TR-85-24

# Stochastic Unit Commitment Scheduling and Dispatch of Electric Power Systems

By

I. Yan G.L. Blankenship

# STOCHASTIC UNIT COMMITMENT SCHEDULING AND DISPATCH OF ELECTRIC POWER SYSTEMS

I. Yan, Student Member IEEE    G. L. Blankenship, Fellow IEEE

Electrical Engineering Department
University of Maryland
College Park, MD 20742

## ABSTRACT

Unit commitment, including economic dispatch, is a key component of short term operation scheduling of an electric energy system. Common industry practice is based on the use of a "priority list" for generation scheduling and a deterministic model for power/energy demand. The priority list specifies the next unit to be started or shutdown in response to an increase or decrease in load. A common problem in the use of priority lists is that the next unit is improperly sized to meet the actual change in load.

The algorithm proposed here is more accurate than the priority list method and much faster than dynamic programming which can hardly be applied to systems of more 5 machines. For a system of 41 machines, the algorithm can determine schedules in 0.1 second which is fast enough to do on-line control. Furthermore, the total generating cost is superior to that obtained by dynamic programming successive approximations.

## 1. INTRODUCTION

In this paper we propose a scheduling algorithm (including economic dispatch) which is fast enough to be used for on-line scheduling in response to *random* changes in demand. The algorithm does not represent a radical departure from current practice. It uses a quadratic function for the fuel cost (heat rate) of thermal units, and a standard exponential function of the cumulative down time to model unit startup costs. A pumped storage facility which may be a composite of several (pumped and unpumped) energy storage systems is included. It uses a *scheduling table* for the generation system which is reminiscent of the priority list, but more comprehensive; and it has an efficient "off-line" scheduling procedure to compute the optimal unit commitment/economic dispatch to meet the (deterministic) expected demand. Based on the solution to the deterministic scheduling problem, a fast, "on-line" algorithm is applied to adjust the commitment and dispatch in response to random fluctuations in demand. The algorithm requires about 0.1 sec to reschedule 41 machines (on a VAX 11/780) over 1 time step. The random fluctuations in load are modeled by white noise or a (non-stationary) Gaussian-Markov process. In the second case, a Kalman filter is used to compute the one-step ahead prediction of the load.

Performance tests for the algorithm are given for systems of 3, 5, 18 and 41 machines. The optimal costs computed are nearly identical to those computed using dynamic programming [1] or a modified dynamic programming successive approximations algorithm [2]. The tests show that the CPU times required to set up the scheduling table and execute the on-line scheduling algorithm grow slower than linearly in the number of machines in the system.

The design and execution of the algorithm are based on the (implicit) assumptions that the starting costs of the generators are substantially smaller than the operating (fuel) costs. We also assume that the short term fluctuations in the load are a small percentage (approximately 4% or less) of the mean load level. The performance of our algorithm reflects the fact that the sensitivity of the system operating cost to perturbations in the demand and "small" changes in the commitment schedule decrease substantially as the number of machines in the system increases. (This fact was exploited in a different way in [3].) We have used a simple model for the scheduling problem. Enhancements to include run time constraints, more complex (Gaussian) load models, and storage system costs (shadow prices) would not substantially impact the performance of the algorithm. Including transmission costs in the scheduling problem would require redesign of the algorithm.

Almost all previous work on unit commitment scheduling has been based on deterministic load models. Work similar to ours in spirit includes the paper of Bertsekas et al[3] who use a duality formulation of the optimal scheduling problem, the work of Turgeon [4] [5] who uses the maximum principle to treat deterministic model similar to the one used here, the thesis of Leguay [6] who uses "impulse control" methods to treat small scale, deterministic scheduling problems, and the recent work of Gonzalez and Rofman [7] [8] who use a clever combinational algorithm to treat modest sized, deterministic scheduling problems including costs for the storage systems (shadow prices). Mathematical treatments of stochastic unit commitment problems were given in Blankenship and Menaldi [9], Li and Blankenship [10]. Starnes's thesis [2] contains an effective *ad hoc* algorithm for treating (deterministic) scheduling problem with more than one performance measure (e.g. system security and cost). More details on the present work may be found in Yan's thesis [11].

## 2. SYSTEM MODEL

We consider a system with M machines (thermal or nuclear units) operating over an interval [0,T] (one day or one week). We assume that the system includes a (composite) energy storage system (pumped hydro). The unit commitment problem is to schedule the startup, operating level, and shutdown of the thermal units and the pumping and withdrawal of energy to and from the storage system to meet the time varying demand for power, L(t), t=0,1,2,...,T, at minimum operating cost. In mathematical terms the problem is

$$\min_{t,i} \sum_{t=1}^{T} \sum_{i=1}^{M} [CG(t,i) + CS(t,i)] \qquad (2.1)$$

subject to the constraints

$$G(t) = \sum_{i=1}^{M} G(t,i) = L(t) + r(t) \qquad (2.2)$$

$$\underline{G}(i) \le G(t,i) \le \overline{G}(i)$$

$$\underline{R} \le r(t) \le R \qquad (2.3)$$

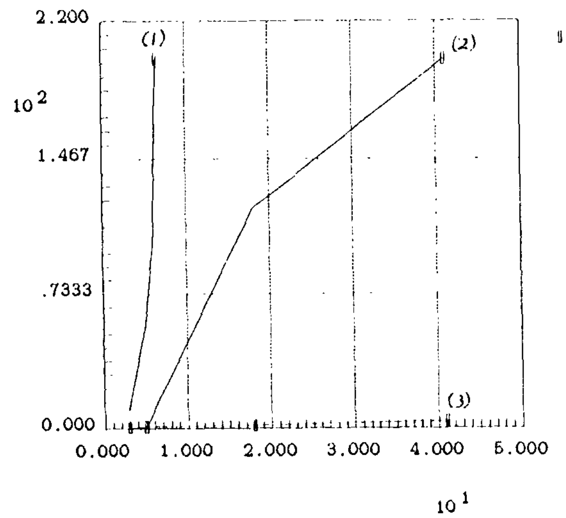
$$S(t) = \min\{\overline{S}, max\{\underline{S}, S(t-1) + r(t-1)\}\} \qquad (2.4)$$

11/780 operating under Berkeley Unix version 4.1.

| NO. OF MACHINES | BUILD UP SCHEDULING TABLE | ITERATION TO OBTAIN 24 HOUR OPTIMAL SCHEDULE |
|---|---|---|
| 03 | 000:30 | 0:42 |
| 05 | 000:50 | 1:10 |
| 18 | 120:00 | 1:35 |
| 41 | 200:00 | 3:50 |

CPU TIME(MIN:SEC) (OFF-LINE)
Table 2. EXECUTION TIME FOR DETERMINISTIC CASES

The trend in computational times versus number of machines is important. Figure 3 shows that both the time required to build the scheduling table and the unit iteration time for the deterministic algorithm have less than linear growth as functions of machine numbers. The search procedure used to construct the scheduling table was not optimized, and the computational times for this procedure could be substantially reduced if effective search procedures were used.




curve (1) for DPSA
curve (2) for building table
curve (3) for deterministic load iteration
CPU time is in 100 minutes.
Figure 3. CPU TIME VS NUMBER OF MACHINES

## 6. ON – LINE STOCHASTIC SCHEDULING

The off-line algorithm determines a schedule and a set of optimal transfer rates $\bar{r}(1),...,\bar{r}(T)$ corresponding to a known, deterministic demand $\bar{L}(t), t = 0,1,...,T$. Starting with $\bar{r}(1),...,\bar{r}(T)$ and $\bar{G}(t,i), i = 1,2,...,M$, the on-line algorithm may be used to determine the best schedule (including starting costs), G(t,i), i=1,...,M., and optimal transfer rates r(1),...,r(T) to meet the actual, observed (random) demand L(t), t=0,1,...,T. The algorithm is fast enough to be used on-line.

