UNIT COMMITMENT USING CONSTRAINED LAMBDA DISPATCH WITH THE IBM:PC

by

Bradley Dean Eckhoff

B.S., Kansas State University, 1983

A MASTER'S THESIS

submitted in partial fulfillment of the requirements

for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1985

Approved:

Major Professor

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

1.0 Literature Overview and Discussion of Technique

"The problem of economic dispatch had its start from the time that two or more units were committed to take on load on a power system whose total capacities exceeded the generation required. The problem that confronted the operator was exactly how to divide up the real load between the two units."

As indicated in this quotation from H. H. Happ's paper - entitled "Optimal Power Dispatch - A Comprehensive Survey"[1] the area of optimal economic unit dispatch and optimal fuel use for electric utilities extends back into the early twentieth century. The conceptual study of selecting the most efficient or economical unit to use from the many units available has resulted in several different techniques ranging from the base load method to the equal incremental dispatch method (λ). In order to present an overview of work that has been performed in the past a brief discussion will be presented in addition to an outline of some of these techniques. The dates of the articles in which these techniques are presented will be given in order to understand the sequential ordering of technique development.

1.1 Base Load - Best Point Loading

Before the 1930s several different economic dispatch methods were used. The first of these was the base load method, in which the unit with the lowest generation cost is used until its maximum capabilities have been exhausted - then the one with the next lowest generation cost is used, etc. The other method was called the best point loading method. This is a method in which the units are loaded to their lowest heat rate point. The heat rate value is determined by dividing the amount of energy produced by a unit by the amount of energy inputted into the unit. The heat rate is often given in units of Btu/kWh. Thermal efficiency is found by dividing 3412 Btu/kWh by the heat rate. A unit's lowest heat rate is the point at which it is functioning in the most energy efficient manner, thus the reason for its desirability.

1.2 Equal Incremental Cost Dispatch (Lambda Dispatch)

By the 1930's the equal incremental loading method was developed. The idea of using the unit or units with the least incremental costs was recognized as yielding the most economical results and is still in widespread use today.

The lambda dispatch principle is demonstrated through relatively simple steps involving differential calculus, power supply and demand constraints, an objective function expressed as cost (to be minimized), and Lagrangian multipliers [the lambda (λ) dispatch method]. The theoretical development is presented in Chapter 2.

1.3 Power Loss Technique Development

The lambda dispatch method proves that equal incremental costs (λ) are a desirable goal. However, it does so with an unrealistic assumption that there are no transmission line losses in moving power from its generation point to its use point. The non-realistic nature of this assumption was realized almost from the outset of lambda dispatch studies. By 1943 Steinberg and Smith[2] advanced the studies in the loss area by developing a penalty factor form very similar to that used today. In 1943 E. E. George[3] extended the work by using source loadings to express total transmission losses. This in turn was simplified in

1945 and extended in 1950 by Ward, Eaton, and Hale[4] using two basic assumptions. The first was that the amount of power produced remains as a constant despite the fluctuations inherent with load apportionment. The second was that as the total system load varies the individual load current varies in direct proportion. Results of other work were published by George, Page and Ward[5] to reduce the time for computations. The idea was to use a linear programming (LP) method and to combine the transmission loss formula with total and incremental fuel costs in preparing loading schedules for a large system.

1.4 Extended Study - Power Loss and λ Dispatch

Other work included solving for λ using simultaneous solutions of power generator equations, as shown in studies done by Travers, Hacker, Long and Harder in 1954[6] and work by Kirchmayer and Stagg[7] to reduce a loss theory developed earlier by Kron[8] to a simpler form. Their studies not only resulted in a much improved loss formula calculation procedure, but also improved an idea brought forth by Ward[9] (and used currently), which is called the classic coordination equations.

A simplified look at the penalty factor development stems from the aforementioned proof that the value of λ is the minimum operating cost. The difference lies in the power demand constraint equation. The total power demanded equals the sum of the differences of the power supplied and the power loss of all the units involved. The theoretical development will be shown in Chapter 2.

1.5 Dynamic Programming

Other studies stressing different techniques have also been evident in the research world. Dynamic programming (DP), as presented by

Lowery[10] in 1966, was seen as a viable technique by some to solve the generating unit commitment problem. In fact, despite the critique of others of the limitations of DP, current work indicates that DP may be a viable solution technique. Such work was presented by H. F. Van Meeteren in July 1984.[11] This required combining DP with LP to arrive at their conclusions.

Van Meeteren[11] used the total fuel cost objective function, i.e., the fuel price of a fuel type and its associated heat rate for every fuel of every segment of the input-output curve of all the units composing all the plants. This function was subjected to several different criteria from which the initial LP solution was found. Subsequently, the fuel allocation (how much and where) was defined by LP optimization. Unit commitment was defined for the given fuel allocation. This was then considered with the minimum limited fuel unit commitment to generate a better LP model. This results in determining the initial LP solution. If this initial LP solution was not adequate, i.e., within prespecified ranges, the process was repeated.

LP techniques have been studied with work presented by Megehed, Taleb, Iskanrdani and Moussa in January of 1977.[14] Their work involved taking the non-linear solution approach and breaking it down into several smaller LP problems. This involved linearizing the objective function and constraints and using the simplex method for the optimal solution.

1.6 Newton's Method

A general solution based on Newton's method was presented by Dommel and Tinney in 1968.[12] This process accounted for dependent constraints by using the minimum costs and penalty functions that were obtained from the gradient adjustment algorithm. This technique appeared to be more in line in working towards the ultimate goal of using one global criterion instead of using several local criteria as was most generally done.

Dommel and Tinney[12] recognized in their analysis that there are two cases which should be treated. First, for optimal real and reactive power flow, where the objective function equals the instantaneous operating costs, the solution equals the exact optimal dispatch. Second, the optimal reactive power flow objective function equals the total system losses, thus the solution equals minimum power losses. The theoretical development for both of these cases will be presented in Chapter 2.

In 1973 Alsac and Stott[13] extended the Dommel-Tinney[12] approach by including exact outage - contingency constraints in the Dommel-Tinney method. This gave an optimal steady-state-secure system operating point.

1.7 Quadratic Programming (QP)

Quadratic programming has been presented as an adequate technique for smaller systems. Studies provided by Nicholson and Sterling in 1972[15] and Reid and Hasdorff[16] both show that linear programming techniques are a necessity with quadratic programming. However, Nicholson and Sterling[15] used Langrangian multipliers which were extended to include the Kuhn-Tucker optimality conditions, much like Dommel and Tinney.[12] Reid and Hasdorff[16] used a method referred to as Wolfes method which, as presented, assures a global minimum.

1.8 Other Techniques

Still other techniques, which have appeared as solutions to this problem are the Fletcher-Powell non-linear technique by Sasson and Merrill in 1974,[17] work in 1978 by R. Lugto[18] using a a simple procedure coupled with the differential algorithm in order to optimize generation schedules while using machine limitations, transmission considerations and system reserve requirements as constraints. Further studies resulted in the hierarchical system theory approach developed by Arafeh and Sage in 1979,[19] a system designed for larger and more complex systems, and security constraints. Also a technique incorporating the use of standard load and applicable fuel constraints was presented by Trefny and Lee in 1981,[20] and a method based on the cartesian coordinate formulation of the problem with the reclassification of state and control variables associated with generator buses developed by Roy and Rao was published in 1983.[21] Another method presented as the branch-and-bound technique, provided by Cohen and Yoshimura in 1983,[22] requires no priority unit ordering and incorporates the time-dependent start-up costs, demand and reserve constraints and minimum up and down time constraints. Further explanation of these two processes will be given in Chapter 2.

The dates of the research mentioned above indicate that, to date, research in this area is alive and well. One of the latest developed processes is an economic dispatch computer program (EDP) developed by EPRI. This program has many options in developing the optimal loading scheme for power production. A few of these are: use of base load priority listings, use of variable constraints, and use of loss coefficients and B-matrices.

This particular method can presently be considered the state of the art. The reason for this conclusion comes from the combined efforts (three volumes) of Vemuri, Kumar, Hackett, Eisenhsuer, and Lugtu.[23] In Part I of the Fuel Resource Scheduling (FRS) series they mention FRS as a hierarchical scheduling scheme in an Energy Management System. However, they go on to mention the EPRI work as work that is being done presently.

Part two continues in this fashion by mentioning the network flow algorithm that is critical in using economic dispatch in determining the units to use (or not to use). A quotation taken from the introduction of this paper [Fuel Scheduling - Part II, July 1984[24]] is:

"The economic dispatch of the total generation requirement of a power system is usually accomplished by loading each generating unit to the same incremental cost level unless otherwise constrained...."

thus, reemphasizing the desirable characteristic of incremental cost.

This article is subsequently followed by Part III which deals specifically with the short term (day-to-day) approach. This paper states how an iterative procedure is used to correct any mismatches between system MW requirements and MBtu consumption after the linear fuel constraints are decoupled from the discrete unit commitment/ decommitment decisions.

