimal path originating from any state in the Kth interval can be found. This is the main advantage of the dynamic programming method. The procedure for determining the optimal schedule and the minimum total fuel cost is shown below.

Procedure : Unit commitment using forward dynamic programming.

begin

 K = 1

 for all states I in interval K

 begin

 for all states L in interval K - 1

$$F_{\text{cost}}(K, I) = \text{MIN} [P_{\text{cost}}(K, I) + S_{\text{cost}}(K-1, L : K, I)]$$

 end

 do begin

 K = K + 1

 {L} = N feasible states in interval K - 1

 for all X = states I in interval K

 begin

 for all states L in interval K - 1

$$F_{\text{cost}}(K, I) = \text{MIN} [P_{\text{cost}}(K, I) + S_{\text{cost}}(K-1, L : K, I) + F_{\text{cost}}(K - 1, L)]$$

```
end
    save N lowest cost strategies
until ( K = M )
    trace optimal schedule
end
```

b. Backward Dynamic Programming Approach.

This dynamic programming approach is the same as the forward dynamic programming except that it uses a backward (in time) approach in which the solution starts at the last interval and proceeds back to the initial point. If the start up cost of a unit is a function of the time it has been OFF line or the unit must satisfy the minimum up and down-time so we need to know the previous state of this unit in order to decide whether the current state is possible then the backward dynamic programming approach is not applicable to unit commitment. Furthermore, the initial conditions are easily specified in forward dynamic programming. Because of the above reasons, the backward dynamic programming will not be mentioned in this thesis.

CHAPTER III

COMPLETE SOLUTION FOR HYDRO-THERMAL SYSTEMS

INTRODUCTION

In the hydro-thermal power system, due to the inexpensive production cost, the use of hydro energy is always preferred to thermal energy, but the hydro electric energy is limited. The optimal scheduling of hydro-thermal power systems has to deal with the problem of how to take best advantage of the available hydro energy while the cost of thermal energy is kept minimal in the system. This thesis presents a unit commitment method that leads to an optimal result in term of cost for this hydro-thermal power system.

The fuel cost of the thermal units is a quadratical function of the energy it produces :

$$C = a_0 + a_1 * P + a_2 * P^2$$

where :

C : the production cost.

a_0 , a_1 , a_2 : predetermined constants.

P : the thermal power.

The cost of the hydro energy is taken as zero.

SOLUTION FOR THE HYDRO UNITS

The priority list approach does not lead to an optimum result for a thermal power system because in practice many other constraints such as minimum up-time, down-time, start-up cost, ... need to be taken into account for the thermal units. However, it is applicable to the hydro units because the hydro units do not have such constraints and the fuel cost (cost of water) is taken as zero. The only constraint on hydro units is their energy capacity because of the limited water storage. Given the limited hydro energy and the number of hydro units, there is no need for a priority order in which the hydro units should be turned ON in order to determine their production cost. However, from computer programming stand point, we still need to arrange all the hydro units in some order so that we can turn them ON or OFF in sequence. Thus, we may arrange the given hydro units in an arbitrary order into a list called an ordered list.

Example:

Assume we have a system consisting of three hydro units (there is no thermal unit here). Remember that the fuel and the start-up costs are zero.

Unit 1 :

Minimum power $P_{\min} = 1$ MW.

Maximum power $P_{\max} = 300$ MW.

Total energy capacity E_{cap} : 300 MWh.

Unit 2 :

$$P_{\text{min}} = 1 \text{ MW.}$$

$$P_{\text{max}} = 150 \text{ MW.}$$

$$E_{\text{cap}} : 200 \text{ MWh.}$$

Unit 3 :

$$P_{\text{min}} = 1 \text{ MW.}$$

$$P_{\text{max}} = 100 \text{ MW.}$$

$$E_{\text{cap}} : 100 \text{ MWh.}$$

The energy demand schedule is shown in Table VI.

TABLE VI
ENERGY DEMAND SCHEDULE

Hour :	Energy demand (MWh)
1	200
2	150
3	250

Checking the state transition of a unit from one hour to the next is not necessary because there is no restriction on minimum up and down time.

We only need to know that at any given time the total energy of each unit is reduced by the same amount as the energy supplied by the unit.

TABLE VII
THE DISPATCHING OF THE HYDRO ENERGY

Hour :	Energy Demand	Energy Supplied			Available Energy			Cost
	(MW)	(MWh)			(MWh)			(Dollars)
		Unit			Unit			
		2	1	0	2	1	0	
1	200	0	0	200	100	200	100	0
2	150	0	50	100	100	150	0	0
3	250	100	150	0	0	0	0	0

The Table VII shows the energy demand schedule, the energy supplied by each unit and the available energy of each unit in each hour. In Table VII, the ordered list of three hydro units is from right to left : unit 0, unit 1 and unit 2.

For computer programming, we use a binary number to represent a combination of the hydro units with each bit corresponding to a unit.

Whenever we want to turn ON a number of available units, the unit corresponding to the least significant bit is always to be turned ON first and the most significant bit to be turned ON last until we have enough units to supply the energy demand. In any particular hour as we know the energy demand, we apply this rule to select the unit corresponding to the least significant bit to find out

whether it has enough energy capacity to provide the energy. If yes, we turn this unit ON and supply the maximum energy, or the maximum energy capacity or the energy demand whichever is lower. If no, we leave the unit OFF. Then we consider the next unit until the required energy is satisfied. But when a unit supplies some energy, the same amount of energy must be deducted from the current energy capacity of the unit. We then know how much energy is still available in the subsequent hours.

By selecting all the hydro units this way, the worst case is that we have to repeat the above process N times which is the number of hydro units to be committed for only one unit combination. We do not need to go through 2^N combinations (as in dynamic programming) in order to come up with the same result.

The use of the ordered list method in committing hydro units is simple and straight forward and it requires minimal computer resources in term of computational time and data storage.

Given a combination of thermal units, we may want to turn on the cheapest unit first and the most expensive one last. This strategy would work for the thermal system but it is not applicable for the hydro-thermal system in term of production cost. For our discussion, let's look at the simple example given below.

Example:

Assume we have one hydro and two thermal units.

TABLE VIII

UNIT DATA

Unit 0	Unit 1	Unit 2
Hydro	Thermal	Thermal
P_{\min} 1 MW	P_{\min} 50 MW	P_{\min} 25 MW
P_{\max} 100 MW	P_{\max} 150 MW	P_{\max} 200 MW
E_{cap} 100 MW		

All the costs are in dollars.

Fuel cost of unit 0 = 0.00

Fuel cost of unit 1 = $510.0 + 7.2P_1 + 0.00142P_1^2$

Full load average cost = 9.78

Fuel cost of unit 2 = $310.0 + 7.85P_2 + 0.00194P_2^2$

Full load average cost = 10.813

Energy demand for the first, second and the third hour are 100, 300, 200 MWh.

The Table IX shows the best solution for the unit commitment based on the principle of turning ON the cheapest unit first and the most expensive unit last for a given unit combination. Note that in the second and the third hour the hydro unit is no more usable because all of its energy was used up. The total production cost after three hours is \$ 5159.40.

TABLE IX

UNIT SCHEDULE WITH THE CHEAPEST UNIT TURNED ON FIRST

Hour	Unit			Energy Supplied by Unit (MWh)			Total Cost (Dollars)
	2	1	0	2	1	0	
	1	0	0	1	0	0	
2	1	1	NA	100	200	NA	\$ 3201.8
3	0	1	NA	0	200	NA	\$ 1957.6
							\$ 5159.4

TABLE X

UNIT SCHEDULE WITH THE HYDRO UNIT TURNED ON
WHEN THE ENERGY DEMAND IS HIGHEST

Hour	Unit			Energy Supplied by Unit (MWh)			Total Cost (Dollars)
	2	1	0	2	1	0	
	1	0	1	0	0	100	
2	0	1	1	0	200	100	\$ 1957.6
3	0	1	0	0	200	0	\$ 1957.6
							\$ 5029.6

Another way of unit commitment for the same example shown in Table X. The total cost after three hours is \$ 5029.60. This unit commitment solution is more economical

than the previous one. Remember the fuel cost of the thermal unit increases quadratically. Given a fixed amount of hydro energy, we want to reduce the fuel cost per energy unit of the thermal unit as much as possible. Thus, we want to consume all the available hydro energy but at the same time we should use it whenever the energy demand is highest in order to reduce the amount of thermal energy during that hour and this strategy in turn will reduce the fuel cost per thermal energy unit.