In testing the algorithm we used two different models for the demand. The first was simply

$$L(t) = \bar{L}(t) + \delta L(t) \qquad (6.1)$$

with $\delta L(t)$ a sequence of independent zero mean random variables uniformly distributed in the interval $[-\alpha\bar{L}(t), \alpha\bar{L}(t)]$ with $\alpha$ a small number. In the tests we used $\alpha = 0.04$, so the demand fluctuations were less than or equal to 4% of the mean at all time steps.

In the second demand model, we assumed that the demand fluctuations were a first order Gaussian-Markov process

$$\delta L(t+1) = a * \delta L(t) + w(t) \qquad (6.2)$$

with 'a' a real constant and $w(t), t = 0,1,....$ a sequence of zero mean, independent, identically distributed Gaussian-Markov process(i.e. a Gaussian white noise process). We used the model

$$y(t) = \delta L(t) + v(t) \qquad (6.3)$$

with $v(t)$ Gaussian white noise process to describe the measurements of the demand fluctuations. We assume

$$E[v(t)w(s)] = 0 \qquad \forall\, t,s$$

$$E[v(t)\delta L(0)] = 0 = E[w(s)\delta L(0)] \qquad \forall\, t,s \qquad (6.4)$$

$$E[w(t)w(s)] = W(t) * \delta_{st}$$

$$E[v(t)v(s)] = V(t) * \delta_{st}$$

$$E[\delta L(0)] = 0, \quad E[(\delta L(t))^2] = \sigma(t)$$

$$\delta_{st} = \begin{cases} 0, & s \neq t \\ 1, & s = t \end{cases} \qquad (6.5)$$

(The parameters a,$w(t)$,$v(t)$ and $\sigma(t)$ must be identified from actual load data.)

The Kalman filter is the best estimator of $\delta L(t)$ given $y(s)$, $s \leq t-1$. The equations are

$$\delta L(t+1|t) = a * \delta L(t|t-1) + K(t) * [y(t) - \delta L(t|t-)] \qquad (6.6)$$

where $\delta L(t|t-1)$ is the best (linear) estimate of $\delta L(t)$ given $y(s), s \leq t-1$, and

$$K(t) = \frac{a * P(t)}{P(t) + V(t)}$$

$$P(t+1) = a * P(t) - \frac{a * P(t)^2}{P(t) + V(t)} + W(t) \qquad (6.7)$$

$$P(0) = \sigma(0)$$

Both K(t) and P(t) can be computed off-line.

The on-line scheduling algorithm based on the Gaussian-Markov load model is shown in Figure 4. the same algorithm is used for the random fluctuations model, with $\delta L(t+1|t)$ in that Gaussian-Markov model substituted by $\delta L(t)$ in the Figure.

The overall scheduling algorithm is shown in Figure 5.

## 7. ON – LINE SCHEDULING FOR 41 MACHINES

In conducting the schedule computations 80 days of (synthetic) hourly demand were used to generate the random load statistics. A break down of the computational times for the algorithm in the two model cases is given in Figure 6. A plot of the computational times required to find the optimal hourly schedule for the two different random demand models versus number of machines is given in Figure 7. Note that the rate of increase is less than linear. The times are longer when the algorithm includes the Kalman predictor, since time is required to execute the additional lines of code. (For the 41 machine case 87 seconds were required to find the optimal schedule for 40 days in 1 hour increments.)

# STOCHASTIC UNIT COMMITMENT SCHEDULING AND DISPATCH OF ELECTRIC POWER SYSTEMS

I. Yan, Student Member IEEE G. L. Blankenship, Fellow IEEE

Electrical Engineering Department
University of Maryland
College Park, MD 20742

*ABSTRACT*

Unit commitment, including economic dispatch, is a key component of short term operation scheduling of an electric energy system. Common industry practice is based on the use of a "priority list" for generation scheduling and a deterministic model for power/energy demand. The priority list specifies the next unit to be started or shutdown in response to an increase or decrease in load. A common problem in the use of priority lists is that the next unit is improperly sized to meet the actual change in load.

The algorithm proposed here is more accurate than the priority list method and much faster than dynamic programming which can hardly be applied to systems of more 5 machines. For a system of 41 machines, the algorithm can determine schedules in 0.1 second which is fast enough to do on-line control. Furthermore, the total generating cost is superior to that obtained by dynamic programming successive approximations.

## 1. *INTRODUCTION*

In this paper we propose a scheduling algorithm (including economic dispatch) which is fast enough to be used for on-line scheduling in response to *random* changes in demand. The algorithm does not represent a radical departure from current practice. It uses a quadratic function for the fuel cost (heat rate) of thermal units, and a standard exponential function of the cumulative down time to model unit startup costs. A pumped storage facility which may be a composite of several (pumped and unpumped) energy storage systems is included. It uses a *scheduling table* for the generation system which is reminiscient of the priority list, but more comprehensive; and it has an efficient "off-line" scheduling procedure to compute the optimal unit commitment/economic dispatch to meet the (deterministic) expected demand. Based on the solution to the deterministic scheduling problem, a fast, "on-line" algorithm is applied to adjust the commitment and dispatch in response to random fluctuations in demand. The algorithm requires about 0.1 sec to reschedule 41 machines (on a VAX 11/780) over 1 time step. The random fluctuations in load are modeled by white noise or a (non-stationary) Gaussian-Markov process. In the second case, a Kalman filter is used to compute the one-step ahead prediction of the load.

Performance tests for the algorithm are given for systems of 3, 5, 18 and 41 machines. The optimal costs computed are nearly identical to those computed using dynamic programming [1] or a modified dynamic programming successive approximations algorithm [2]. The tests show that the CPU times required to set up the scheduling table and execute the on-line scheduling algorithm grow slower than linearly in the number of machines in the system.