Chapters two and three are dedicated to further theoretical development of a selected number of the techniques presented thus far.

2.0 Technique Theory Development

This chapter is divided into three areas: 1) Further theoretical development and discussion are presented of some of the more important work as presented in Chapter 1. 2) Development and discussion of the three λ dispatch equations used in this work, i.e., known power, known incremental costs, and known system demand and set of candidate units are presented. 3) Development and discussion concerning the plot of the incremental cost (λ) versus the total system power demand and the independent power production level of each unit are given.

2.1 Theoretical Development:

Selected methods of those presented in Chapter 1 will be presented in more detail in this section. These methods consist of lambda dispatch (transmission loss and no-transmission loss cases), the branch-and-bound method, dynamic programming, and dynamic programming with linear programming, cartesian coordinate formulation, and load flow analysis with consideration of transmission loss. While other important methods do exist the theoretical development and discussion of these specific methods will give the most comprehensive and up-to-date knowledge of what is being done in the area of unit commitment.

2.2 Lambda Dispatch

Lambda dispatch as the optimal solution for selection of the settings of generators for a given demand was recognized in the late 1920's and early 1930's.[25] The theory of lambda dispatch is most easily explained by using a system which has no transmission losses, i.e.,

$$\text{Power loss} = P^L = 0 \text{ MW} \tag{2.1}$$

Power demand is met exactly by the power supplied from all the units used in the system.

$$\text{Power demand} = PD = \sum_{i=1}^{N} P_i^s = \text{Power supplied by N units} \tag{2.2}$$

A cost function for each unit must be developed. This function depends upon heat rate (the input/output function for the unit), the power setting for each, and the average input fuel cost. For most cases, as is the case here, the cost function is described as quadratic, i.e.,

$$f_i = (\alpha_i + \beta_i P_i^s + \gamma_i P_i^{s2})\ c_i \quad i = 1, 2, \ldots, N, \tag{2.3}$$

where α_i, β_i, γ_i are input/output function coefficients, c_i is the average fuel cost for unit i.

By summing the cost functions of the system's units the following system cost function is obtained:

$$F = \sum_{i=1}^{N} f_i\ (P_i^s) \tag{2.4}$$

Now, as dictated by the Lagrangian multiplier process (25), the constraint is added to the objective function to obtain the following equation:

$$\hat{F} = \sum_{i=1}^{N} f_i + \lambda\ (PD - \sum_{i=1}^{N} P_i^s), \tag{2.5}$$

where λ is the Lagrangian multiplier and can be interpreted as an incremental cost, as will be shown below.

This expression can be minimized by differentiating with respect to the power supplied and setting the result equal to zero.

$$\frac{\partial \hat{F}}{\partial P_i^s} = \left(\frac{\partial f_i(P_i^s)}{\partial P_i^s}\right) - \lambda = 0 \ , \ i = 1, 2, \ldots, N \tag{2.6}$$

Solving Eqn. (2.6) yields:

$$\lambda = \frac{\partial f_i(P_i^s)}{\partial P_i^s} \ , \ i = 1, 2, \ldots, N. \tag{2.7}$$

Thus λ is a constant and is interpreted as the incremental change in cost per incremental change in power setting. The optimal case is for the incremental cost per incremental power setting change to be equal for all units, i.e., the so-called equal incremental cost condition. This is correct for generating units which are not constrained by minimum and maximum power settings or for units which have λ as a function of P_i^s which cover the same λ region. This will be clearer in the discussion presented below. These conditions plus the condition of functions which are non-overlapping (i.e., λ (P_i^s) for one unit does not cover the same λ space as the λ (P_i^s) of another unit), cause problems for analysis which will be treated below. For a system with transmission losses the lambda dispatch process is very similar to the process discussed above. The difference lies in the fact that the constraint of "power demanded (P^D) equals the sum of all power produced"[25] is changed to "power demanded equals the sum of all power produced minus the system loss (P^L)", i.e.,

$$P^L > 0 \tag{2.9}$$

$$P^D = \sum_{i=1}^{N} P_i^s - P^L \tag{2.10}$$

This alteration of the constraints changes the Lagrangian function (F) to:

$$\hat{F} = F + \lambda \left(P^D + P^L - \sum_{i=1}^{N} P_i^s\right) \tag{2.11}$$

$$\hat{F} = \sum_{i=1}^{N} f_i(P_i^s) + \lambda \left(P^D + P^L - \sum_{i=1}^{N} P_i^s\right). \tag{2.12}$$

To optimize $\hat{F}$, differentiate with respect to the power supplied and set the result equal to zero, i.e.,

$$\frac{\partial \hat{F}}{\partial P_i^s} = \frac{\partial f_i}{\partial P_i^s} + \lambda \left(\frac{\partial P^L}{\partial P_i^s} - 1\right) = 0, \; i = 1, 2, \ldots, N. \tag{2.13}$$

The value $\left(1 - \frac{\partial P^L}{\partial P_i^s}\right)$ is called the incremental loss factor (ILF). When ILF is approximately zero Eqn. (2.13) reduces to the no transmission loss optimal solution, Eqn. (2.8). The penalty factor (L_i), i.e., the factor by which the losses affect the solution, is defined as

$$L_i = \left(\frac{\partial P^L}{\partial P_i^s} - 1\right)^{-1}, \; i = 1, 2, \ldots, N. \tag{2.14}$$

Equation (2.13) becomes:

$$\left(\frac{\partial f_i}{\partial P_i^s}\right) - \left(\frac{\lambda}{L_i}\right) = 0, \; i = 1, 2, \ldots, N. \tag{2.15}$$

Equation (2.15) can be solved for λ to yield

$$\lambda = L_i \frac{\partial f_i}{\partial P_i^s}, \; i = 1, 2, \ldots, N \tag{2.16}$$

Equation (2.16) means in a system with transmission losses, the optimal solution (power settings for all units) is the point where the penalized incremental costs are equal for all units. A theoretical overview of why this is desirable is presented in Appendix 1.

Unfortunately, lambda dispatch, by itself, cannot be used on a realistic basis. Independent unit constraints must also be used. This concept will be further developed in Chapter Four.

2.3 The Branch-And-Bound Method

The branch-and-bound technique is representative of new techniques which try to recognize often forgotten variables such as generation constraint and start-up costs. However, this turns into a tedious and often very difficult process that does not have the benefits that make it worth using for KPL or any Kansas electric utility. To become familiar with the reason for this judgement a brief description of the branch-and -bound technique as presented by Cohen and Lee[22] is given.

A precise definition of the branch-and-bound method comes from Cohen and Lee.[22]

"Branch-and-bound is a technique to solve a discrete variable problem by solving a sequence of similar problems derived from the original problem. The search is organized via a branch-and-bound tree (Fig. 2.1). The solution of each problem on the tree gives a lower bound on the solutions of all problems that are descendants, of that problem, on the tree. The leaves of a tree correspond to all the feasible solutions. The basic idea of branch-and-bound is that if, at any time, the solution of a lower-bound problem, say P, is greater than a feasible solution to the original problem (or in general an upper bound to the original problem), then it is not necessary to evaluate those nodes below P on the branch-and-bound tree since their solution must be greater than the existing feasible solution and therefore cannot be optimal."

A typical problem starts with each unit having a minimum and maximum start (time) interval ($s_i = [\underline{s}_i, \bar{s}_i]$) and stop interval ($e_i = [\underline{e}_i, \bar{e}_i]$). The problem then comes from trying to find the minimum cost solution where the start-up time is in the range of possible start-up times:

$$\underline{t}_i \ \varepsilon \ s_i \tag{2.17}$$

and the shut down time is in the range of possible shut down times.

$$\bar{t}_i \ \varepsilon \ e_i \tag{2.18}$$

The lower bound on the generator cost can be found at time k if all of the three following requirement are met:

1) Unit i is shut down before the start interval and after the stop interval.

$$P_i^k = 0, \text{ when } k \leq \underline{s}_i \text{ or } k > \bar{e}_i \tag{2.19}$$

2) If start and stop periods are disjoint the unit must be on at times between start and stop intervals.

$$P_i^k \ \varepsilon \ [P_i^{min}, P_i^{max}] \tag{2.20}$$

(Disjoint refers to the situation such that there is a period of time between the last point considered a start time and the first point considered as an end time, i.e., a unit must start by hour 4 but need not shut down until hour 7.)

3) The unit may be off or on at other times.

$$P_i^k \ \varepsilon \ (0, P_i^{max}) \text{ if } k \ \varepsilon \ [\underline{s}_i, s_i - 1] \text{ or } k \ \varepsilon \ [\underline{e}_i - 1, e_i] \tag{2.21}$$

This lower bound problem can be solved by solving at each necessary time point ($k = 1,2,\ldots,24$) for power levels P_i^k that minimize:

$$\sum_{i=1}^{N} L_i \, (P_i^k), \qquad i = 1,2,\ldots,N \tag{2.22}$$

Equation. (2.22) is constrained by the fact that total power generated must equal total system load (demand) at time k.

$$\sum_{i=1}^{N} P_i^k = L^k \qquad \text{for all } k \tag{2.23}$$

The units are constrained by Equations (2.19) - (2.21).