It is best to demonstrate how the above proposal can be implemented. We have an energy demand schedule and a fixed amount of hydro energy. First, we find the highest energy demand in the schedule. Then we reduce this energy demand by a small amount and call the result as the initial thermal energy cutoff point. Next, we walk through the energy demand in each hour. If the energy demand for any particular hour is more than the thermal energy cut off point, then we reduce the energy demand by the difference between the energy demand and the thermal energy cut off point. At the same time the hydro energy is allocated by the same amount for the hour in consideration. The process is repeated and the new thermal energy cut off point will move down as more hydro energy is allocated each time. Finally, the process will be ended when the entire hydro energy is consumed. The thermal energy cut off points in each hour may be different because of the constraint of the maximum energy supplied by

the hydro units and the minimum energy supplied by the thermal units. The above process is illustrated by the example below.

Example:

Consider a hydro-thermal power system consists of 3 thermal and 2 hydro units.

The load demand schedule is 450, 530, 600, 1000, 300, 290, 700 MWh.

TABLE XI
UNIT DATA

Unit	Type	Pmax (MW)	Pmin (MW)	Capacity (MWh)
0	Hydro	100	1	300
1	Thermal	400	60	
2	Thermal	300	25	
3	Thermal	600	20	
4	Hydro	100	1	300

Figure 3 shows how the hydro energy is scheduled. Observe : the thermal energy cut off point for hour 4 is different than for hour 2, 3 and 7.

After finding the thermal energy cut off points, we know exactly the amount of the hydro and thermal energy for each hour. We can commit the hydro units independently from the thermal units because we want to apply the ordered list

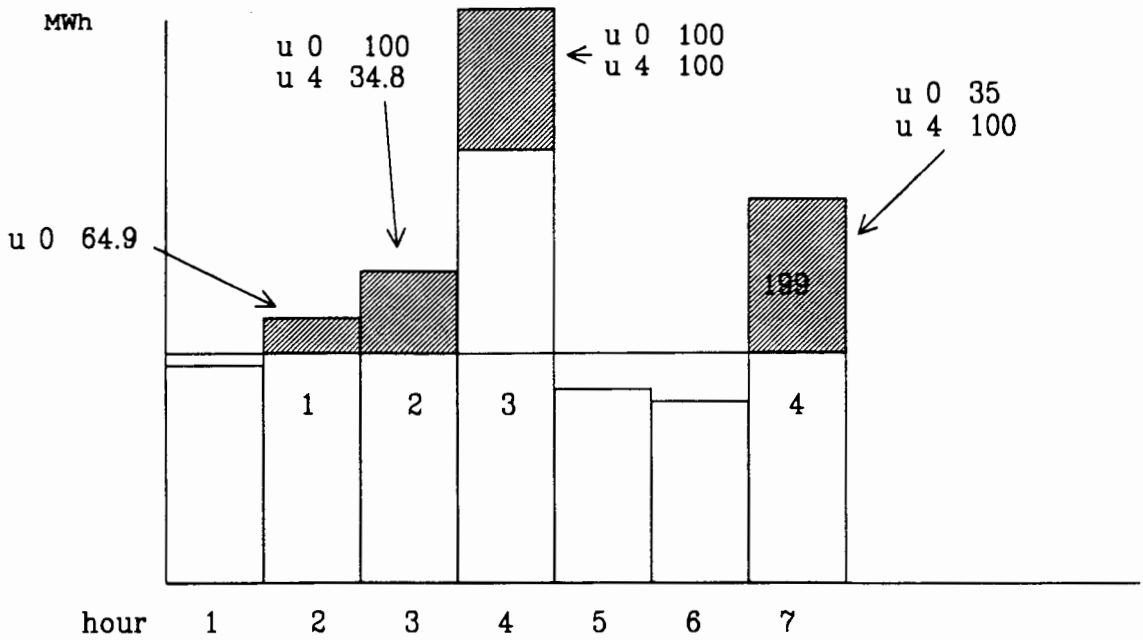


Figure 3. The Failure of The Hydro Energy Allocation in Hour 7.

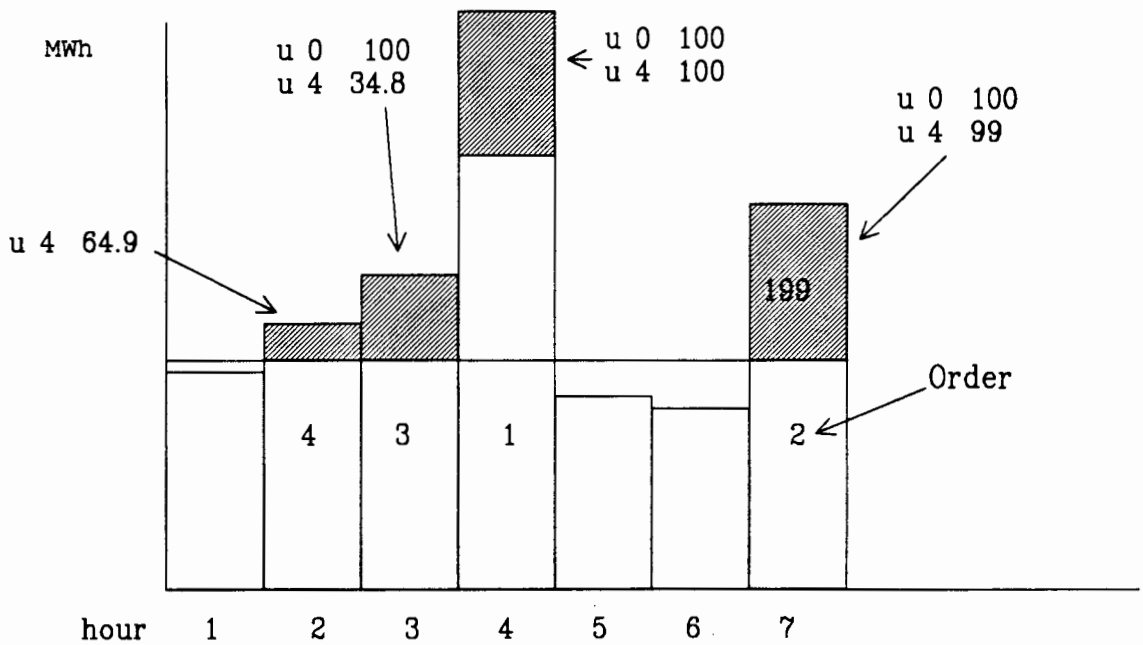


Figure 4. The Order of The Hydro Energy Allocation.

scheme to commit the hydro units and the dynamic programming to commit the thermal units. If we use the ordered list scheme to commit a part of the system, this significantly reduces computational time and computer memory resource. The saving of these computer resources will depend on the number of units in the system committed by the ordered list scheme.

In order to use the ordered list to commit the hydro units effectively, we have to commit the hydro units according to the peak hydro energy demand in the schedule first and the least demand last because of the restriction of the maximum hydro energy provided in each hour and the energy capacity of each unit. If we commit the hydro units in this way, we can dispatch all the hydro energy throughout the schedule. If we do not proceed in the above manner, then a problem may arise when the hydro energy demand is too high at the end. The example below illustrates the method to dispatch the hydro energy.