The design and execution of the algorithm are based on the (implicit) assumptions that the starting costs of the generators are substantially smaller than the operating (fuel) costs. We also assume that the short term fluctuations in the load are a small percentage (approximately 4% or less) of the mean load level. The performance of our algorithm reflects the fact that the sensitivity of the system operating cost to perturbations in the demand and "small" changes in the commitment schedule decrease substantially as the number of machines in the system increases. (This fact was exploited in a different way in [3].) We have used a simple model for the scheduling problem. Enhancements to include run time constraints, more complex (Gaussian) load models, and storage system costs (shadow prices) would not substantially impact the performance of the algorithm. Including transmission costs in the scheduling problem would require redesign of the algorithm.

Almost all previous work on unit commitment scheduling has been based on deterministic load models. Work similar to ours in spirit includes the paper of Bertsekas et al[3] who use a duality formulation of the optimal scheduling problem, the work of Turgeon [4] [5] who uses the maximum principle to treat deterministic model similar to the one used here, the thesis of Leguay [6] who uses "impulse control" methods to treat small scale, deterministic scheduling problems, and the recent work of Gonzalez and Rofman [7] [8] who use a clever combinational algorithm to treat modest sized, deterministic scheduling problems including costs for the storage systems (shadow prices). Mathematical treatments of stochastic unit commitment problems were given in Blankenship and Menaldi [9], Li and Blankenship [10]. Starnes's thesis [2] contains an effective *ad hoc* algorithm for treating (deterministic) scheduling problem with more than one performance measure (e.g. system security and cost). More details on the present work may be found in Yan's thesis [11].

## 2. *SYSTEM MODEL*

We consider a system with M machines (thermal or nuclear units) operating over an interval [0,T] (one day or one week). We assume that the system includes a (composite) energy storage system (pumped hydro). The unit commitment problem is to schedule the startup, operating level, and shutdown of the thermal units and the pumping and withdrawal of energy to and from the storage system to meet the time varying demand for power, L(t), t=0,1,2,...,T, at minimum operating cost. In mathematical terms the problem is

$$\min_{t,i} \sum_{t=1}^{T} \sum_{i=1}^{M} [CG(t,i) + CS(t,i)] \qquad (2.1)$$

subject to the constraints

$$G(t) = \sum_{i=1}^{M} G(t,i) = L(t) + r(t) \qquad (2.2)$$

$$\underline{G}(i) \leq G(t,i) \leq \overline{G}(i)$$

$$\underline{R} \leq r(t) \leq R \qquad (2.3)$$

$$S(t) = \min\{\overline{S}, max\{\underline{S}, S(t-1) + r(t-1)\}\} \qquad (2.4)$$

where G(t,i) is the generation level of the ith machine in time interval t, r(t) is the energy pumped to (r(t)>0) or withdrawn from (r(t)≤0) storage systems in t, S(t) is the total stored energy at t, and the lower and upper bounds $(\underline{G}(i), \bar{G}(i), \underline{R}, ...)$ are physical constraints. The generation cost per unit time is the sum of the running costs, CG(t,i), and starting costs, CS(t,i), of the machines. We assume

$$CG(t,i) = a_1(i) + a_2(i) * G(t,i) + a_3(i) * [G(t,i)]^2 \qquad (2.5)$$

and

$$CS(t,i) = b_1(i) * \{1 - b_2(i) * exp[-b_3(i) * d(t,i)]\} \qquad (2.6)$$

where all the cost coefficients $a_1(i), b_1(i)$, etc., are non-negative and d(t,i) is the cumulative down time of the ith machine at time t.

The *controls* or decision variables for the problem are r(t), G(t,i), t=0,1,...,T, i=1,...,M, and the generation schedule I, an M-vector of 0's and 1's with 1 in the ith position indicating machine i is on, and a zero indicating off. The power set $[(0,1)]^M$ is the collection of all possible schedules. Since continuous controls r(t), G(t,i) are bounded and the schedules I(t) are discrete, the optimization problem involves a non-differentiable objective function.

The *states* of the system are the down time of the machines d(t,i), i=1,2,...,M, and S(t), the stored energy. If a dynamic load model is assumed (see section 6), then the state variables of that model must also be included. If a more elaborate machine model including (minimum) run time constraints or a "banked" state is used, then these states variables must also be added to the state vector. Since our primary concern is to develop an efficient scheduling and dispatch algorithm, we shall not include these features. They do not change the essential structure of the optimization algorithm.

We shall treat the following types of load models : (1) deterministic loads; (2) loads with a pure random fluctuation (white noise) about the mean; and (3) loads with a first order Gaussian-Markov process fluctuation about the mean.

As posed, the problem includes unit commitment (selection of I(t)) and economic dispatch (setting G(t,i) for on-machines and r(t)). The dispatch problem is solved by having all on machines operate at the same incremental cost level

$$\lambda = \frac{d[CG(t,i)]}{d[G(t,i)]} \qquad (2.7)$$

otherwise, by shifting the load of a higher incremental cost unit to machines with lower incremental costs, the overall generation cost could be reduced.

Separating economic dispatch from unit commitment, as was classically done, does not guarantee achievement of the minimum short term scheduling cost.

## 3. SCHEDULING ALGORITHM

Our algorithm is designed in two stages: First, an extensive off-line computation is done to compute a "scheduling table," reminiscient of the priority list in common use in the industry. This table need be computed only once for each system. Using the table, the deterministic unit commitment problem is solved "off-line." Then a simple, efficient "on-line" algorithm is used to fine tune the schedule in response to unexpected (or predicted random) changes in load. In effect the off-line algorithm establishes a rough correspondence between the total generation $G(t) = \sum_{i=1}^{M} G(t,i)$ required to serve the load, and the individual generation assignments, G(t,i), in the "first several cheapest generation schedules" for each given level of demand. The on-line scheduling control is then responsible for modest modifications of the machine schedule (and dispatch) to match the actual load. It does this by selecting the best schedule among the "first several cheapest ones" to achieve the optimal total cost when starting costs are taken into account. The implicit assumption in constructing the table is that the starting costs associated with various schedules are substantially less than the fuel costs.

For each load level, the "cheapest" schedules are selected to achieve the corresponding total generation level, with the individual generation levels assigned to the points where all on machines have the same λ (2.7) in the candidate schedules.

### OFF-LINE ALGORITHM

First, we build up a scheduling table with K candidate cheapest schedules (G(t,i),i=1,...,M) for total generation G(t) to meet different demand levels. Then given the deterministic demand $\bar{L}(t)$, t=0,1,...,T, with bar implying the stochastic mean, we select the initial power transfer r(t), t=0,1,...,T, to make $G(t)=\bar{L}(t) + r(t)$ as flat as possible. Disturbances are added to r(0),...,r(T) to discover the lowest cost. At each iteration the individual generation levels G(t,i) required to achieve G(t) (total mean load at time t) are selected from the scheduling table. Minor adjustment and compensation of r(0),...,r(T) are made to match physical constraints and demand and achieve minimum cost. This limits the computational burden when dealing with a large number of machines. The final r(t), G(t,i) and G(t) are the mean transfer rates, individual generations and total generation, denoted by $\bar{r}(t), \bar{G}(t,i)$ and $\bar{G}(t)$, t=1,..., T, i=1,...,M.