2.4 Dynamic Programming (Dynamic Programming and Linear Programming)

Dynamic Programming (DP), as presented by Lowery,[10] is a desirable method for solving the unit commitment problem when the problem dimensions are small because, as Lowery[10] states:

"...complicating factors: for example, fuel prices are not necessarily the same at all plants, the unit input-output curves are not straight lines emanating from the origin, and hot standby cost (if any), start-up and shut down costs are generally different for various units..."

Dynamic programming is a very good method of determining the optimum combination of units given a small set of units and system power demand. The purpose of the DP method is to find the unit's optimal output between the unit's minimum and maximum power production capacity. The advantage is that in solving the system for N units it becomes simpler to find the optimal unit use for N+1 units. The following is the theoretical development of DP for the unit commitment problem as presented by Lowery.[10]

Similar to the lambda dispatch development, one of the first things that is recognized is the power production capacity constraints (P_i^{min}, P_i^{max}) on each i unit.

$$P_i^{min} < P_i < P_i^{max} \text{ for } i = 1,2,\ldots,N, \tag{2.24}$$

where P_i^{min}, P_i, P_i^{max} are the power production minimum, actual production level, and power production maximum for unit i.

Also the power level should be allowed to be zero since it may be more economic to turn off the unit. Thus:

$$PP_N = \{P_i | P_i = 0 \text{ or } P_i^{min} \leq P_i \leq P_i^{max}\}, \tag{2.25}$$

which reads PP_N equals the set of all P_i such that $P_i = 0$ or $P_i^{min} < P_i < P_i^{max}$.

The cost function is defined as the minimum cost in dollars per hour of generating power to meet the demand by using the first N units.

$$\text{cost function:} \quad f_N(x) \tag{2.26}$$

This means that the admissible x values in $f_N(x)$ are $x = 0$ and $c^{min} < x < c^{max}$ where

$$c^{min} = \text{Min}[P_1^{min}, P_2^{min},\ldots,P_N^{min}] \tag{2.27}$$

and

$$c^{max} = \sum_{i=1}^{N} P_i^{max} \tag{2.28}$$

A general form for the Nth set is:

$$X_N = \{x | X = 0 \text{ or } c^{min} \leq X \leq c^{max}\}. \tag{2.29}$$

Letting $g_i(P_i)$ be the cost curve of the ith unit (the dollar cost per hour of generating P_i MW on the unit i) one must now consider the expression:

$$g_n(P) + f_{N-1}(x-P) \tag{2.30}$$

for $P_i \leq PP_N$. (P_i is an element of set PP_N) and $(x-P) \varepsilon X_{N-1}$. This then gives the total generation cost

$$P + (x-P) = X \text{ MW}, \tag{2.31}$$

By definition $f_{N-1}(x-P)$ is the minimum generation cost for producing (x-P) MW. Thus, to get $F_N(x)$, P must be chosen to minimize Eqn. (2.30). This means one can obtain the functional equation:

$$f_N(x) = \underset{\substack{P \varepsilon PP_N \\ (X-P) \varepsilon X_{N-1}}}{\text{MIN}} \{g_n(P)+f_{N-1}(x-P)\}, \text{ for } N = 2,3,... \tag{2.32}$$

From which one has:

$$f_1(x) = g_1(x), \tag{2.33}$$

since if only one unit can be used, the choice has to be to produce the entire demand on that unit.

Since one knows $f_1(x)$ is known for $x \varepsilon X_1$, Eqn. (2.32) can be used to determine $f_2(x)$ for $x \varepsilon X_2$. Then, the $f_2(x)$ value and Eqn. (2.32) are used to find the $f_3(x)$ value.

The use of dynamic programming has recently been expanded by Van Meeteren by combining it with linear programming (LP). Van Meeteren presents two ways to obtain an initial solution:

: minimize limited fuel unit commitment followed by fuel allocation.

: approximate limited fuel prescheduling and unit commitment.

The second of these will be presented because it deals specifically with unit commitment. This approach uses input-output (IO) models that have upper bounds which are convex in nature. This allows approximate unit commitment and fuel allocation to be determined.[11]

With this process unit commitment follows fuel allocation. The results of the fuel allocation can be included in two different ways:

:allocate the limited fuel, available for the hour, to the entire system to units that are committed by a combination processor.

:set fuel allocation of each unit for a fixed schedule. Any increase of fuel use will have to come from "unlimited fuels".

The second approach was chosen by Van Meeteren because of expected better results than the first approach as the optimal solution is approached. The first piece of given information is the representation of the IO model used in this analysis, Fig. 2.2. The total fuel cost is given by combining the cost of the units which are designated as being usable.

$$c^{\ell}Q^{\ell} = f^{\ell} \ , \tag{2.34}$$

where $c\ell$ is the lower bound cost, Q^{ℓ} is the lower bound heat rate, and f^{ℓ} the lower bound cost function.

The total fuel cost of the upper bounded unit is

$$c^{u}Q^{u} = f^{u} \tag{2.35}$$

where c^{u} is the upper bound cost, Q^{u} is the lower bound heat rate, and f^{u} the upper bound cost function.

Thus:

$$C = c^{\ell}Q^{\ell} + c^{u}Q^{u} = f^{\ell}+f^{u} \tag{2.36}$$

By assuming the unlimited fuel type is used in measuring the IO curve one can say that the lower limit product of efficiency and heat rate added to the upper limit product of the same two multiplicants equals the product of the upper bound efficiency and the upper bound of the IO curve.

Thus:

$$\eta^{\ell}Q^{\ell} + \eta^{u}\ Q^{u} = \eta^{u}\ H^{u}(P) \tag{2.37}$$

By using the substitution principle with these last two equations the following equations can be derived:

$$C = \left(C^{\ell} - \frac{\eta^{\ell}}{\eta^{u}}\, C^{u}\right) Q^{\ell} + C^{u}\ H^{u}\ (P) \tag{2.38}$$

From this equation and Fig. 2.2, we note that the IO curve is related only to the unlimited fuel types.

Linear programming is used in almost all the other areas except for unit commitment. Dynamic programming is what is used for the actual unit commitment. Other recent works completed, e.g., Roy and Rao[21] and Trefny and Lee,[20] have also proven worth discussion.

2.5 Cartesian Coordinate Formulation[21]

Roy and Rao[21] presented a study in which a cartesian coordinate formulation is the bases for optimal real and reactive power generations. The method of solution is summarized as follows.

First minimize the objective (cost function) L_i as in Eqn. (2.3). Recognize that now the power setting has two components, real and reactive.

$$P_i^s = P^s(e,f) = w(x,\mu), \tag{2.39}$$

where e = real power and f = reactive power

This is subject to two constraints. First, the constraint of total power (real and reactive) must equal 0:

$$y(x,u) = 0 \ , \tag{2.40}$$

where x is the dependent variable expressed as

$$x = P(e,f) + C = 0 \text{ , the total real power load} \tag{2.41}$$

u is the control variable expressed as

$$u = Q(e,f) + D = 0\text{, the total reactive power load} \tag{2.42}$$

The second constraint is the voltage magnitude constraint which must be zero or above (negative voltage values cannot exist).

$$z(x,u) \leq 0 \ , \tag{2.43}$$

where x and u are as stated in Eqns. (2.40) and (2.41)

The Lagrangian function is then formed as:

$$F(x,u,\lambda) = w(x,u) + p(x,u) + \lambda * y(x,u), \tag{2.44}$$

where p(x,u) is the term corresponding to the sum of the penalty term times the square of the deviation from the limit.

Every time a limit is violated there is a penalty associated with it that can be expressed as:

$$p(x,u) = r_i h_i \qquad \text{where } i = 1,2,\ldots,N. \tag{2.45}$$

When w(x,u) is minimized the following conditions should be satisfied for the optimal solution:

$$\frac{\partial F}{\partial \lambda} = y(x,u) = 0 \tag{2.46}$$

$$\frac{\partial F}{\partial \lambda} = \frac{\partial w}{\partial x} + \frac{\partial P}{\partial x} + \left(\frac{\partial y}{\partial x}\right) \lambda = 0 \tag{2.47}$$

$$\frac{\partial F}{\partial u} = \frac{\partial w}{\partial u} + \frac{\partial P}{\partial u} + \left(\frac{\partial y}{\partial u}\right) \lambda = 0 \tag{2.48}$$

Comparison of these methods yields the conclusion that this method is, in effect, the lambda dispatch solution which includes transmission losses and fuel constraints.