Example :

A system has two hydro and three thermal units with the unit data shown in Table XII below.

Energy demand schedule : 450, 530, 600, 1200, 300, 290, 700 (MW) .

TABLE XII
UNIT DATA

Unit	Type	P_{\max} (MW)	P_{\min} (MW)	E_{cap} (MWh)
0	Hydro	100	1	300
1	Thermal	400	60	
2	Thermal	300	25	
3	Thermal	600	20	
4	Hydro	100	1	300

Figure 3 shows the failure to commit the hydro units at the final hour if we commit the hydro units according to the first hour first and the last hour last. This problem arises because the hydro energy is not distributed moderately through out the schedule. In turn, the capacity of some units runs out early and when the hydro energy demand peaks late in the schedule, the rest of the hydro units can not supply enough hydro energy because of their maximum energy constraint.

Figure 4 shows the result of the hydro energy dispatching that commits the units according to the largest hydro energy amount first and the smallest last. This scheme evenly distributes the hydro energy through out the schedule.

SOLUTION FOR THE THERMAL UNITS

To commit a thermal unit, the unit must satisfy minimum up and down-time; the start-up and production cost can vary substantially among different units. For the thermal units, the dynamic programming is more suitable because it can provide many possible combinations of the units to be selected in order to satisfy all the imposed constraints. This method also provides us a capability to select a number of optimal decisions in each hour.

In dynamic programming, we need :

a. To define a strategy how to turn ON the units of a given combination.

b. To have a search table that contains all the feasible combinations of all the units.

c. The following steps to accept a combination the thermal units :

1. Select a combination that can supply the energy demand.
2. Check for must ON condition of all the units.
3. Check for must OFF condition of all the units.
4. Check for maintenance schedule of all the units.
5. Check for minimum OFF time condition if the unit is ON and previously OFF.

6. Check for minimum ON time if the unit is OFF and previously ON.
 7. if any of the conditions in 2, 3, 4 ,5 and 6 is not satisfied then go to step 1.
 8. Calculate the total production cost and start-up cost of all the units.
- d. Select the combinations that produce the most economical schedule.
- e. Repeat step c and d until the end of the schedule is reached.

The Order to Turn On the Units

We use a binary number to represent for all the units such that the least significant bit designates the unit that has the cheapest full load average cost and the most significant bit designate the most expensive unit. When we want to commit the units of a unit combination, we turn ON the least significant bit first and the most significant bit last.

Example:

Three thermal units:

Unit 0 has Full load average production cost of \$ 10.50.

Unit 1 has Full load average production cost of \$ 8.25.

Unit 2 has Full load average production cost of \$ 9.50.

The least significant binary bit represents Unit 1, the next bit unit 2 and the most significant bit unit 0. A

combination of unit 0 OFF, unit 2 ON and unit 1 ON has a binary representation of 011. The order to turn ON the units is from the least significant bit to the most significant bit (i.e. from right to left).

The Search Table

We need a table to contain all the feasible combinations of units. We have to search through the table in each hour to find the combinations that can supply the energy demand and satisfy all the constraints. We will calculate and select the most economical solutions out of these combinations. The table is called the search table. To minimize the time to search for a feasible combination, the table should contain only the combinations that can supply enough or more than the minimum energy demand in the schedule.

In computer programming, we use a binary number as discussed in the above section. This binary number is initialized to 0 and then is incremented until it reaches 2^n where n is the number of units. Each increment gives a new combination of all the units and we only keep the feasible combinations in an array (the search table). A combination is saved in the table if the ON units can supply the energy that is more than the minimum energy demand of the schedule. Then this table is ascendingly sorted according to the maximum fuel costs of the combinations and then according to

the maximum power the combinations can supply. With this arrangement, when we search for a combination that can produce a given amount of energy, the first feasible combination that we encounter is the cheapest combination in term of fuel cost. All the subsequent combinations always are guaranteed to satisfy the energy demand. However, the present combination may not have the cheapest production cost because another factor contributes to the total production cost of a combination that is still unknown is the start-up cost, etc. Each entry of the search table contains the binary combination of the units and the total maximum power of the units.

Example:

We have three thermal units, the Table XIII shows the data of the units.

TABLE XIII

UNIT DATA

Unit	Power Supply		a0	a1	a2
	Max (MW)	Min (MW)			
0	300	25	150	20.8	0.002
1	300	60	120	18.0	0.005
2	600	20	100	23.8	0.003

The energy schedule : 450, 530, 600, 900, 300, 290, 700

(MWh) .

The search table is shown in Table XIV. Note, each entry of the search table contains a binary combination of the units and the maximum power supplied by the combination. The costs of the maximum power of the combinations are shown so that the reader can see the order of the table is sorted according to the fuel cost and the power supply of the combinations of the units.

TABLE XIV
THE SEARCH TABLE

Unit Combination	Maximum Power (MW)	Cost (Dollars)
0 0 1	300.00	5970.00
0 1 0	300.00	6570.00
0 1 1	600.00	12540.00
1 0 0	600.00	15460.00
1 0 1	900.00	21430.00
1 1 0	900.00	22030.00
1 1 1	1200.00	28000.00

Select a Combination of Units That Satisfies The Energy Demand

To find a combination of units that can supply the energy demand in a particular hour, we start searching from top to bottom of the search table for an entry that can provide the maximum power greater than the energy demand.

Example:

The energy demand is 600 MWh. We start from top to bottom of the Table XIV and find that the combination 011 is suitable.

Check The Units for Must On Condition

In each hour, we need to check if all the units of any given unit combination for meeting their must ON condition. A binary number, used as a mask, is created where each bit is ON if the corresponding unit must be ON. The selected combination is ANDED with this mask to check for the must ON condition. If the combination does not satisfy the must ON condition then it will be rejected.

Example:

We have three units 0, 1 and 2 arranged from least significant bit to most significant bit. Unit 0 must be ON. The must ON mask is 001.

If we have combinations 110 and 011 :

110 AND 001 = 000

011 AND 001 = 001

The second combination is selected because its result is equal to the must ON mask which is 001.

Check The Units for Must Off Condition

In each hour, we also check for the must be OFF condition of all the units of any given unit combination. A mask is created with each bit is ON if the corresponding

unit must be OFF. The mask and the combination are ANDED to check for the must be OFF condition. If the combination does not satisfy the must OFF condition then it will be rejected.

Example:

We have three units 0, 1 and 2 arranged from least significant bit to most significant bit. Unit 0 must be OFF. The must OFF mask is 001.

If we have combinations 110 and 011 :

110 AND 001 = 000

011 AND 001 = 001

The first combination is selected because its result is equal to zero.

Check The Units for Maintenance Schedule

A mask is also created to check for the maintenance schedule condition. The combination is rejected if it does not satisfy the condition.

Example:

We have three units 0, 1 and 2 arranged from least significant bit to most significant bit. Unit 0 must be MAINTAINED. The MAINTENANCE mask is 001.

If we have combinations 110 and 011 :

110 AND 001 = 000

011 AND 001 = 001

The first combination is selected because its result is equal to zero.

Check for Minimum Off time

During the dynamic programming in progress, each strategy that is saved contains all the information such as the combination of units, the total cost of the energy demand and the total cost from initial strategy up to the current strategy. The ON and OFF times of each unit are also recorded in each strategy. Given a previous and current combinations, each bit corresponding to the unit in question is checked if it is OFF in the previous and ON in the current combination. If the condition above is true then the minimum OFF time of the unit is examined to see if the corresponding unit satisfies its minimum OFF time.

The minimum OFF time of all the units under consideration should be checked before any fuel cost calculation of any unit of the combination is made. The performance of the program significantly increases with this checking placed in the correct order.

One best way is to create a mask for each strategy with each bit corresponding to the unit that does not satisfy the minimum OFF time is set to ON. This mask is used to checked the minimum OFF time condition of each unit combination of the subsequent hour from the current strategy before any fuel cost calculation occurs.