### ON-LINE ALGORITHM

When the load is a random process, on-line scheduling is required to compensate the (planned) deterministic generation and pumping schedule. We regard the stochastic demand L(t),t=0,1,...,T, as a random fluctuation about the mean demand $\bar{L}(t)$. The power transfer $r(t) = \bar{r}(t) + \delta r(t)$ is adjusted to make $G(t) = \bar{G}(t) + \delta G(t)$ as flat as possible for t=0,1,...,T. By using $\bar{r}(t)$, t=0,1,...,T, as the starting point for the on-line iteration and searching the scheduling table for close feasible solutions, the optimal schedule for the actual current load may be found very rapidly. If the demand fluctuation process can be modeled by Gaussian-Markov process, the Kalman state estimator provides one-step ahead prediction of the demand. This permits a better (lower cost) control since we can smooth out the fluctuations between two time intervals (present and next time steps) by using the storage system and power transfers.

The advantage of this algorithm over the conventional priority list is the consideration of the operating status of "all" the machines in response to a change in demand. The priority list indicates the next machine to be turned on or off in response to an increase or decrease in demand. However, in some cases the optimal response to an increase in demand is to turn some machines on while turning others off. The reverse can happen when demand decreases.

The overall algorithm is summarized in Fig 1.

## 4. CONSTRUCTION OF THE SCHEDULING TABLE

A typical scheduling table is shown in Figure 2. The total generation levels G(t) in increments, $\delta G(j)$ which can vary in size, are listed along the left most column. The individual generations G(t,i) required to achieve those levels (including economic dispatch) are listed in the rows of the table, commencing with the cheapest schedule (ignoring starting costs) and continuing to the Kth cheapest schedule. Note for this case, the cheapest schedule to serve level 7420 has machine G1 down and G2 up; the reverse of the solution at level 7400. The number, K, of candidate schedules for each level of demand is chosen in one of two ways:

(1) If $t[j,k,M+1]$ is the cost entry for the (j,k) row schedule of the table (cost in the right most column), K is chosen in such a way that $t[j,K+1,M+1]-t[j,K,M+1]$ is larger than the

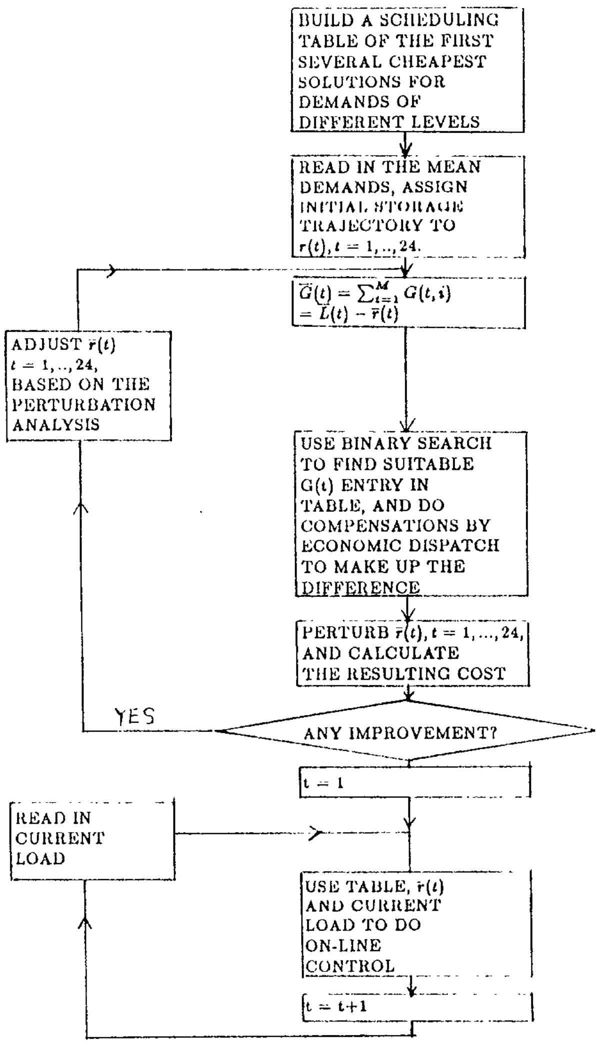



Figure 1. DETERMINISTIC SCHEDULING ALGORITHM

most expensive starting cost of machines which are allowed to switch.

(2) Alternately, empirical evidence from simulation or operation experience may indicate that among the first K cheapest schedules, only the first 2 or 3 are ever applied. Then K may be safely reduced to 4 or 5. This enhances the on-line speed of the algorithm.

The increment of the total generation, $\delta G(j)$, is chosen by the rule

$$\delta G(j) = \min(\delta G, X[G]) = t[j+1,1,0] - t[j,1,0] \quad (4.1)$$

where $\delta G$ is a nominal increment (100MW in our examples) and X[G] is the smallest increment of generation such that G+X[G] has a different optimal unit commitment (I(t,i)) from that at level G. In the example in Figure 2, with $\delta G = 100$, the choices are

$$\min(100, X[7300]) = 100$$

$$\min(100, X[7400]) = 20 \quad (4.2)$$

since

for j=1 schedule (20,0,40,40,...,600) -> commitment (1,0,1,1,...,1)
for j=2 schedule (30,0,30,40,...,600) -> commitment (1,0,1,1,...,1)
for j=3 schedule (0,50,60,60,...,600) -> commitment (0,1,1,1,...,1)

The power transfer rates r(t), t=0,1,...,T, are chosen to assure:

(i) the stored energy is periodic S(t)=S(t+T) which implies:

$$\sum_{t=1}^{T} r(t) = 0 \quad (4.3)$$

(ii) the total generation G(t) is as flat as possible; and

(iii) the constraints $\underline{R} \le r(t) \le \overline{R}$ hold.

The key condition (ii) is a consequence of the quadratic form of the generation cost functions CG(t,i).

| G | G1 | G2 | G3 | G4 | ..... | GM | COST | PRIORITY |
|---|---|---|---|---|---|---|---|---|
| 7300 | 22 | 0 | 44 | 41 | ..... | 600 | 18000 | first |
| | 0 | 29 | 47 | 41 | ..... | 600 | 18120 | second |
| | 30 | 36 | 0 | 45 | ..... | 600 | 18439 | 3rd |
| (j=1) | . | . | . | . | . | . | . | |
| | . | . | . | . | . | . | . | |
| | . | . | . | . | . | . | . | |
| | 30 | 11 | 0 | 67 | ..... | 600 | 18700 | Kth |
| 7400 | 34 | 0 | 32 | 49 | ..... | 600 | 18900 | |
| | 31 | 0 | 0 | 87 | ..... | 600 | 19240 | |
| (j=2) | . | . | . | . | . | | | |
| | . | . | . | . | . | . | | |
| | 32 | 46 | 48 | 42 | ..... | 600 | 22320 | |
| 7420 | 0 | 53 | 62 | 67 | ..... | 600 | 24440 | |
| | . | . . | . | . | . | . | | |
| (j=3) | . | . | . | . | . | | | |
| | . | . | . | . | . | . | | |
| | . | . | . | . | . | . | | |
| 15000 | 48 | 42 | 59 | 50 | ..... | 600 | 65229 | first |
| (j=302) | . | . | . | . | | | | |

Figure 2. A Typical Scheduling Table.