2.6 Standard Load Constraints

The method developed by Trefny and Lee[20] parallels the work by Rao and Roys[21] by using applicable fuel constraints but in addition their method includes standard load constraints. Another difference is that the model used is not quadratic but it is non-linear. The non-linearity stems from the fact that the third term of the heat rate expression is cubed instead of squared as is most generally done. The steps followed for problem formulation are presented by Trefny and Lee[20] and are summarized as follows.

Find vector $\bar{x}^*$ to minimize (with respect to $\bar{x}$):

$$e(\bar{x}) = \sum_{i=1}^{N} E_i = \sum_{i=1}^{N} (A_i + B_i X_i + D_i X_i^3) = \text{heat rate, } i = 1,2,\ldots N. \tag{2.49}$$

with respect to X.

The first constraint is expresses as:

$$p(\bar{x}) = \text{Load} - \sum_{i=1}^{N} X_i = 0, \; i = 1,2,\ldots,N, \qquad (2.50)$$

where X_i equals the generating level of unit i and x is the vector of real variables $X_1, X_2, \ldots, X_N$

The second constraint is that the production level of unit i lies in between the maximum level and the minimum level:

$$LGL_i \leq X_i \leq HGL_i. \qquad (2.51)$$

The Lagrangian equation is now developed taking into account the objective function and constraint:

$$L(\bar{x},\lambda) \overset{d}{=} e(\bar{x}) + \lambda\, p(\bar{x}) \qquad (2.52)$$

Expanding this equation one can derive the following:

$$L(\bar{x},\lambda) = \sum_{i=1}^{N} (A_i + \beta_i X_i + D_i X_i^3) + \lambda\, (\text{Load} = \sum_{i=1}^{N} X_i) \qquad (2.53)$$

Using the condition that $e(\bar{x}^*)$ be a constrained minimum expressed as:

$$\bar{\nabla} L(\bar{x}^*, \lambda^*) = 0 \qquad (2.54)$$

and assuming that equation (2.50) is satisfied yields the following local minimum:

$$\frac{\partial L_i}{\partial X_i} = \beta_i + 3\, D_i\, X_i^2 = \lambda, \; i = 1,2,\ldots,N \qquad (2.55)$$

Both of these last two methods have brought in the use of fuel constraints and transmission loss cases. Much important work has been done in the area of transmission loss with probably the most popular work done by Dommel and Tinney.[12]

2.7 Dommel-Tinney Method - Load Flow Analysis

As stated by H. H. Happ:[1]

> "The work in reference 66 (Dommel and Tinney) must be ranked as one of the most important that has so far been advanced in solution techniques of the optimal load flow problem."

For this reason, the process is discussed below.

As Dommel and Tinney[12] recognized in their work, there are two cases that should be considered when working with the load flow problem. First, the optimal real and reactive power flow case, where the objective function equals the instantaneous operating costs, the solution then equals the exact optimal dispatch. Second, when the optimal reactive power flow objective function equals the total system losses, the solution yields minimum losses.

Before continuing, basic terminology from this work should be understood. A node is a point from which power is supplied. While it can include a generating unit it may not necessarily include one. It can also be a tie-line, a point at which transmission lines from two or more units come together. V_i denotes the voltage magnitude at node i while θ_i denotes the voltage phase angle at node i. $G_i^m + jB_i^m$ is the element of the nodal admittance matrix devised specifically for this work. The superscript "m" denotes the system being used. P_i is the net real (actual) power entering node i and Q_i is the reactive (loss) power entering node i.

With this terminology in mind, the feasible power plant settings begin with the voltage equations involving the real and reactive quantities.

$$P_i - j\theta_i = (V_i) - j\theta_i \sum_{i=1}^{N} (G_i^m + j\beta_i^m)(V^m)\, e^{j\theta^m}, \quad \begin{array}{l} i = 1,2,\ldots,N \\ j = 1,2,\ldots,M \end{array} \tag{2.56}$$

This is broken down into the equality constraints:

$$P_i(V,\theta) - P_i = 0 \tag{2.57}$$

$$Q_i(V,\theta) - Q_i = 0 \tag{2.58}$$

All of the relevant unknows (V,θ) are then placed into one vector with all the specified values being put into a separate vector. The polar form of Newton's method[12] is then used with the Jacobian matrix[12] to derive the solution.

Optimal power flow is considered with and without the inequality constraints. Without the constraints the cost function is as before:

$$F = \sum_{i=1}^{N} f_i\,(P_i^s),\ i = 1,2,\ldots,N \tag{2.59}$$

It is realized that with no power costs associated with the slack node (also called node 1 or the reference node where $\theta_1 = 0$, V and θ values are specified while real and reactive power values must be determined) that the minimizing process would attempt to supply the slack node with all the power:

$$F = P_1(V,\theta). \tag{2.60}$$

The fixed variable vector can be grouped into separate parts: the control parameters ([u]) which are the real and reactive powers generated and the fixed (or disturbance) parameters ([p]) which are the power demanded. Thus:

$$[g] = \begin{pmatrix} [u] \\ [p] \end{pmatrix} . \tag{2.61}$$

From here the classic differentiation and Lagrangian techniques are performed subject to equality constraints:

$$[g(x,u,p)] = 0 \tag{2.62}$$

The equations produced are nonlinear and are most simply solved by the gradient method (steepest decent).

The with-equality-constraints procedure follows basically the same pattern as presented above except that the control vector parameters are now constrained as

$$[u^{min}] < [u] < [u^{max}] \tag{2.63}$$

From this the Kuhn-Tucker theorem proves that the following conditions must hold true in order for the minimum to be obtained (given convex functions).

1) The functional change per control vector unit change equals zero when the control vector value lies between the minimum and maximum values:

$$\frac{\partial f}{\partial u} = 0 \qquad , \text{ if } u_i^{min} < u_i < u_i^{max} \tag{2.64}$$

2) The functional change per control vector unit change equals or is less than zero if the control vector value is the maximum possible value:

$$\frac{\partial f}{\partial u} \leq 0 \qquad , \text{ if } u_i = u_i^{max}. \tag{2.65}$$

3) The functional change per control vector unit charge is greater than or equal to zero if the control vector value is the minimum possible value:

$$\frac{\partial f}{\partial u} \geq 0 \qquad , \text{ if } u_i = u_i^{min}. \qquad (2.66)$$

In order to complete the solution process the complexities found in the functional inequality constraints, which present themselves in this technique, are dealt with in terms of a penalty method. When constraints are violated, the objective function adds in a penalty weight factor (W) which then adjusts the solution values. Therefore, the objective function, which is generally referred to as an augmented cost function is:

$$f = f(x,u) + \sum_{i=1}^{N} w_i. \qquad (2.67)$$

Using differential calculus, Lagrangian multipliers, Jacobian matrices, and the above-mentioned iterative process yields a minimum cost.

2.8 Alsac and Stott - Load Flow Analysis: Transmission Loss

Further work was done on the DT method by Alsac and Stott in 1973. The basic outline followed by their approach is as stated below.[13]

1) Solve the optimal case load flow by DT.
2) Monitor the outage-security using a fast AC (voltage) load-flow method. (Outage-security deals with chances of unexpected unit shut-downs)
3) Continue the optimal load-flow solution, using constraints uncovered by each step until all insecurities have been reached and/or one optimum has been reached.
4) Recycle from step 2 until an optimum secure solution is obtained.

The mathematical formulation of this problem is as follows:[13] The objective function is a function of system control and state variables expressed as

$$f = f(x^{\circ},u]. \tag{2.68}$$

The node load-flow equations are the equality constraints expressed as

$$[g^{\circ}(x^{\circ},u] = 0, \tag{2.69}$$

with inequality constraints being plant and transmission system operating limits expressed as a vector inequality

$$[h^{\circ}(x^{\circ},u)] \leq 0. \tag{2.70}$$

The security constraints are developed next. There are the additional equality and inequality constraints associated with outage contingencies (the chance of a unit not being able to produce the necessary power when needed). These constraints are characterized into two different types. First the nodal load-flow equations, expressed as:

$$[g^{k}(x^{k},u)] = 0, \tag{2.71}$$

and second, plant and transmission system operation limits expressed as:

$$[h^{k}(x^{k},u)] \leq 0. \tag{2.72}$$

2.9 B Coefficient Method

To account for transmission losses a load flow analysis is often used which requires considerable knowledge and description of the utility transmission system. To meet the demands of a grid system the power can flow from any generator which is on line to any point in the system which demands power. Thus, for a system with ten generators on line and 100 demand points requires characterization of the transmission lines between any generator and any demand point, i.e., 10 * 100 or 1000 transmission line characterizations.

To alleviate the dimensionality of this problem Kirchmayer[26] and others have transformed this problem into one in which the demands at all points on the system are viewed as one system demand supplied with power by all on-line generators which are connected in parallel. Thus, the transmission losses are represented by a double sum of the triple product of source loadings and constants which characterize the system,

$$P^L = \sum_{m=1}^{N} \sum_{n=1}^{N} P^{sm} B_n^m P_n^s, \tag{2.73}$$

where B_n^m are the coefficients which characterize the power system.