Example:

A combination of 10111 with the unit 4 corresponding to the most significant bit does not satisfy the minimum OFF

time and all the others (unit 0 to unit 3 correspond to bit 0 to bit 3) do. If the procedure applies the formula :

$$C = a_0 + a_1.P + a_2.P^2$$

to calculate the fuel cost of unit 0, unit 1 and unit 2 and finally the combination is rejected because the unit 4 is found not to satisfy the minimum OFF time. A lot of computer time is wasted before a combination is rejected.

Example:

Use the data in the above example. We have a personal computer (80286 microprocessor) with a math coprocessor installed. A floating point multiplication operation costs 225 microprocessor clocks and an addition operation costs 24 clocks [14, 15].

The combination 10111 is rejected after applying the formula to calculate the fuel costs of the first three ON units. Each time the formula is calculated, it takes $24 + 225 + 24 + 225 * 2$ microprocessor clock time. The total of 2169 microprocessor clocks are used to reject the combination. But if the minimum OFF time mask is created and the use of an AND operation to reject the same combination, only 4 microprocessor clocks are consumed. With a system of a larger number of thermal units, the AND operation significantly improves the performance of the program.

Example:

We have three units 0, 1 and 2 arranged from least significant bit to most significant bit. Table XV shows the

strategies of the previous hour and the information related to the OFF time of each unit.

TABLE XV
OFF TIME INFORMATION OF EACH UNIT

Binary Number Corresponds To the Units			Off Time of Unit (Hours)		
2	1	0	2	1	0
0	1	0	1	0	2
0	1	1	3	0	0
1	1	1	0	0	0

If the combination of the previous hour is 010 and then unit 0 and 2 are checked for minimum OFF time condition. If the required minimum OFF time of unit 2 is 2 hours and unit 0 is 2 hours then the minimum OFF time mask 100 is created because unit 2 does not satisfy the minimum OFF time condition.

If we have combinations 111 and 011 then :

$$111 \text{ AND } 100 = 100$$

$$011 \text{ AND } 100 = 000$$

The combination 111 is rejected because the result of the AND operation is not equal to 0. The combination 011 is accepted.

Check for Minimum On Time

A minimum ON time mask is also created for the same reason as above.

Example:

We have three units 0, 1 and 2 arranged from least significant bit to most significant bit. Table XVI shows the strategies of the previous hour and the information related to the ON time of each unit.

If the combination of the previous hour is 011 then unit 1 and unit 0 are checked for minimum ON time condition. If the required minimum ON time of unit 1 is 3 hours and unit 0 is 2 hours then a minimum ON time mask of 010 is created.

TABLE XVI
ON TIME INFORMATION OF EACH UNIT

Binary Number Corresponds to The Units			On Time of Unit (Hours)		
2	1	0	2	1	0
0	1	0	0	2	0
0	1	1	0	1	3
1	1	1	1	1	2

If we have combinations 101 and 011 :

010 AND 011 = 010

010 AND 101 = 0

The combination 011 is accepted because the result is equal to the mask 010. The combination 101 is rejected.

Calculating The Fuel Cost

When the combination satisfies all the conditions stated in the above sections then the fuel cost is calculated. The fuel cost of each unit is calculated right after the generated energy of the unit is known. Starting from the least significant bit to the most significant bit of the combination, if the unit is ON then its maximum energy or the energy demand whichever is lower is used to calculate the fuel cost. Finally, the total fuel cost of the combination is computed.

Example:

Table XVII shows the total production cost of the path of from initial strategy to the saved strategies of the previous hour.

Using the combination 011 of the current hour, if the fuel cost of unit 0 is \$ 300.00 and unit 1 is \$ 400.00 then the fuel cost of the combination is :

$$\text{\$ } 300.00 + \text{\$ } 400.00 = \text{\$ } 700.00$$

If the previous strategy is 010 then the total production cost sofar is (excluding start-up cost):

$$\text{\$ } 1000.00 + \text{\$ } 700.00 = \text{\$ } 1700.00$$

TABLE XVII

THE TOTAL PRODUCTION COST FROM INITIAL STRATEGY

Binary Number Corresponds			Total Production Cost Sofar
To Units			(Dollars)
2	1	0	
0	1	0	\$ 1000.00
0	1	1	\$ 1100.00
1	1	1	\$ 1200.00

Calculating The Start-up Cost

If the bit corresponding to the unit is OFF in the previous and ON in the current combination, we use the information related to the OFF time of the unit saved in the previous combination to check if the OFF time is greater than the specified cold-start time then we need to add cold-start cost into the total cost of the current combination otherwise we use the hot-start cost.

Example:

We have three units 0, 1 and 2 arranged from least significant bit to most significant bit. Table XVIII shows the start-up cost of each unit, Table XIX the information of the previous strategies and Table XX shows the fuel cost of each combination in the current hour.

TABLE XVIII

UNIT DATA

Unit	Cold-start Time (Hours)	Cold-start Cost (Dollars)	Hot-start Cost (Dollars)
0	2	150	70
1	3	100	60
2	2	120	80

If the current combination is 011 and the previous one is 010 then the OFF time of unit 0 is examined. In this case, the cold-start cost is \$ 150.00. But if the current combination is 110 then the hot-start of unit 2 is \$ 80.00.

TABLE XIX

OFF TIME INFORMATION OF EACH UNIT

Binary Number Corresponds to the Units			Off Time of Unit (Hours)			Total Production Cost Sofar (Dollars)
2	1	0	2	1	0	
0	1	0	1	0	3	1000.00
0	1	1	3	0	0	1200.00
1	1	1	0	0	0	1100.00

TABLE XX
FUEL COST

Binary Number Corresponds			Fuel Cost
To Units			(Dollars)
2	1	0	
0	1	1	900.00
1	1	0	1100.00

Calculate The Total Production Cost

The total production cost from the initial strategy up to the current strategy is calculated by adding up the start-up cost, fuel cost of the current strategy to the total production cost of the previous strategy.

Example:

Use the data of the example in the previous section. If the current combination is 011 and the previous one is 010 then the OFF time of unit 0 is examined. In this case, total production cost of the previous strategy, the fuel cost and the cold-start cost are added up to give the total production cost of the current strategy :

$$\$ 1000.00 + \$ 900.00 + \$ 150 = \$ 2050.00$$

But if the current combination is 110 then the hot-start of unit 2 is used:

$$\$ 1000.00 + \$ 1100.00 + \$ 80 = \$ 2180.00$$

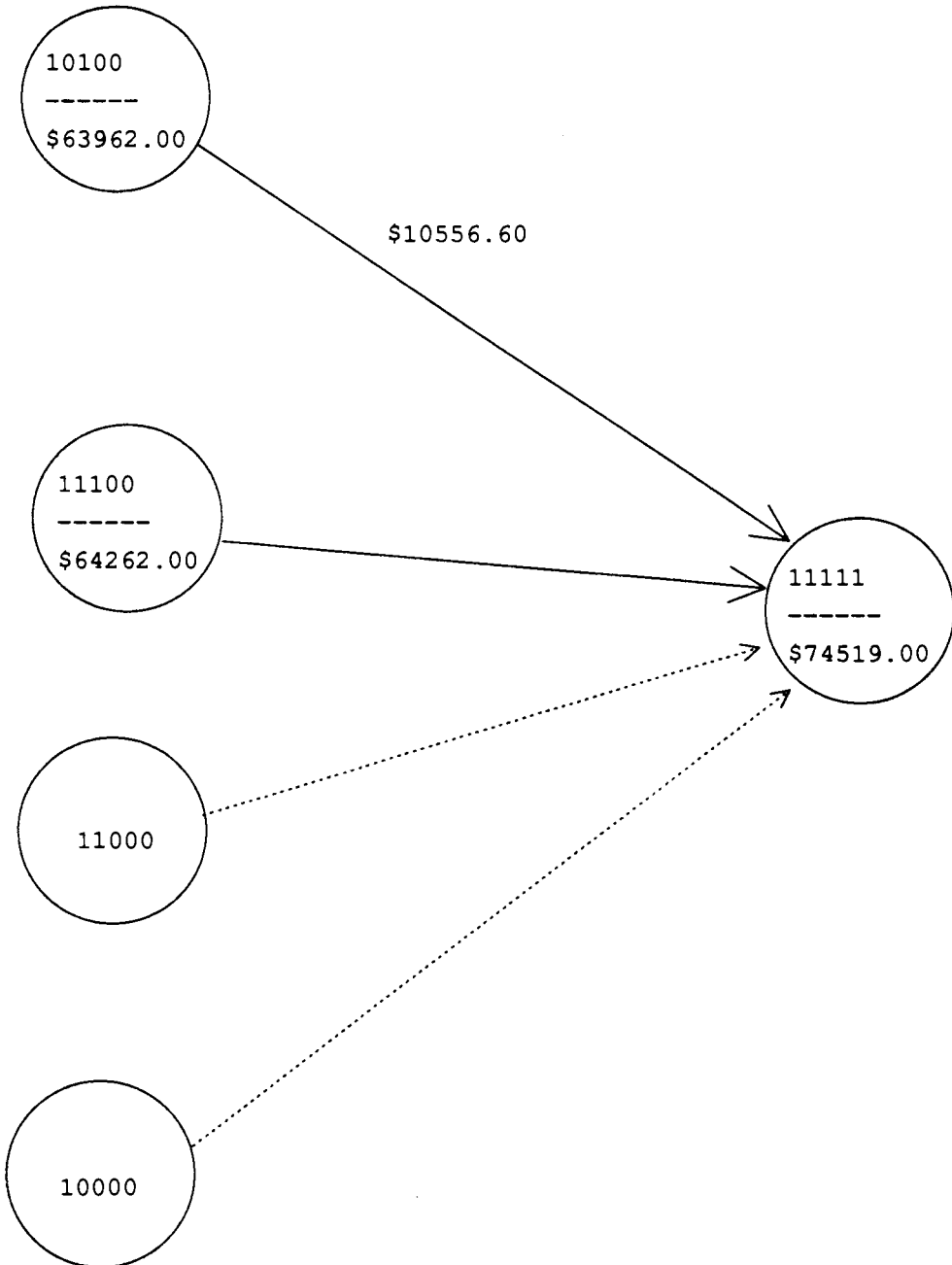


Figure 5. The Calculation of a Current Strategy.

Saving The Strategies in Each Hour

All the above steps are repeated for a predetermined number of combinations called temporary strategies. We only save some strategies, called strategies, out of the number of temporary strategies based on the most economical total production cost from initial up to the current states.

The following three steps are done in order to save the strategies in each hour.

a. Calculate the total production cost from initial strategy up to the combination of units of the current hour. This process is repeated for each of the strategies of the previous hour.

Example:

A number of four strategies must be saved in each hour. Figure 5 shows that in hour 7, a feasible combination of units is selected and four calculations are made for the transitions from each of the strategies of the previous hour which is hour 6.

b. Select the cheapest solution out of the results of the above steps.

Example:

Figure 5 shows that only one cheapest path from the current strategy to the previous strategy is saved after four calculations are made as described in the example in step a.

c. Repeat step a and b for a new unit combination

until we have a specified number of strategies.

d. Select a specified number of the most economical strategies out of the results of step c.

DESCRIPTION OF THE UNIT COMMITMENT PROCEDURES

This is the general description of a computer program that apply all of these above proposals into the unit commitment of a system that includes both hydro and thermal energy. The program takes into account the following restrictions of minimum up and down-time, spinning reserve, must on-time, must off-time, start-up cost that includes cold-start and hot-start and maintenance time.

The 'Unit commitment' procedure is the main program of the hydro-thermal unit commitment. It first initializes all the information related to the system such as unit identification of the hydro or the thermal units, minimum up and down-time, spinning reserve, start-up cost (cold-start and hot-start), maintenance time, etc of each unit. The thermal energy cut off points are calculated next. The hydro units are committed. Then, we use the above result to calculate and set up the search table for all the feasible combinations of thermal units as described in previous sections. Next, we compute the feasible strategies every hour (including both thermal and hydro units). Finally, look at all the strategies in the last hour and find the cheapest strategy. From there we go backward and get the most

economical path. The explanation of the algorithm below demonstrates the method to calculate the strategies in each hour. Let's call n the number of hours, the procedure is called n times to calculate the strategies of the system planning in a period of n hours.

Procedure : Unit commitment

begin

initialize all the global system information.

find the thermal energy cut off points of every hour.

set up the ordered list for the hydro units.

commit the hydro units.

set up the search table for all the feasible combinations of
all the thermal units.

;Apply dynamic programming algorithm.

for(hour = 1 to number_of_hours)

calculate the strategies in each hour.

if there is solution at the final hour then

begin

find the optimal path.

print out the best result

end

else

message "there is no solution"

end

The procedure below determines the thermal energy cut off points in each hour.

Procedure : Find the thermal energy cut off point in each hour.

begin

find the largest energy demand per hour in the energy schedule and set it as the initial cut off point.

get the total available hydro energy capacity.

done = false

initialize step_size.

set the energy demand in each hour as thermal energy.

;locate the thermal energy cut off points for every hour.

while not done

begin

cut off point = cut off point - step size

;locate the cut off point for each hour

for(hour = 1 to number of hours) and not done

begin

hydro capacity = hydro capacity - step size

if(hydro capacity \geq 0) then

begin

if (the total allocated hydro energy \geq the maximum hydro energy can be supplied by the available hydro units) or ((energy demand - hydro energy) $>$ greatest minimum energy of any thermal unit) then

do not allocate any more hydro energy

```

        for this hour.
    else if( energy demand > cut off point ) then
        allocate a small amount of hydro energy
        and reduce the thermal energy in this
        hour by the same amount.
    end
else
    done = true
end
end
end
end

```

After the thermal energy cut off points are determined, the hydro energy is allocated in each hour and the hydro units are committed.

Procedure : Commit hydro units

begin

arrange the amounts of the allocated hydro energy in each hour such that the largest amount first and the least last.

for(hour = 1 to number of hydro energy hours)

begin

get the hydro energy demand in the particular hour

done = false

set all the bits OFF.

for((bit = 0 to number of hydro unit - 1) or done)

begin

```

;check this unit still has available energy
if energy capacity of the unit > Pmin of the unit
    then
        begin
            turn ON this bit.    ;a hydro unit.
            energy capacity of the unit = energy capacity
                of the unit - energy supply
            hydro energy demand = hydro energy demand -
                hydro energy supply
            if the hydro energy demand is 0 then
                done = true
            end
        else
            turn OFF this bit.    ;a hydro unit
        end
    end
end
save the total hydro energy supply.
end

```

With the application of the ordered list, the procedure 'turn ON a hydro unit' commits a hydro unit. We always need to keep track of the total energy it has produced. We need to save that information because the hydro units have limited energy capacity. We use the information together with its limited energy capacity to figure out how much energy is still available in order to commit them.