## 5. *DETERMINISTIC SCHEDULING*

Several test problems with a deterministic demand for power were treated with the (off-line) algorithm to establish a base line for the (on-line) stochastic scheduling algorithm. Operating and starting cost data for the two larger examples, 18 and 41 machines, are listed in Tables 3 and 4 in the Appendix. These systems have been treated earlier using a modified dynamic programming successive approximations algorithm DPSA [2], and the performance results were used as a check on the current algorithm. Smaller examples involving 3 and 5 machines which can be treated by dynamic programming were also used to validate the algorithm. The overall cost figures obtained in these tests are shown in Table 1. The differences in costs in the smaller examples are primarily due to the propagation of quantization errors in the dynamic programming algorithm. The cost differences in the two larger examples are primarily due to quantization effects and the inherent inaccuracy of DPSA.

| NO. OF MACHINES | DPSA OR DP | PRESENT ALGORITHM |
|---|---|---|
| 03 | 0022288.77 | 0022232.448 |
| 03 | 0030210.63 | 0030135.504 |
| 03 | 0022283.14 | 0022251.835 |
| 05 | 0038115.60 | 0038108.249 |
| 18 | 0461047.00 | 0459235.710 |
| 41 | 1623799. 00 | 1584421.308 |

Table 1. OPTIMAL COST COMPARISON

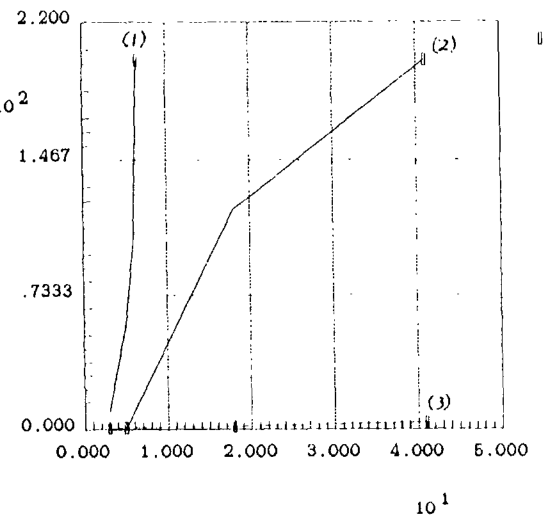
Execution times for the algorithm in this paper are shown in Table 2, including the time required to construct the scheduling table. One day refers to an interval of 24 time steps. The program is written in PASCAL and the times are obtained on a VAX

11/780 operating under Berkeley Unix version 4.1.

| NO. OF MACHINES | BUILD UP SCHEDULING TABLE | ITERATION TO OBTAIN 24 HOUR OPTIMAL SCHEDULE |
|---|---|---|
| 03 | 000:30 | 0:42 |
| 05 | 000:50 | 1:10 |
| 18 | 120:00 | 1:35 |
| 41 | 200:00 | 3:50 |

CPU TIME(MIN:SEC) (OFF-LINE)
Table 2. EXECUTION TIME FOR DETERMINISTIC CASES

The trend in computational times versus number of machines is important. Figure 3 shows that both the time required to build the scheduling table and the unit iteration time for the deterministic algorithm have less than linear growth as functions of machine numbers. The search procedure used to construct the scheduling table was not optimized, and the computational times for this procedure could be substantially reduced if effective search procedures were used.




curve (1) for DPSA
curve (2) for building table
curve (3) for deterministic load iteration
CPU time is in 100 minutes.
Figure 3. CPU TIME VS NUMBER OF MACHINES

## 6. *ON – LINE STOCHASTIC SCHEDULING*

The off-line algorithm determines a schedule and a set of optimal transfer rates $\bar{r}(1), ..., \bar{r}(T)$ corresponding to a known, deterministic demand $\bar{L}(t), t = 0, 1, ..., T$. Starting with $\bar{r}(1), ..., \bar{r}(T)$ and $\bar{G}(t,i), i = 1, 2, ..., M$, the on-line algorithm may be used to determine the best schedule (including starting costs), G(t,i), i=1,...,M., and optimal transfer rates r(1),...,r(T) to meet the actual, observed (random) demand L(t), t=0,1,...,T. The algorithm is fast enough to be used on-line.

In testing the algorithm we used two different models for the demand. The first was simply

$$L(t) = \bar{L}(t) + \delta L(t) \qquad (6.1)$$

with $\delta L(t)$ a sequence of independent zero mean random variables uniformly distributed in the interval $[-\alpha\bar{L}(t), \alpha\bar{L}(t)]$ with $\alpha$ a small number. In the tests we used $\alpha = 0.04$, so the demand fluctuations were less than or equal to 4% of the mean at all time steps.

In the second demand model, we assumed that the demand fluctuations were a first order Gaussian-Markov process

$$\delta L(t+1) = a * \delta L(t) + w(t) \qquad (6.2)$$

with 'a' a real constant and $w(t), t = 0, 1, ....$ a sequence of zero mean, independent, identically distributed Gaussian-Markov process(i.e. a Gaussian white noise process). We used the model

$$y(t) = \delta L(t) + v(t) \qquad (6.3)$$

with $v(t)$ Gaussian white noise process to describe the measurements of the demand fluctuations. We assume

$$E[v(t)w(s)] = 0 \qquad \forall\ t, s$$

$$E[v(t)\delta L(0)] = 0 = E[w(s)\delta L(0)] \quad \forall\ t, s \qquad (6.4)$$

$$E[w(t)w(s)] = W(t) * \delta_{st}$$

$$E[v(t)v(s)] = V(t) * \delta_{st}$$

$$E[\delta L(0)] = 0, \quad E[(\delta L(t))^2] = \sigma(t)$$

$$\delta_{st} = \begin{cases} 0, & s \neq t \\ 1, & s = t \end{cases} \qquad (6.5)$$

(The parameters a,$w(t)$,$v(t)$ and $\sigma(t)$ must be identified from actual load data.)

The Kalman filter is the best estimator of $\delta L(t)$ given $y(s)$, $s \leq t-1$. The equations are

$$\delta L(t+1|t) = a * \delta L(t|t-1) + K(t) * [y(t) - \delta L(t|t-)] \qquad (6.6)$$

where $\delta L(t|t-1)$ is the best (linear) estimate of $\delta L(t)$ given $y(s), s \leq t-1$, and

$$K(t) = \frac{a * P(t)}{P(t) + V(t)}$$

$$P(t+1) = \frac{a * P(t) \quad a * P(t)^2}{P(t) + V(t) + W(t)} \qquad (6.7)$$

$$P(0) = \sigma(0)$$

Both K(t) and P(t) can be computed off-line.