Happ[1] states, in his work of comparing classic λ dispatch including line losses by the B coefficient method to more rigorous and newer methods:

"... It was concluded therefore that from an economic standpoint the classic technique does as good a job as the rigorous method so long as the B matrix is updated to incorporate important line changes. Current B matrix techniques are at a level where updating is possible."

2.10 Lambda Dispatch - Justification

With all the techniques presented one may wonder how the no-transmission loss lambda dispatch can be selected as the proper technique. First, in looking at the branch-and-bound technique, this technique states that its biggest asset is that units need not be prioritized with respect to the cost of running them. With the KPL problem this has already been done, it is given information and the

Lagrangian method generally gives a more optimal solution (Appendix 1). The same holds true for dynamic programming and linear programming used in conjunction with dynamic programming. While given solutions are feasible to some, the feasible solution obtained from Lagrangian multipliers are generally more optimal. In addition, dynamic programming becomes almost useless if the dimension of the problem (number of variables) becomes very large. The Lagrangian method is not limited in this way.

As far as the work done by Roy and Rao,[21] Trefny and Lee,[20] Dommel and Tinney,[12] and Alsac and Stott[13] all these works deal with transmission loss cases which can become very involved. As it stands now the dispatch solutions established by KPL do not directly deal with transmission losses. The reasons for this will be further explained in Chapter 4, but simply stated, transmission losses are just added in as part of the actual demand so that the no-transmission case can be applied. The same reasoning is used for not using the B method. The B method was not included in this work for the reason that KPL currently does not consider transmission losses an important parameter in their dispatch solutions. (B coefficients for KPL's system have just been developed but were unavailable for this study.) The no-transmission loss technique is a much simpler technique so that there is no need to involve transmission loss and load flow equations with the dispatch solution at this time.

In essence, while the techniques may be good for specific situations, none of these situations exist with the KPL scenario. The situation, as it exists today, lends itself most readily to the no-transmission loss lambda dispatch method. Further, as Happ[1] stated:

"A comparison study was recently undertaken by this author aimed at determining the financial benefits of changing from the classic MW dispatch to a rigorous method of dispatching... No significant difference in production costs were realized, although there were differences in the two dispatches provided;....

The reason then for employing more advanced techniques cannot be on the basis of savings alone, but because more rigorous models are required for executing different functions associated with the security of operations."

This need does not presently exist at KPL on a level that would call for the use of these other techniques. In fact, the less rigorous technique is even less expensive as will be shown in Chapter 5.

2.11 Dispatch Equations:

From the material presented thus far, the technique which seems to hold the most promise for KPL with respect to the degree of difficulty and time the method takes is a constrained lambda dispatch, with no-transmission losses. As stated, the reason for no loss will be discussed in Chapter Four. The constrained concept comes from unit generation capacity constraints (P^{min}, P^{max}).

As stated previously, this concept will now be expanded upon as follows: 1) Development of equations determining a lambda when the unit power setting is known. 2) Development of equations determining unit power setting when lambda is known. 3) Development of equations determining individual unit settings when system load and usable system candidate units are known.

1) Determining Incremental Cost Setting:

Recall Eqn. (2.8),

$$\lambda = \left(\frac{\partial f_i^s (P_i^s)}{\partial P_i^s}\right) \tag{2.8}$$

Also recall the relationship:

$$f_i (P_i^s) = (\alpha_i + \beta_i P_i^s + \gamma_i P_i^{s2}) c_i \tag{2.3}$$

Thus, Eqn. (2.8) becomes:

$$\lambda = (\beta_i + 2\gamma_i P_i^s) c_i \tag{2.74}$$

Hence, when everything is known about any individual unit, i.e., β_i, γ_i, and c_i,

$$P_i^s = \left(\frac{\lambda - \beta_i c_i}{2 \gamma_i c_i}\right). \tag{2.75}$$

2) Determining Individual Unit Power Settings when System Demand and Usable Units are Known

The formulation of the system lambda equation for the case where power demand and the candidate units are known can best be demonstrated by example. Thus, the following three examples.

2 Bus Problem

The Lagrange function is [Eqn. (2.5)]

$$\hat{F} = f_1 + f_2 + \lambda P^{TOT} - \lambda P_1 - \lambda P_2 \tag{2.76}$$

$$\hat{F} = c_1(\alpha_1+\beta_1 P_1+\gamma_1 P_1{}^2) + c_2(\alpha_2+\beta_2 P_2+\gamma_2 P_2{}^2) - \lambda(P_1+P_2-P^{TOT}) \tag{2.77}$$

Differentiating Eqn. (2.77) with respect to, first, P_1, and, second P_2, and equating the results to zero yields

$$P_1 = (\lambda - c_1\beta_1)/2c_1\gamma_1 \quad (2.78)$$

$$P_2 = (\lambda - c_2\beta_2)/2c_2\gamma_2 \quad (2.79)$$

Use the constraint [Eqn. (2.2)], i.e.,

$$P^{TOT} = P_1 + P_2, \quad (2.80)$$

yields three equations and three unknowns, namely, P_1, P_2, and

$$\lambda = [2(c_1\gamma_1)(c_2\gamma_2)\ P^{TOT} + (c_1\beta_1)(c_2\gamma_2) + (c_2\beta_2)(c_1\gamma_1)]/(c_1\gamma_1+c_2\gamma_2) \quad (2.81)$$

3 Bus Problem

For this problem just add a similar equation as Eqns. (2.77) and (2.78) for P_3,

$$P_3 = (\lambda - c_3\beta_3)/3c_3\gamma_3. \quad (2.82)$$

Add P_3 to the left hand side of Eqn. (2.79) to obtain

$$P^{TOT} = P_1 + P_2 + P_3. \quad (2.83)$$

Solve Eqns. (2.78), (2.79), (2.82), and (2.83) for λ

$$\lambda = [2(c_1\gamma_1)(c_2\gamma_2)(c_3\gamma_3)\ P^{TOT} + (c_2\gamma_3)(c_1\beta_1) + (c_1\gamma_1)(c_3,\gamma_3)(c_2\beta_2)$$
(2.84)

$$+ (c_1\gamma_1)(c_2\gamma_2)(c_3\beta_3)]/[(c_2\gamma_2)(c_3\gamma_3) + (c_1\gamma_1)(c_3\gamma_3) + (c_1\gamma_1)(c_2\gamma_2)]$$

N Bus Problem

This procedure can be generalized by noting the solution form for λ [Eqns. (2.81) and (2.84)]

$$\lambda = [2P^{TOT} \prod_{i=1}^{N} c_i\gamma_i + \sum_{i=1}^{N} c_i\ \beta_i \prod_{\substack{j=1 \\ j\neq i}}^{N} c_i\gamma_i]/\left(\sum_{i=1}^{N} \prod_{\substack{j=1 \\ j\neq i}}^{N} c_i\gamma_i\right) \quad (2.85)$$

Equation (2.85) is the generalized form for the Lagrangian multiplier (λ) in terms of the fuel cost and input/output function coefficients for all units and the total system demand (P^{TOT}). The value for λ, calculated from Eqn. (2.84), can be used in Eqns. like (2.78), (2.79), and (2.82) to find the optimum power settings, P_i, $i = 1,2,\ldots,N$, to satisfy the total system demand (P^{TOT}).

This equation is of utmost importance to the procedure followed by the program developed in this work. It will be referred to often in the discussion of the algorithm and computer program (Chapter 4).

2.12 Development of Lambda versus Power Plot:

In order to understand the constrained lambda dispatch (no loss) problem and to discuss the different scenarios clearly, the plot shown in Fig. (2.3) is essential. However, in order to understand this plot one must understand the origin of the plot.

Figure (2.3) was developed solely from Eqn. (2.74).

$$\lambda = (\beta_i c_i + 2\gamma_i c_i \; P_i^s)$$

The unit's maximum lambda value was calculated by using the unit's maximum level of power generation.

$$\lambda_i^{max} = (\beta_i + 2\gamma_i \; P_i^{max}) \; c_i, \quad i = 1,\, 2,\, \ldots,\, N. \tag{2.85}$$

The unit's minimum lambda value was calculated using its minimum power level.

$$\lambda_i^{min} = (\beta_i + 2\gamma_i \; P_i^{min}) \; c_i \quad , \; i = 1,2,\ldots,N. \tag{2.86}$$

The minimum and maximum power levels as well as the input/output function coefficients were provided by Robert Fackler.[27] The data as well as calculated lambda values are given in Table 2.1.

After calculating the minimum and maximum lambda values for each unit these values were plotted against their respective power values. A line was drawn to connect each unit's minimum and maximum lambda values. By noting the region the line covers one can see what power range and incremental cost range each unit covers as well as which units are more or less expensive at given power levels (Fig. 2.3). Figure 2.3 provides a guide to aid in the selection of allowable optimum solutions to satisfy a system demand using the (λ) dispatch procedure.