Procedure : Turn ON a hydro unit.

```
begin
calculate the available hydro energy of the unit.
hydro energy capacity of the unit = the initial hydro energy
    capacity - total hydro energy consumed.
if the hydro energy capacity >= the required hydro energy
    then
        allocatable hydro energy = required hydro energy.
else
    allocatable hydro energy = hydro energy capacity.
if allocatable hydro energy > 0 then
    begin
        if allocatable hydro energy > maximum energy of the
            unit then
                begin
                    use the maximum energy of this unit.
                    calculate the production cost for this amount of
                        energy.
                    add this production cost to the total production
                        cost of the current state of the hydro units.
                    save this information into a temporary_strategy.
                    hydro energy demand is reduced by this maximum
                        hydro energy.
                end
            else if allocatable hydro energy >= minimum energy of
                the unit then
                    begin
```

```
use the allocatable hydro energy of this unit.  
save this information into a temporary_strategy.  
hydro energy demand is reduced by this amount  
    hydro energy.
```

```
end
```

```
else if allocatable hydro energy < minimum energy of  
the unit then
```

```
begin
```

```
if hydro energy capacity left < minimum energy of  
the unit then turn OFF this unit.
```

```
else
```

```
begin
```

```
use the minimum hydro energy of this unit.
```

```
end
```

```
save this information into a temporary_strategy.
```

```
hydro energy demand = 0.
```

```
end
```

```
else if allocatable hydro energy is 0 then
```

```
turn OFF this unit.
```

```
end
```

```
use the previous strategy to update the unit on time sofar.
```

```
update the total energy consumed sofar of this unit.
```

```
end
```

```
save this hydro unit combination with all the status
```

```
information of the units.
```

```
end
```


Now, the search table is set up with the procedure below.

Procedure : Set up the search table.

begin

calculate the full load average for all the units.

use an unsigned integer such that the least significant bit represents for the unit that has least expensive full load average cost.

find the smallest energy in the energy schedule.

number of combinations = 0

;generate all the combinations

for(number of combinations = 0 to $2^n - 1$)

begin

total energy of this combination is 0

;calculate the maximum energy of each combination

for(bit = 0 to (n - 1))

begin

if this bit is 1 then add the maximum energy that the corresponding unit can supply to total energy of this combination.

if the total energy of this combination is greater than the smallest energy then

begin

calculate the pcost of the combination.

save this combination, the total energy and the total cost into the search table.

increment the total number of combinations in
the search table.

end

end

end

sort the search table in ascending order according to the
total cost of the max power.

sort the search table in ascending order according to the
total energy of each combination.

end

The procedure 'Calculate the strategies in each hour' expects a set of strategies of the previous hour as a general case. For the first hour this procedure is called, a previous strategy is initialized with all the units are off and the total production cost is zero for hour zero. This procedure produces a new set of strategies in each hour. The first section sets up all the conditions of each unit such as the unit must be OFF, ON or maintained etc. The second section mainly commits the thermal units. The last one is selecting the best strategies of the current hours out of the number of feasible strategies calculated in the previous sections.

All the information we need to know in order to commit the units are the thermal and hydro energy demand at the time in question. A set of bit masks is created to denote the conditions of the units must be met. If the units must

be ON in this hour, the corresponding bits are set to ON and the bit mask is called must ON mask, and the same rule is applied to set the MAINTENANCE mask for the units must be in maintenance condition and must OFF mask for the units must be in the must-OFF condition. The minimum OFF and ON time masks are also created. Then some other initialization takes place for house keeping purpose.

The dynamic programming approach is used for the thermal unit commitment. The process will involve selecting a number of combinations of the units that meet all the conditions required in the current hour called states, then calculate the production cost for the current state and the total production cost from the initial strategy up to this state. The status of all the units are also recorded such as the up-time, down-time etc.

The last part of the procedure selects some states that are the most economical out of a number of states produced by the above process and save them. They are called the strategies of the current hour. This completes the dynamic programming process for the current hour and these strategies will be used as previous strategies in this procedure for the next hour.

Procedure : Calculate the strategies in each hour.

begin

get the thermal energy demand in this hour from the energy
schedule.

all the bits correspond to the units to be maintained are
 set.

all the bits correspond to the units to be ON are set.

all the bits correspond to the units to be OFF line are set.

create a minimum OFF time mask for each strategy with all
 the ON bits correspond to the units that do not satisfy
 the minimum OFF time.

create a minimum ON time mask for each strategy with all the
 ON bits correspond to the units that do not satisfy the
 minimum ON time.

number of states = 0

;commit the thermal units :

for (current_state_id = 0 to number of thermal combinations)
 and (number of states <= number of calculated
 strategies)

begin

feasible combination = select a feasible combination.

; this routine updates the current_state_id

if feasible_state is valid then

begin

;calculate the solutions of a feasible combination

valid state = calculate a state from all the

previous strategies.

if there is valid state then

begin

select the most economical state.

```

save it into an array of calculated
    strategies and check it valid.
;save all the information related to this
    strategy
next state
end

```

```
end
```

```
end
```

```

now select the most economical strategies out of the array
    of calculated strategies and save them into system
    strategies for the current hour and check it valid.
end

```

Procedure 'select a feasible combinations' is a part of the unit commitment process in each hour. A feasible combination of the thermal unit must meet all the conditions applied to each unit such as some units must be ON, some others must be OFF and some must be MAINTAINED off line. The combination must be able to supply the required thermal energy plus the spinning reserve in the current hour. Finally, it must satisfy the minimum ON and OFF time.

Procedure : Select a feasible combination.

```
begin
```

```

select the thermal combination that satisfies the thermal
    energy demand plus spinning reserve.

```

```

;if the combination satisfies all these conditions : must be
    on, off, maintained

```

```
if ( must_on_mask == combination and must_on_mask ) and
    ((combination and must_off_mask) > 0) and ((combination
and maintenance) > 0) then
    return this combination is valid.
else
    return this combination is invalid.
end
```

For each feasible combination, procedure 'calculate a state from all the previous strategies' will compute the production cost for it during a particular hour and the total production cost since the initial strategy up to and including the state of the unit combination of the current hour. This can be accomplished by calculating the total production costs from each of the previous strategy and select only one best result and save it. The selected combination is called a strategy. We keep continue finding a number of states as we want. Then only a selected number of strategies that prove to be the most economical are picked out. They are called strategies of the current hour. The procedure below describes the process to select a strategy out of the number of states of the current combination from all the previous strategies. The state is a unit combination that has the minimum total production cost from the initial strategy up to and including itself.

Procedure : Calculate a state from all the previous strategies.

```
begin
initialize temporary strategy list = invalid
initialize the solution list for all the solutions that will
    be calculated
solution = no solution
;calculate the solutions in the current hour from
;all the strategies of the previous hour.
for (previous_strategy = 1 to number of strategies of the
    previous hour)
    begin
        thermal energy = thermal energy demand - allocated
            hydro energy.
        ck_solution = calculate a solution of the current hour
            from a strategy of the previous hour.
        if ck_solution == has_solution then
            begin
                at_least_one_solution = true.
                check this solution valid.
                save the current combination.
                save the previous strategy id.
                save the total production cost sofar.
                for (unit = 0 to number of hydro units - 1)
                    begin
                        if the current unit is a hydro plant then
                            save the hydro energy consumed sofar.
                    end
            end
    end
```

```
        end
    end
    if there is at least one solution then take the best
        solution out of the number of solutions calculated
        above and save it as a state.
    return (there is(not) state)
end
Procedure : Calculate a solution in the current hour from a
    strategy of the previous hour.
begin
    if previous strategy is valid then
        begin
            if((combination and minimum_off_time_mask != 0 ) or
                (combination and minimum_ON_time_mask !=
                    minimum_ON_time mask)) then
                return no_solution.
            initialize a temporary storage to save a strategy to be
                calculated.
            ck_solution = check for the transition of all the units
                from a previous strategy to the current state and
                calculate the energy supply, the production cost
                of each unit, the total production cost of this
                state and the total production cost from the
                initial state up to this state.
            end
        return (there is (not) solution : ck_solution )
```



```

then
    on_to_on.
    if the same unit in the previous state is OFF then
        off_to_on
    end
if the unit in this state is OFF then
    begin
        if the same unit in the previous state is ON then
            on_to_off
        if the same unit in the previous state is OFF then
            off_to_off
        end
    end
end
end