The on-line scheduling algorithm based on the Gaussian-Markov load model is shown in Figure 4. the same algorithm is used for the random fluctuations model, with $\delta L(t+1|t)$ in that Gaussian-Markov model substituted by $\delta L(t)$ in the Figure.

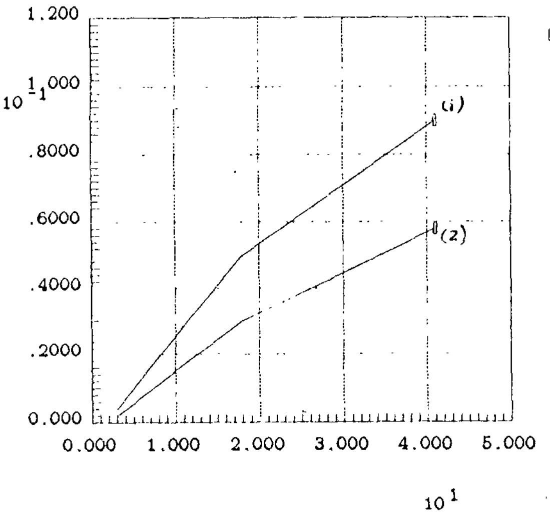
The overall scheduling algorithm is shown in Figure 5.

## 7. *ON – LINE SCHEDULING FOR 41 MACHINES*

In conducting the schedule computations 80 days of (synthetic) hourly demand were used to generate the random load statistics. A break down of the computational times for the algorithm in the two model cases is given in Figure 6. A plot of the computational times required to find the optimal hourly schedule for the two different random demand models versus number of machines is given in Figure 7. Note that the rate of increase is less than linear. The times are longer when the algorithm includes the Kalman predictor, since time is required to execute the additional lines of code. (For the 41 machine case 87 seconds were required to find the optimal schedule for 40 days in 1 hour increments.)

| TYPE OF EXECUTION | RANDOM DISTURBANCE | GAUSSIAN MARKOV DISTURBANCE |
|---|---|---|
| READ IN TABLE | 37.0(off-line) | 30.0(off-line) |
| DETERMINISTIC ONE HOUR SCHEDULE | 0.6667(off -line) | 0.6667(off-line) |
| STOCHASTIC ON-LINE ONE HOUR CONTROL | 0.06063(on-line) | 0.090625(on-line) |

Figure 6. 41 MACHINES SCHEDULING TIMES




curve (1) for Gaussian-Markov disturbance
curve (2) for random disturbance
CPU time is in 0.01 second.

Figure 7. ON-LINE SCHEDULING TIME VS NUMBER OF MACHINES

## 8. *CONCLUSIONS*

The proposed algorithm provides a fast and effective procedure for scheduling unit commitment and randomly varying demand. The speed of the algorithm is a result of the extensive off-line computation done to construct the scheduling table. Since this need only be done once for a given system, the computational expense of this operation is not excessive. The computational requirements of the on-line component of the algorithm are approximately linear in the number of machines - a dramatic improvement over DPSA. By taking advantage of the fact that actual loads different only by a small percentage from the mean expected load for a given day, an extremely accurate initial schedule for generation and storage can be computed (off-line). The (optimal) on-line schedule is essentially a perturbation of the (mean) deterministic schedule, and, as the number of machines increases, the sensitivity of the cost to small perturbations in the demand or in the corresponding schedule is substantially reduced.

The speed of the scheduling algorithm means that it can be used on-line to adjust generation schedules and fine tune dispatch to minimize cost.

*ACKNOWLEDGEMENT*: This research was supported in part by the Division of Electric Energy Systems of the Department of Energy under contract DE-AC01-79ET29244.

## *REFERENCE*


1. F. S. Kuhl, "Unit Commitment and Storage Scheduling via Dynamic Programming," MS Thesis, Electrical Engineering Department, University of Maryland, College Park, 1980.
2. R. A. Starnes, "Multiobjective Optimization in Power System Unit Commitment," MS Thesis, Electrical Engineering Department, University of Maryland, College Park, 1980.
3. D. P. Bertsekas, G. S. Lauer, N. R. Sandell, and T.A. Posbergh, "Optimal Short-Term Scheduling of Large-Scale Power Systems," IEEE Trans. Power Apparatus and Systems, Jan. 1983.
4. A. Turgeon, "Optimal Unit Commitment," IEEE Trans. Automatic Control, Vol. AC-21, PP223-227, April 1977.
5. A. Turgeon, "Optimal Scheduling of Thermal Generaing Units," IEEE Trans. Automatic Control, Vol. AC-23, PP 1000-1005, Dec. 1978.
6. C. Legauy, "Application du Controle Stochastic a un Probleme de Gestion Optimale D'energie," These de Docteur-Ingenieur, Universite de Paris $\overline{IX}$, 1975.
7. R. Gonzalez and E. Rofman, "On Deterministic Control Problems: an Approximation Procedure For The Optimal Cost," SIAM J. Control and Optimization, to appear.
8. E. Rofman, "Approximation to Hamalton-Jacobi-Bellman Equation in Deterministic Control Theory: An Application to Energy Production Systems," in Recent Mathematical Methods in Dynamic Programming, I. Capuzzo Dolcetta, W. H. Fleming, and T. Zolezzi, eds., Lecture Notes in Mathematics, Vol.1119, Springer Verlag, New York, 1984, PP.152-189.
9. G. L. Blankenship and J.L. Menaldi, "Optimal Stochastic Scheduling of Power Systems With Delays and Large Cost Differentials," SIAM J. Control, 22(1984), pp.121-132.
10. C. W. Li and G. L. Blankenship, "Optimal Stochastic Scheduling of Systems with jump process coefficients," submitted for publication.
11. I. Yan, "An Algorithm For Stichastic Scheduling of Electric Power System," MS Thesis, Electrical Engineering Department, University of Maryland, College Park, 1983.