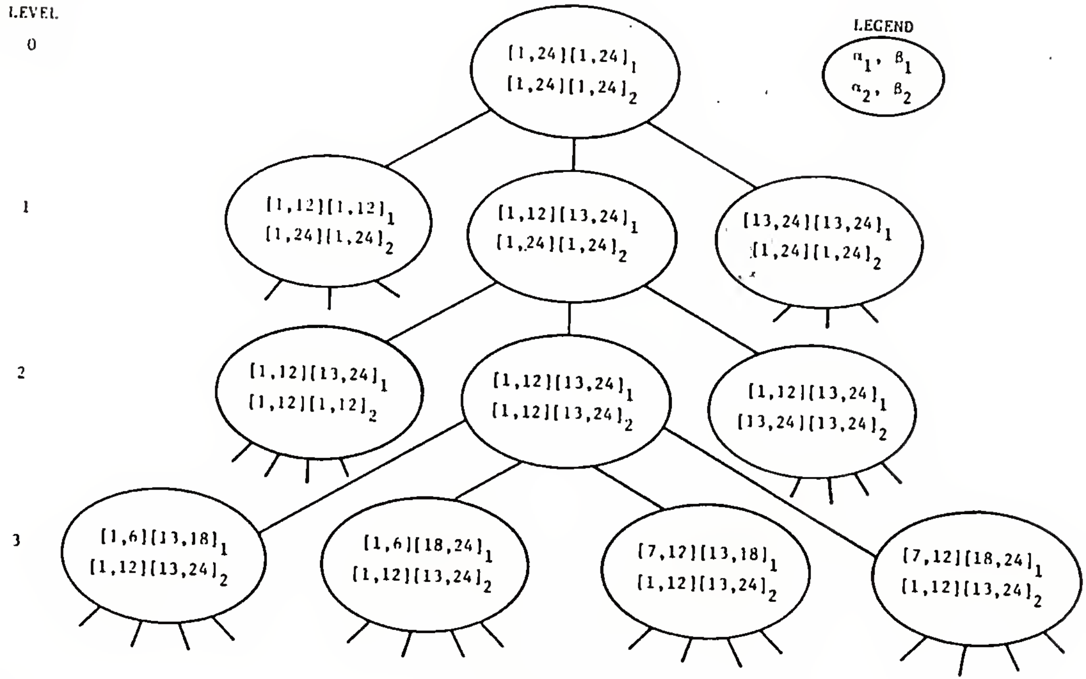

Fig. 2.1. Branch-and-Bound Decision Tree (From Reference 22).

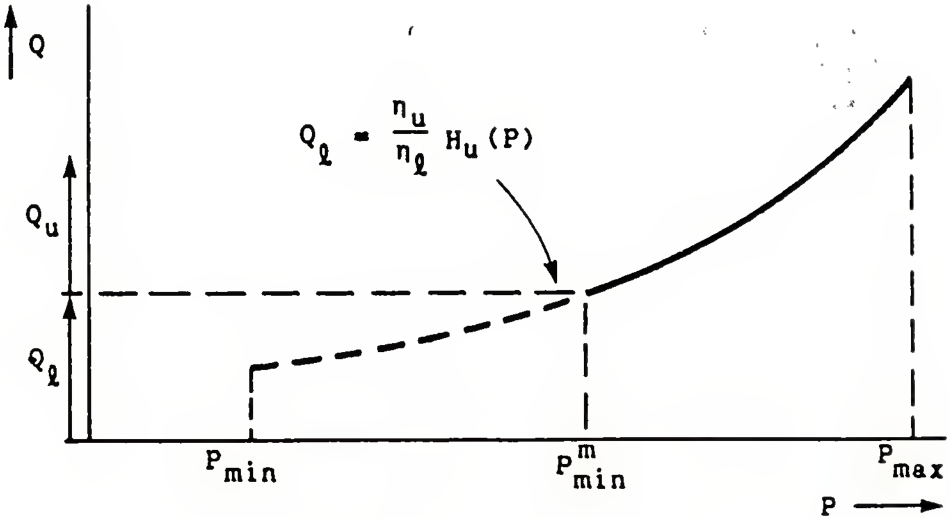

Fig. 2.2. Input/Output Model (From Reference 11).

Table 2.1. Given KPL Data (From Reference 26)

Unit	α (MBTu/h)	β (MBTu/MW^{2}h)	$\gamma(10^3)$ (MBTu/MW^{2}h)	C ($/MBTu)	Power MIN (MW)	Power MAX (MW)	Lambda MIN $/MWh	Lambda MAX $/MWh
J1	518.798	9.070	1.86	1.45	165	405	14.042	15.336
J2	508.525	8.891	1.82	1.45	165	405	13.764	15.033
J3	513.662	8.981	1.84	1.45	165	405	13.903	15.185
J5	550.969	7.223	5.22	2.20	120	270	18.650	22.097
L3	78.404	9.019	14.5	2.20	20	45	21.131	22.742
T7	110.335	8.786	11.3	2.22	20	65	20.510	22.771
L4	169.431	8.225	10.6	2.20	30	55	19.496	20.663
H4	152.679	8.449	8.60	2.90	55	140	27.247	31.487
H3	32.258	11.906	24.2	2.90	15	30	36.636	38.744
H2	15.514	12.120	58.5	2.90	10	19	38.541	41.594
MCP2	34.699	9.031	089.	2.90	15	25	33.832	38.927
ABILE CT	232.100	8.453	4.300	2.99	25	65	25.917	26.946

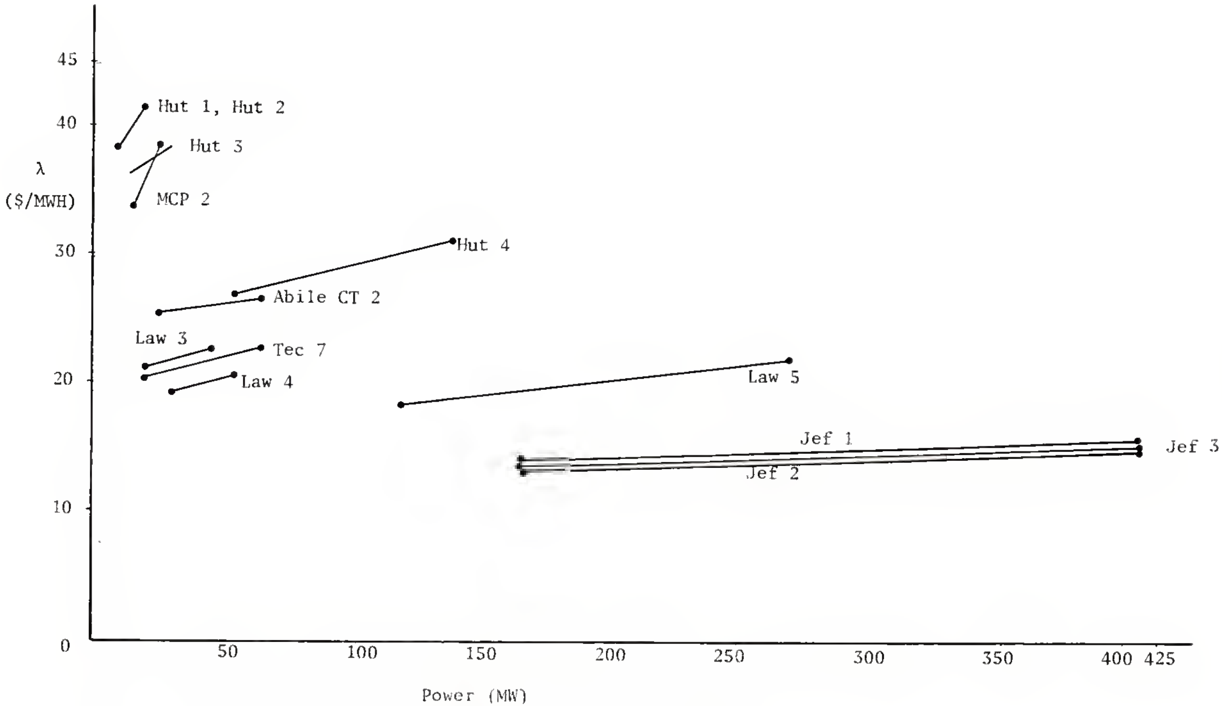

Fig. 2.3 Incremental Costs and Power Production Relationship

3.0 EPRI Economic Dispatch Program

3.1 Introduction

The objective of this research is to develop a computer code for a unit commitment fuel scheduling program in Basic Language to use on the IBM:PC and/or compatible machines and compare results of the PC code to that of the Electric Power Research Institute (EPRI) computer code presently being used by the Kansas Power and Light Company (KPL) on a time-sharing basis with the Boeing Computer Services Company. If the results are comparable, the use of the PC code could reduce the cost of unit commitment/fuel scheduling because use of the PC can be less expensive than the time-sharing program. Also KPL personnel will have more control over the PC code than they presently have with the time-sharing code.