```

This procedure 'unit is from ON to ON' takes care of the case of a unit transition from ON to ON in the current hour. For thermal unit, it commits the amount of thermal energy supply and calculate the production cost. Then, it will reduce the required thermal energy by the amount of its energy supply so that we know how much thermal energy demand left in order to commit the other units. After all, it uses the available up-time information of the unit in the previous strategy to update the time it has been ON. This information is very important, not only we need to know how much time the unit has been ON but it is also used to check for its valid state transition every hour to see if the

required minimum up-time is met before the unit is committed

Procedure : The transition of a unit from ON to ON.

begin

if the required thermal energy \geq maximum energy of the unit

then

begin

use maximum energy of this unit.

calculate the production cost for this amount of
energy.

add this cost to the total production cost of the
current state.

end

else if the required thermal energy \geq minimum energy of the
unit then

begin

use the required thermal energy of this unit.

calculate the production cost for this amount of
energy.

add this production cost to the total production cost
of the current state.

the thermal energy demand is 0.

end

else if the thermal energy demand $<$ minimum energy of the
unit but not 0 then

begin

use the minimum energy of this unit.

```
calculate the production cost for this amount of
    energy.
add this production cost to the total production cost
    of the current state.
the required thermal energy is 0.
end
else if the required thermal energy is 0 then
begin
use 0 energy of this unit.
calculate the production cost for this amount of
    energy.
;there is a fixed production cost here
add this production cost to the total production cost
    of the current state.
the thermal energy demand is 0.
    end
use the previous strategy information to update the
    total on time sofar of this unit.
end
end
Procedure : Unit has a transition from OFF to ON.
begin
if the thermal energy demand >= maximum energy of the unit
    then
begin
use maximum energy of this unit.
```

calculate the production cost for this amount of
thermal energy.

add this production cost to the total production cost
of the current state.

thermal energy demand is reduced by this maximum
energy.

end

else if the thermal energy demand \geq minimum energy of the
unit then

begin

use the thermal energy demand of this unit.

calculate the production cost for this amount of
thermal energy.

add this production cost to the total production cost
of the current state.

thermal energy demand = 0.

end

else if the thermal energy demand $<$ minimum energy of the
unit but not 0 then

begin

use the minimum energy of this unit.

calculate the production cost for this amount of
thermal energy.

add this production cost to the total production cost
of the current state.

thermal energy = 0.

```

end
else if the thermal energy demand is 0 then
begin
use the 0 power of this unit
calculate the production cost for this amount of
    thermal energy.
;there is a fixed production cost here
add this production cost to the total production cost
    of the current state.
    thermal energy demand is 0.
end
use the previous strategy information to update the total on
    time sofar of this unit.
calculate the transition production cost .
unit OFF time = satisfied.
unit ON time = 1.
end

```

The following procedures check the status of the unit when a transition takes place from ON to OFF and from OFF to OFF.

Procedure : Unit has a transition from ON to OFF.

```

begin
set the unit on time is satisfied.
unit OFF time = 1.
end

```

Procedure : unit has a transition OFF to OFF.

```
begin  
increment the unit OFF time.  
end
```

CHAPTER IV

EXAMPLE

A hydro-thermal system consists of 3 thermal and two hydro units. Spinning reserve is 50 MW. Table XXI to XXV describes the data of all the units.

TABLE XXI
DATA OF THE THERMAL UNITS

Unit	Power Generation		a0	a1	a2
	Max (MW)	Min (MW)			
2	300	25	150	20.8	0.002
1	300	60	120	18.0	0.005
3	600	20	100	23.8	0.003

TABLE XXII
START-UP COST

Unit	Cold-start Cost (Dollars)	Hot-start Cost (Dollars)	Minimum On Time (hours)	Minimum Off Time (hours)	Cold-start Time (hours)
2	150	75	1	2	3
1	170	80	1	3	5
3	100	45	2	2	3

TABLE XXIII
MUST ON AND OFF TIME

Unit	Must ON (Hours)	Must Off (Hours)	From Time	To Time
1		yes	1	3
3	yes		6	7

TABLE XXIV
MAINTENANCE SCHEDULE

Unit	From Time (Hours)	To Time (Hours)
2	2	3

TABLE XXV
DATA OF THE HYDRO UNITS

Unit	Power Generation Max (MW)	Min (MW)	Capacity (MWh)
0	100	1	300
4	100	1	300

The energy schedule is 450, 530, 600, 1000, 300, 290 and 700 (MWh) .

The search table is created as shown in Table XXVI.

TABLE XXVI
THE SEARCH TABLE

Unit	Maximum Power (MW)	Fuel Cost (Dollars)
3 2 1		
0 0 1	300	5970.00
0 1 0	300	6570.00
0 1 1	600	12540.00
1 0 0	600	15460.00
1 0 1	900	21430.00
1 1 0	900	22030.00
1 1 1	1200	28000.00

The commitment schedules are shown below. In each hour,

the first row is the power generation in MW , second row is fuel cost and third row is the start-up cost in dollars.

TABLE XXVII
THE UNIT COMMITMENT SCHEDULE

Unit : 3	2	1	4	0	Production Cost (Dollars)	Total Production Cost (Dollars)

hour : 1						
ON	ON	OFF	OFF	OFF		
150.00	300.00	0.00	0.00	0.00		
3737.50	6570.00	0.00	0.00	0.00		
100.00	150.00	0.00	0.00	0.00		
					10557.50	10557.50
hour : 2						
ON	OFF	OFF	ON	ON		
466.00	0.00	0.00	64.00	0.00		
11842.27	0.00	0.00	0.00	0.00		
0.00	0.00	0.00	0.00	0.00		
					11842.27	22399.77
hour : 3						
ON	OFF	OFF	ON	ON		
466.00	0.00	0.00	34.00	100.00		
11842.27	0.00	0.00	0.00	0.00		
0.00	0.00	0.00	0.00	0.00		
					11842.27	34242.04
hour : 4						
ON	ON	ON	ON	ON		
201.00	300.00	300.00	99.00	100.00		
5005.00	6570.00	5970.00	0.00	0.00		
0.00	75.00	170.00	0.00	0.00		
					17790.00	52032.04
hour : 5						
ON	OFF	ON	OFF	OFF		
0.00	0.00	300.00	0.00	0.00		
100.00	0.00	5970.00	0.00	0.00		
0.00	0.00	0.00	0.00	0.00		
					6070.00	58102.04
hour : 6						
ON	OFF	ON	OFF	OFF		
0.00	0.00	290.00	0.00	0.00		
100.00	0.00	5760.50	0.00	0.00		
0.00	0.00	0.00	0.00	0.00		
					5860.50	63962.54
hour : 7						
ON	ON	ON	ON	ON		
0.00	201.00	300.00	99.00	100.00		
100.00	4411.60	5970.00	0.00	0.00		
0.00	75.00	0.00	0.00	0.00		
					10556.60	74519.14

TABLE XXVIII

THE UNIT COMMITMENT SCHEDULE IF THE MAINTENANCE
SCHEDULE OF UNIT 2 IS OPTIMIZED

Unit : 3	2	1	4	0	Production Cost (Dollars)	Total Production (Dollars)
hour : 1						
ON	ON	OFF	OFF	OFF		
150.00	300.00	0.00	0.00	0.00		
3737.50	6570.00	0.00	0.00	0.00		
100.00	150.00	0.00	0.00	0.00		
					10557.50	10557.50
hour : 2						
ON	ON	OFF	ON	ON		
166.00	300.00	0.00	64.00	0.00		
4133.47	6570.00	0.00	0.00	0.00		
0.00	0.00	0.00	0.00	0.00		
					10703.47	21260.97
hour : 3						
ON	ON	OFF	ON	ON		
166.00	300.00	0.00	34.00	100.00		
4133.47	6570.00	0.00	0.00	0.00		
0.00	0.00	0.00	0.00	0.00		
					10703.47	31964.44
hour : 4						
ON	ON	ON	ON	ON		
201.00	300.00	300.00	99.00	100.00		
5005.00	6570.00	5970.00	0.00	0.00		
0.00	0.00	170.00	0.00	0.00		
					17715.00	49679.44
hour : 5						
ON	OFF	ON	OFF	OFF		
0.00	0.00	300.00	0.00	0.00		
100.00	0.00	5970.00	0.00	0.00		
0.00	0.00	0.00	0.00	0.00		
					6070.00	55749.44
hour : 6						
ON	OFF	ON	OFF	OFF		
0.00	0.00	290.00	0.00	0.00		
100.00	0.00	5760.50	0.00	0.00		
0.00	0.00	0.00	0.00	0.00		
					5860.50	61609.94
hour : 7						
ON	ON	ON	ON	ON		
0.00	201.00	300.00	99.00	100.00		
100.00	4411.60	5970.00	0.00	0.00		
0.00	75.00	0.00	0.00	0.00		
					10556.60	72166.55

CHAPTER V

CONCLUSION

This thesis presents a method of committing generating units in a hydro-thermal power system within practical computer resources such as computer time and data storage. One way to solve this problem would be the dynamic programming methods similar to that proposed in references [6, 1].