## *APPENDIX*

*Table* 3. COST COEFFICIENTS FOR 18 MACHINES

GENERATION COST DATA

| M | $\underline{G}$ | $\overline{G}$ | a3[i] | a2[i] | a1[i] |
|---|---|---|---|---|---|
| 1 | 15 | 045 | 0.027039 | 7.4840 | 0025 |
| 2 | 15 | 045 | 0.046586 | 7.1030 | 0044 |
| 3 | 15 | 079 | 0.041011 | 7.2710 | 39.6 |
| 4 | 20 | 053 | 0.023020 | 6.8356 | 42.0 |
| 5 | 20 | 053 | 0.023020 | 6.8356 | 42.0 |
| 6 | 30 | 130 | 0.011685 | 5.0250 | 94.3 |
| 7 | 30 | 105 | 0.012176 | 5.4530 | 68.6 |
| 8 | 30 | 105 | 0.012167 | 5.4530 | 68.6 |
| 9 | 30 | 105 | 0.121670 | 5.4530 | 68.6 |
| 10 | 30 | 105 | 0.012167 | 5.4530 | 68.6 |
| 11 | 35 | 167 | 0.005206 | 5.6787 | 89.0 |
| 12 | 35 | 167 | 0.005206 | 5.6787 | 89.0 |
| 13 | 35 | 167 | 0.005206 | 5.6787 | 89.0 |
| 14 | 35 | 167 | 0.005206 | 5.6787 | 89.0 |
| 15 | 25 | 388 | 0.002269 | 5.8581 | 68.0 |
| 16 | 00 | 425 | 0.000769 | 4.6860 | 28.2 |
| 17 | 50 | 670 | 0.000827 | 6.4870 | 60.4 |
| 18 | 50 | 670 | 0.000827 | 6.4870 | 60.4 |

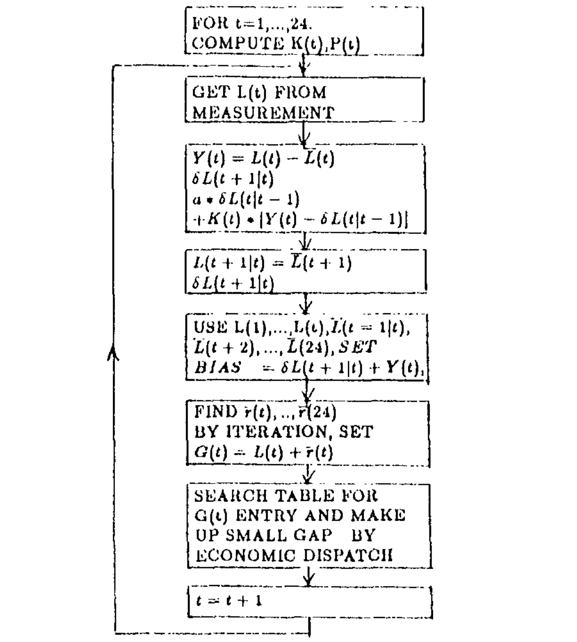
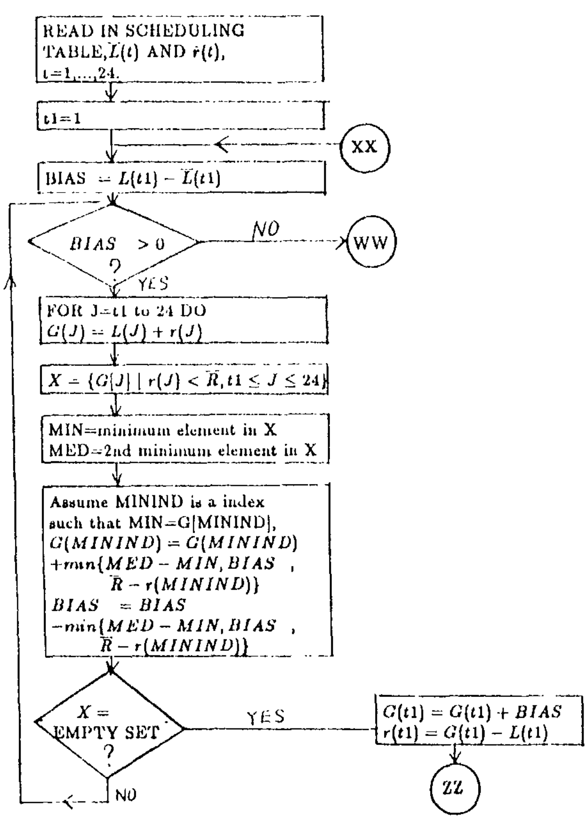



Figure 4. GAUSSIAN MARKOV MODEL SCHEDULING



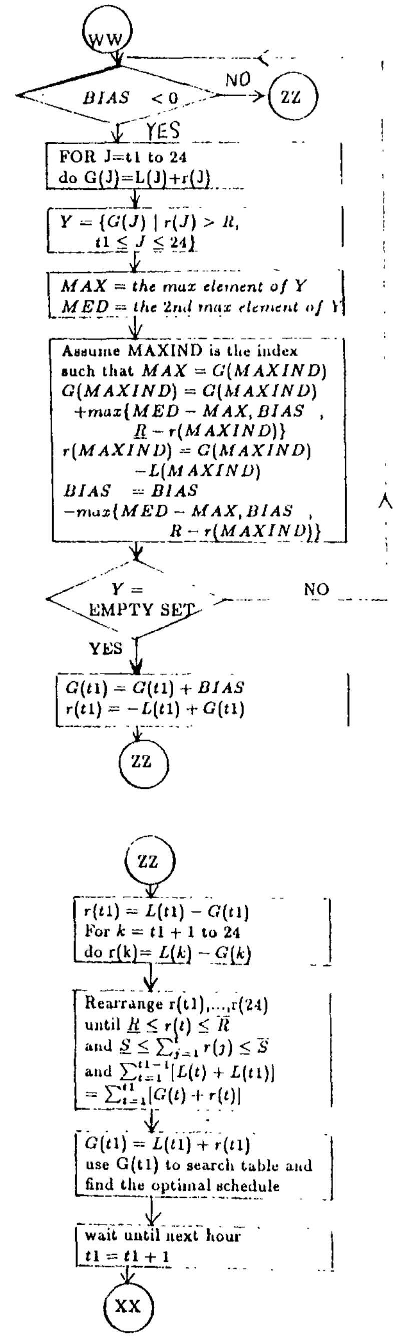




Figure 5. OVERALL ON-LINE ALGORITHM

The algorithm is very effective in smoothing variations in the demand. In a test where the standard deviation of demand fluctuations was 1.42% the average increase of the unit generation cost (defined as the total operating and starting costs divided by total generation in MW) over the deterministic cost was 0.058% for the random disturbance load model and 0.031% for the Gaussian-Markov noise model.