This chapter is composed of two main parts. The first part describes the EPRI program presently being used by KPL. The second part of this chapter is devoted to the presentation and discussion of variable input and relevant portions of the PC computer program.

3.2 Economic Dispatch Program (EDP) Presentation

The EPRI program was divided into three distinct parts: the long term (year), the mid-term (month/week), and short term (daily). The section pertaining directly to the PC code development is the mid-term (month/week). This is the only section that is presently used by KPL personnel. It is still on only a trial basis there. However, its results are being used to determine the best unit loading (fuel requirements) schedule for a week, given specific power demands.

The mid-term has much variation. It can be presented in short form or long form, priority lists for variable plants and units used can either be calculated by hand or developed by the computer. The period in question can vary from a day to a month, several fuel types for each plant can be dealt with at one time, as well as other variants. The results as shown include such things as total generation costs, system lambdas as they change every hour, system fuel use summaries, as well as others. These results are checked by personnel for the final decision of whether a unit should be brought on-line or taken off-line.

3.3 Model System

The explanation of the system model can be broken down into two distinct areas. First, a general description of the overall system with the basic assumption used in the setup of the system. Second, the data, in terms of what it looks like and what it means.

3.4 Description and Assumptions

The overall set-up for the KPL system consists of 19 separate units in six different plants. Six of these units are combustion turbines and the other 13 are steam-type generators. Combustion turbines are generally more expensive to run over a long period but can be very useful for meeting short-time peak demands. This is true because combustion turbines do not take as long to fire-up and once there they do not take as long to cool down. Table 3.1 is a listing of the 19 separate units with their respective unit type.

There are two types of fuel used: coal and natural gas. As shown in Table 3.1, the 13 steam engines use coal and the six combustion turbines use natural gas. Oil is also a viable fuel source but it was not included in the set of data supplied by KPL.

No interchange data are considered. Interchange is the condition when extra electrical power must be purchased (sold) because the maximum (minimum) capacity of the available units has been exceeded (can not be used).

Two interesting characteristics can be seen in the assumption used for these data. Generally each separate unit is considered as an individual bus (a node point in the system circuit), of which there can be three kinds: (the real and reactive power demanded is known for all three)

1) The real and reactive power generated are known. The voltage magnitude and phase angle are solved for.
2) The real power generated and voltage magnitudes are known. The reactive power generated and phase angle are solved for.
3) The voltage magnitude and phase angles are known. The real and reactive power generated are solved for.

In addition, some buses are supplied with generators while others are not.

For the KPL analysis, static load flow equations (SLFE) are not necessary in calculating the independent unit load and demands as is usually done. With this set of data all 19 units are considered as ONE bus subject to meeting ONE demand and no transmission losses between generators and demands are included.

Transmission loss analysis is quite an involved process which includes several iterative steps to determine the appropriate line loss between each generator and each demand point. Instead, KPL assumes eight percent of each demand can be attributed directly to system transmission losses. In this way the time consuming B-coefficient use is avoided and the unit commitment/fuel scheduling process is simplified.

3.5 Data Description

There can be as many as 17 different types of data for each unit. However, only 10 types are used in this analysis as supplied by KPL. Table 3.2 shows the 17 possible data types and indicates which data types are used. Table 3.3 shows the sample input values, as supplied by KPL. In order to understand the data and what it means, each data type and its respective data values will be discussed as presented in Volume 3, Section 6 of the EPRI study reporting on long term, mid-term and short term unit commitment.

There are ten data types which deserve specific recognition. These ten are Model Description, Generation Unit Identification, Generating Unit Performance Characteristics, Generating Unit Cost, Initial Condition, Manual Schedule, Load Model, Plant Identification, Plant Fuel, and Fuel Identification.

3.5.1 Model Description: This set of data serves a very broad purpose. For example, it is in this set of data that the period considered is determined as well as what form of output is desired and how losses are handled with the model. Load types, since they vary from day-to-day as well as season-to-season, are determined as are peaking values for load data.

In addition, the choice of using priority lists is decided here as well as the initial and final convergence limits that should be used for the iterative processes. Maximum allowable changes in λ for large changes in iteration values, production costing schemes, and loss estimation parameters are also dealt with.

Other data used deals with the spinning reserves necessary by the hour as well as reserves on hand. Proportional cold-start cost of unit cost, number of system entry points, and interfacing capabilities (so that stored Long-Term program data can be used) are presented also.

3.5.2 Generating Unit Identification: This set of data identifies all the generating units and system tie lines. Informative data that are included here are the unit name as well as what type of unit it is (dispatchable, non-dispatchable, hydro, interchange tie line). If necessary the entry point where the unit enters the system and the plant number of which the particular unit is a part is specified. In addition the unit's individual priority code with respect to other units, the code indicating the fuel used and the maximum and minimum power generation limits are presented.

3.5.3 Generating Unit Performance Gharacteristics: This data section describes the input/output (IO) curve, i.e., the energy required per hour for each unit as a function of generator power setting, in addition to the start-up and cool-down times for each individual unit. This means that the constant, linear, and quadratic terms of the I/O model are described here (see Eqn. (2.3)).

3.5.4 Generating Unit Gost: Data in this section include the cold start cost and boiler cool-down times along with a constant reciprocal penalty factor (which is optimal) for dispatchable units.

3.5.5 Initial Conditions: This section describes the units characteristics before the scheme begins. The specific characteristic mentioned is for how many consecutive periods the unit has been on- or off-line previous to the time period for which the program is being run.

3.5.6 Manual Schedule: This section allows one to control what units may or may not be used. The time period being modeled is required here as well as what type of unit is being used. Also, the fixed MW level of generation is presented.

3.5.7 Load Model: These data are used to normalize the load data for every hour of every day. The input value is the fraction of the total peak that is expected to occur.

3.5.8 Plant Identification: These data are used for reference purposes. Each plant used in the schedule is defined by a number. This number is used throughout the input data whenever the plan is being referenced.

3.5.9 Plant Fuel: These data are input by plan instead of by unit as done most frequently up to this point. The necessary information presented is the plant number, its individual fuel type, the average and dispatch fuel price, the target, minimum, and maximum (MBtu) fuel use for the commitment schedule period, and the number of additional fuel constraint periods. Additional fuel constraints can be added, if necessary.

3.5.10 Fuel Identification: This is also a reference process. As with the specified plants, each type of fuel used is referenced with a code number and thereafter the code number is used in place of the fuel name.

3.6 The Selection IBM:PC - (Why and How)

The reason for selecting the IBM:PC above other personal computer types is because KPL, for whom this research should directly benefit as well as being the company that supplied the data, have IBM:PC compatible computers in their offices. The IBM:PC is very widely used throughout the business and scientific communities. Thus, the transportability of the computer code for use by other electric utilities may allow for significant monetary savings when solving their kinds of problems. Thus, using common equipment can easily result in more common use.

In order to gain some insight about how optimality ideas are formed, the following points about the KPL data are offered. 1) There are certain system constraints which are inherent to the system. Structural flaws and defects in the units as well as line impedance and load carrying ability from bus-to-bus are limitations which exist but must be considered as part of the system. 2) Location of units with respect to one another is also a situation that must be accepted and dealt with. One obviously can not ask that, since area demand has switched from one area to another, the individual units should be moved to correct for such a problem. 3) The minimum and maximum power generating limits of all units are limitations which also must be accepted and not changed. 4) Finally, the entire scenario depends on demand. However, knowledge of specific demand values will never be known. The future can not be read in this industry. This is a system in which one must judge, to the best of one's capabilities, the need that must be met -- for the need MUST be met. This is the sole reason for the existence of this system. In view of these points, the following is the general flow of events in developing a simple modeling scheme.

As a common first step a logic (flow) diagram will have to be established in order to follow the process and its many "twists and turns" from beginning to end.

Next a program code (in Basic Language) will have to be developed. When the code has been developed, data will be used in order to test the program for error free running. The data will be that supplied by KPL. Other data may be contrived to fit logical extensions of the KPL system.

When it has been determined that the program is producing error free results, the results of this program will be compared to those found with the EPRI program (as used by KPL). If they prove better or essentially the same then the newly developed program use can be justified by KPL personnel.