One such method [6] uses two stages to solve the unit commitment called suboptimizer and optimizer. In the suboptimizer stage, the thermal units are committed to satisfy the energy demand for the commitment period first using the dynamic programming. The hydro energy scheduling is specified later. The optimizer optimizes the energy schedule by improving the start-up costs of all the units. This method leads to an optimal unit commitment but requires a lot of computational time and data storage.

The method presented in this thesis applies two different approaches which are the ordered list and the dynamic programming to solve the unit commitment of the hydro and thermal units. This can be done because the units have two different types of constraints. Each approach has its own strength to be the most suitable to each type of

units.

The combination of the two approaches is feasible because the optimal scheduling of the hydro energy is found before the thermal unit commitment proceeds. The optimal allocation of the hydro energy is made whenever the energy demand is peak based as the fuel cost of the thermal units increases quadratically with their supplied energy. The fuel cost of the thermal units per energy unit is minimized by this hydro energy allocation scheme.

The entire hydro energy capacity is used by committing the hydro units in the hour which demands the most hydro energy first and the least last. The ordered list used to commit the hydro units is simple and requires a minimal computer resource.

The commitment of the thermal units is quite complex due to many constraints needed to be considered. The dynamic programming is effectively applied to the thermal system. The search table is arranged in such a way to optimize the time to search for a feasible unit combination in each hour. The order to commit the units of each combination maximizes the use of the most economical units. The use of the minimum ON and OFF time masks significantly reduce the time to reject or accept a combination for the minimum OFF and ON time condition.

The efficient use of the above strategies makes the unit commitment program run effectively within a practical

time and the limited main memory of a personal computer. The program using the method described in this thesis ran on a IBM-PCAT 286 personal computer with one mega bytes of DRAM to solve the unit commitment of ten units in a period of 24 hours.

Fully applying the dynamic programming approach for the thermal unit commitment makes this method only suitable for an optimal short-term unit commitment for the system due to the limit of computer main memory using as data storage to compute the thermal energy schedule (the computer resource only depends on the number of thermal units). However, an improvement can be made by not depending on the main memory to store all the strategies in each stage but using virtual memory such as hard disk to store all the strategies. This scheme will save up to 80% of the computer resource and the number of thermal units included in the system can be drastically increased.

For a larger system, a more heuristic approach that is truncated dynamic programming may be used to replace the current dynamic programming to commit the thermal units [4, 5].

The truncated dynamic programming unit commitment procedure is divided into two major parts. The first involves the formation of a unit selection list and the other part consists of a search technique which determines optimal or near optimal feasible schedules for a given

period.

The purpose of having a unit selection list is to reduce the number of combinations of the thermal units that need to be examined at each hour. The formation of the selection list is based upon an ordered list of all the thermal units in the system and the criterion for placing individual units at various priority levels is the average incremental production cost.

Example :

A system contains 10 units, four of them must be on line at a given hour. These units form the first four units in the selection list. The other units are added onto the selection list in a prespecified ordered priority and they must satisfy all the necessary conditions such as minimum ON time or OFF time, etc. Then, determine the minimum number of units which must be on-line in the order of the selection list to meet the required load.

Assume unit 1 to unit 6 are in the selection list, then a search is performed about unit 6 such that some units above and below the unit may be included. This search range would be defined by specifying the number of units below unit 6 and the number of units in the search range.

Example:

If the number of unit below unit 6 is one and the number of units in the search range is 3, then the search range from unit 5 through unit 7 is specified. There are 3

units in the search range, thus yielding $2^3 - 1 = 7$ combinations to be examined. Table XXIX shows the number of combinations under consideration.

TABLE XXIX
THE NUMBER OF COMBINATIONS IN
THE SEARCH RANGE

Unit						
7	6	5	4	3	2	1
0	0	1	1	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	1	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

By varying the search range, the extent of the search for economic and feasible unit commitment schedules can be controlled.

Having define the search range and the probable combinations of units for an hour in the study period, the next step is to determine economically feasible unit commitment schedule. The dynamic programming is then applied as usual on the selected combinations. This method proved to be quite effective [4].

The presented method in this thesis would solve unit

commitment for a larger system effectively in term of computational time and computer data storage with the following strategies added:

1. Apply the selection of the units described in the truncated dynamic programming.
2. Store all the resulted strategies in each hour into virtual memory instead of main memory.

REFERENCES

- [1] Allen J. Wood, Bruce F. Wollenberg, 'Power generation operation & control', John Wiley & Sons, 1984.
- [2] Ellis Horowitz, Sartaj Sahni, 'Fundamentals of computer algorithms', Computer Science Press, 1984.
- [3] Robert Sedgewick, 'Algorithms', Addison - Wesley, 1983.
- [4] C. K. Pang, G. B. Sheble, F. Albuyeh, 'Evaluation of Dynamic Programming based methods and multiple area representation for thermal unit commitments', IEEE Trans., Vol. PAS-100, pp. 1212 - 1218, March 1981.
- [5] C. K. Pang, H. C. Chen, 'Optimal short-term thermal unit commitment', IEEE Trans., Vol. PAS-95, pp. 1336 - 1341, July/August 1976.
- [6] H. H. Happ, R. C. Johnson, W. J. Wright, 'Large scale hydro-thermal unit commitment method and results', IEEE Trans., Vol. PAS-90, pp 1373 - 1384, 1971.
- [7] P.G. Lowery, 'Generating unit commitment by Dynamic Programming', IEEE Trans., Vol. PAS-85, pp 422 - 426, 1966.
- [8] A. K. Ayoub and A. D. Patton, 'Optimal thermal generating unit commitment', IEEE Trans., Vol. PAS-90, pp. 1752 - 1756, 1971.
- [9] R. E. Bellman, 'Dynamic Programming', Princeton University Press, Princeton, New Jersey, 1957.
- [10] C. J. Baldwin, K. M. Dale and R. F. Dittrich, 'A study of the economic shutdown of generating units in daily dispatch', AIEE Trans., Vol. 78, part III, pp. 1272 - 1284, 1959.
- [11] J. D. Guy, 'Security Constrained unit commitment', IEEE Trans., Vol. SA-90, pp. 1385 - 1390, 1971.
- [12] Neuhauser, G. L., 'Introduction to Dynamic Programming', Wiley, New York, 1966.

- [13] B. Bernholtz, L. J. Graham, 'Hydrothermal Economic Scheduling', Part I, Solution by Incremental Dynamic Programming, AIEEE Trans, Vol. 79, pp 921 - 932, Dec 1960.
- [14] Intel, 'iAPX 286 Hardware Reference Manual', Intel, 1987.
- [15] Intel, 'iAPX 80286 and 80287 Programmers's Reference Manual', Intel, 1987.