STARTING COST AS A FUNCTION OF DOWN-TIME

| M | CS(1) | CS(2) | CS(3) | CS(4) | CS(5) | CS(6) |
|---|---|---|---|---|---|---|
| 01 | 083 | 163 | 242 | 318 | 392 | 464 |
| 02 | 083 | 164 | 242 | 318 | 292 | 464 |
| 03 | 065 | 149 | 220 | 290 | 357 | 424 |
| 04 | 040 | 080 | 114 | 149 | 182 | 214 |
| 05 | 040 | 080 | 114 | 149 | 182 | 214 |
| 06 | 133 | 258 | 376 | 486 | 590 | 688 |
| 07 | 181 | 330 | 454 | 555 | 639 | 709 |
| 08 | 039 | 077 | 114 | 150 | 185 | 220 |
| 09 | 039 | 077 | 114 | 150 | 185 | 220 |
| 10 | 055 | 069 | 102 | 135 | 167 | 299 |
| 11 | 156 | 309 | 457 | 602 | 743 | 881 |
| 12 | 156 | 309 | 457 | 602 | 743 | 881 |
| 13 | 156 | 309 | 457 | 602 | 743 | 881 |
| 14 | 156 | 309 | 457 | 602 | 743 | 881 |
| 15 | 9998 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 16' | 9998 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 17 | 9998 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 18 | 9998 | 9999 | 9999 | 9999 | 9999 | 9999 |

*Table 4.* COST COEFFICIENTS FOR 41 MACHINES

GENERATION COST DATA

| M | $\underline{G}$ | $\bar{G}$ | a3[i] | a2[i] | a1[i] |
|---|---|---|---|---|---|
| 1 | 3.5e+01 | 1.7e+02 | 0.00521 | 5.67900 | 89.00000 |
| 2 | 3.5e+01 | 1.7e+02 | 0.00521 | 5.67900 | 89.00000 |
| 3 | 3.5e+01 | 1.7e+02 | 0.00521 | 5.67900 | 89.00000 |
| 4 | 3.5e+01 | 1.7e+02 | 0.00521 | 5.67900 | 89.00000 |
| 5 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 6 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 7 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 8 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 9 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 10 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 11 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 12 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 13 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 14 | 2.5e+01 | 1.0e+02 | 0.00458 | 6.59200 | 35.00000 |
| 15 | 1.3e+02 | 3.9e+02 | 0.00227 | 5.85800 | 162.00000 |
| 16 | 1.3e+02 | 3.9e+02 | 0.00227 | 5.85800 | 162.00000 |
| 17 | 2.5e+02 | 5.5e+02 | 0.00197 | 6.41700 | 275.00000 |
| 18 | 2.5e+02 | 5.5e+02 | 0.00197 | 6.41700 | 275.00000 |
| 19 | 6.5e+01 | 2.5e+02 | 0.00405 | 6.35400 | 250.00000 |
| 20 | 6.5e+01 | 2.5e+02 | 0.00405 | 6.35400 | 250.00000 |
| 21 | 6.5e+01 | 2.5e+02 | 0.00405 | 6.35400 | 250.00000 |
| 22 | 6.5e+01 | 2.5e+02 | 0.00405 | 6.35400 | 250.00000 |
| 23 | 5.0e+01 | 2.0e+02 | 0.00354 | 6.14200 | 60.000000 |
| 24 | 5.0e+01 | 2.0e+02 | 0.00354 | 6.14200 | 60.000000 |
| 25 | 5.0e+01 | 2.0e+02 | 0.00354 | 6.14200 | 60.000000 |
| 26 | 2.5e+02 | 6.7e+02 | 0.00083 | 6.48700 | 360.00000 |
| 27 | 2.5e+02 | 6.7e+02 | 0.00083 | 6.48700 | 360.00000 |
| 28 | 2.0e+02 | 5.0e+02 | 0.00064 | 7.07600 | 185.00000 |
| 29 | 2.0e+02 | 5.0e+02 | 0.00064 | 7.07600 | 185.00000 |
| 30 | 2.0e+02 | 5.0e+02 | 0.00064 | 7.07600 | 185.00000 |
| 31 | 1.2e+02 | 3.5e+02 | 0.00081 | 6.89900 | 149.00000 |
| 32 | 1.2e+02 | 3.5e+02 | 0.00081 | 6.89900 | 149.00000 |
| 33 | 3.2e+02 | 7.5e+02 | 0.00061 | 6.70400 | 410.00000 |
| 34 | 2.0e+02 | 4.3e+02 | 0.00077 | 6.68600 | 128.00000 |
| 35 | 2.0e+02 | 4.3e+02 | 0.00077 | 6.68600 | 128.00000 |
| 36 | 2.0e+02 | 4.3e+02 | 0.00077 | 6.68600 | 128.00000 |
| 37 | 4.0e+02 | 8.0e+02 | 0.00095 | 4.55100 | 390.00000 |
| 38 | 4.0e+02 | 8.0e+02 | 0.00095 | 4.55100 | 390.00000 |
| 39 | 4.0e+02 | 8.0e+02 | 0.00095 | 4.55100 | 390.00000 |
| 40 | 4.0e+02 | 8.0e+02 | 0.00095 | 4.55100 | 390.00000 |
| 41 | 6.0e+02 | 1.2e+03 | 0.00063 | 3.95100 | 586.00000 |

STARTING COST AS A FUNCTION OF DOWN-TIME

| M | CS(1) | CS(2) | CS(3) | CS(4) | CS(5) | CS(6) |
|---|---|---|---|---|---|---|
| 1. | 156.39 | 308.82 | 457.39 | 602.21 | 743.35 | 880.93 |
| 2. | 56.39 | 308.82 | 457.39 | 602.21 | 743.35 | 880.93 |
| 3. | 156.39 | 308.82 | 457.39 | 602.21 | 743.35 | 880.93 |
| 4. | 156.39 | 308.82 | 457.39 | 602.21 | 743.35 | 880.93 |
| 5. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 6. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 7. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 8. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 9. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 10. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 11. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 12. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 13. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 14. | 180.96 | 330.34 | 453.66 | 555.45 | 639.48 | 708.85 |
| 15. | 35.710 | 70.850 | 105.42 | 139.44 | 172.92 | 205.86 |
| 16. | 35.710 | 70.850 | 105.42 | 139.44 | 172.92 | 205.86 |
| 17. | 250.00 | 300.00 | 390.00 | 550.00 | 660.00 | 770.00 |
| 18. | 250.0 | 300.00 | 390.00 | 550.00 | 660.00 | 770.00 |
| 19. | 156.3 | 308.82 | 457.39 | 602.21 | 743.35 | 880.93 |
| 20. | 156.3 | 308.82 | 457.39 | 602.21 | 743.35 | 880.93 |
| 21. | 156.3 | 308.82 | 457.39 | 602.21 | 743.35 | 880.93 |
| 22. | 156.3 | 308.82 | 457.39 | 602.21 | 743.35 | 880.93 |
| 23. | 113.2 | 221.64 | 325.27 | 424.38 | 519.18 | 609.85 |
| 24. | 113.29 | 221.64 | 325.27 | 424.38 | 519.18 | 609.85 |
| 25. | 113.29 | 221.64 | 325.27 | 424.38 | 519.18 | 609.85 |
| 26. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 27. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 28. | 6320.57 | 8645 | 9501 | 9815 | 9931 | 9974 |
| 29. | 6320.57 | 8645 | 9501 | 9815 | 9931 | 9974 |
| 30. | 6320.57 | 8645 | 9501 | 9815 | 9931 | 9974 |
| 31. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 32. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 33. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 34. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 35. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 36. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 37. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 38. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 39. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 40. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 41. | 9997.77 | 9999 | 9999 | 9999 | 9999 | 9999 |