Table 3.1: KPL Generator Listing

Number	Unit	Unit Type	Fuel
1	Jec 1	steam generator	coal
2	Jec 2	steam generator	coal
3	Jec 3	steam generator	coal
4	Law 5	steam generator	coal
5	Tec 8	steam generator	coal
6	Law 4	steam generator	coal
7	Tec 7	steam generator	coal
8	Law 3	steam generator	coal
9	Hutch 4	steam generator	coal
10	Hutch 3	steam generator	coal
11	Hutch 2	steam generator	coal
12	Hutch 1	steam generator	coal
13	Mcp 2	steam generator	coal
14	Mcp ct 2	combustion turbine	gas
15	Hutcht 1	combustion turbine	gas
16	Hutcht 2	combustion turbine	gas
17	Hutcht 3	combustion turbine	gas
18	Abile ct	combustion turbine	gas
19	Mcp ct 1	combustion turbine	gas

Key: Jec = Jeffreys Energy Center
Law = Lawrence
Abile = Abilene
Hutch = Hutchinson
Mcp = McPherson
Tec = Tecumseh

Table 3.2: Data Input Types - Used and Not Used

Number	Data Input Type	Used (U)/ Not Used (NU)
1	Model Description	U
2	Generating Unit Identification	U
3	Generating Unit Performance Characteristics	U
4	Generating Unit Cost	U
5	Interchange	NU
6	Initial Condition	U
7	Manual Schedule	U
8	Load Model	U
9	Load	NU
10	B Constant	NU
11	B Constant	NU
12	Title Data	NU
13	Plant Identification	U
14	Plant Fuel	U
15	Fuel Identification	U
16	Generating Unit Power Limits	NU
17	Generating Unit Fuel	NU

Table 3.3: KPL (Real) Data (From Reference 26)

STANDARD INPUT FILE											
1											
100	1	6.		22.			84.	24.	6.	27.	84.
101		27.		1.							
102	1	2.		3.			1.	1.	1.	1.	1.
103		1660.		1680.			1700.	1710.	1720.	1720.	1750.
104		1.		50.			50.		4.	4.	
105							-10.				
106		1.		10.			20.		4.	4.	
107		1.		10.			20.		4.	4.	
108				0.				0.			
109		20.							19.		
15											
15	1	Coal		2	Gas			3	Oil		
13											
13	1	Jeffery		2	Lawrence			3	Tecumseh	4	Hutch
13	5	Abilene		6	McPhersn						
2											
2	12	Jec 1	1		1	19	1			395.	165.
2	13	Jec 2	1		1	18	1			370.	165.
2	14	Jec 3	1		1	17	1			395.	165.
2	19	Law 5	1		2	16	1			270.	120.
2	23	Tec 8	1		3	15	1			110.	40.
2	18	Law 4	1		2	14	1			55.	5.
2	22	Tec 7	1		3	13	1			65.	20.
2	17	Law 3	1		2	12	1			45.	20.
2	39	Hutch 4	1		4	11	2			140.	55.
2	6	Hutch 3	1		4	10	2			30.	15.
2	5	Hutch 2	1		4	9	2			19.	10.
2	4	Hutch 1	1		4	8	2			19.	10.
2	26	MCP 2	1		6	7	2			25.	15.
2	28	MCP CT 2	2		6	6	2			45.	20.
2	8	HUTCHT 1	2		4	5	2			45.	20.
2	9	HUTCHT 2	2		4	4	2			45.	20.
2	10	HUTCHT 3	2		4	3	2			45.	20.
2	3	ABILE CT	2		5	2	2			65.	25.
2	27	MCP CT 1	2		6	1	2			45.	20.
3											
3	12	513.6615		8.980591			.00184143	72.	72.		
3	13	513.6615		8.980591			.00184143	72.	72.		
3	14	513.6615		8.980591			.00184143	72.	72.		
3	19	550.9698		7.223485			.00522346	72.	72.		
3	18	169.4305		8.224950			.01061296	12.	24.		
3	17	78.4038		9.018903			.01464727	12.	24.		
3	23	201.4463		7.187002			.01729802	12.	24.		
3	22	110.3347		8.785682			.01132096	12.	24.		
3	39	152.6788		8.449719			.00859942	12.	24.		

Table 3.3: KPL (Real) Data (Cont.)

3	6	32.2580	11.906040	.02423113	6.	18.
3	4	15.5141	12.120150	.05848686	6.	18.
3	5	15.5141	12.120150	.05848686	6.	18.
3	3	232.0997	8.453138	.00429833	1.	12.
3	8	161.6832	9.654569	-.0039202	1.	12.
3	9	161.6832	9.654569	-.0039202	1.	12.
3	10	161.6832	9.654569	-.0039202	1.	12.
3	26	34.6990	9.030818	.08784360	6.	18.
3	27	161.6832	9.654569	-.0039202	1.	12.
3	28	161.6832	9.654569	-.0039202	1.	12.
4						
4	12	44000.	72.			
4	13	44000.	72.			
4	14	44000.	72.			
4	17	3000.	24.			
4	18	4500.	24.			
4	19	20000.	72.			
4	22	2500.	24.			
4	23	6500.	24.			
4	39	6000.	24.			
4	3	400.	12.			
4	4	700.	18.			
4	5	700.	18.			
4	6	700.	18.			
4	8	400.	12.			
4	9	400.	12.			
4	10	400.	12.			
4	26	400.	18.			
4	27	400.	12.			
4	28	400.	12.			
6						
6	12	80				
6	13	80				
6	14	80				
6	17	80				
6	18	80				
6	19	80				
6	22	80				
6	23	80				
6	39	-80				
6	4	-50				
6	5	-50				
6	6	-50				
6	3	-20				
6	8	-20				
6	9	-20				
6	10	-20				
6	26	-50				
6	27	-20				
6	28	-20				

Table 3.3: KPL (Real) Data (Cont.)

7							
7	12			1			
7	13			1			
7	14			1			
7	17			1			
7	18			1			
7	19			1			
7	23			1			
7	22			1			
7	26			-1			
7	4			-1			
7	5			-1			
7	6			-1			
8							
8	1						
.6040	.5775	.5510	.5185	.4860	.5530	.6200	.7030
.7860	.8365	.8870	.9135	.9400	.9615	.9830	.9915
1.0000	.9740	.9480	.9250	.9020	.8415	.7810	.6925
8	2						
.5890	.5610	.5330	.5240	.5150	.5280	.5410	.6015
.6620	.7010	.7400	.7485	.7570	.7630	.7690	.7830
.7970	.7885	.7800	.7610	.7420	.7030	.6640	.6265
8	3						
.5570	.5310	.5050	.4980	.4910	.4935	.4960	.5415
.5870	.6245	.6620	.6825	.7030	.7085	.7140	.7280
.7420	.7405	.7390	.7400	.7410	.7035	.6660	.6115
14							
14	1	1	1.45	1.45	16	5	
14	2	1	2.20	2.20	41	2	
14	3	1	2.22	2.22	11	4	
14	4	2	2.90	2.90		2	
14	5	2	2.99	2.99	35	1	
14	6	2	2.90	2.90	39	1	

4.0 Constrained Lambda Dispatch: Code Development and Discussion

This chapter will be divided into two sections. The first section will be the statement of underlying assumptions used throughout the development of the constrained lambda dispatch (CLD) program. The second section will be the presentation and explanation of the specific code and algorithmic process of the computer program. A logic diagram will be provided also in this chapter (Fig. 4.1). A complete program listing is provided in Appendix 2.

4.1 Assumption Listing

Critical to the program development was the knowledge and understanding of specific criteria and assumptions that KPL works with in dispatching generating units. There are five assumptions, listed below, which were used. Any necessary explanations of these assumptions are also supplied in this list.

1) There is no transmission loss which need be considered independently of the system power demand. KPL currently assumes that all system transmission losses would be approximately eight percent of the actual power demanded. Hence, instead of producing enough power to meet 100% of actual power demanded, enough power is produced to meet 108% of the actual power demanded. For example, if 1000 MW is the total actual system demand then KPL would need to generate 1080 MW of electricity to meet this demand and to account for real transmission losses.
2) All system units are considered as one bus to meet one demand. Generally, one generating unit constitutes one bus. However,

rather than deal with 14 buses and the complications associated with a multi-bus system, KPL assumes that there is only one bus composed of their 14 generating units.

3) Generating units never are completely shut down. It is generally expensive to start up a unit which is not running. It is also difficult on the wear-and-tear of a unit. So the number of times that this task is actually undertaken is minimal. This being the case, it is simply assumed that a unit is never started from 0 MW and start-up costs are not a factor.

4) Combustion turbines are not used in this program. The calculations show that combustion turbines, with their inherent heat rate terms, produce negative lambda values. These results indicate that the optimal level at which to dispatch a combustion turbine is always at its maximum power level. For this and the additional reason that any one combustion turbine is generally never run for a long period of time to meet a demand, combustion turbines are not considered in this work.

5) All units must operate within their minimum and maximum power limits.

4.2 Computer Algorithm and Code Explanation

This section is divided into 24 different areas. Seven of these deal with subroutines found in this program while the remaining 17 areas will be independent sections of the program. These sections of the program consist of groups of statements that serve a common purpose.

Each one of these areas will be shown in the logic diagram (see Fig. 4.1), and briefly explained to obtain an understanding of the program logic. The order of area discussions will follow the program code as it is presented in Appendix 2.

Several references are made throughout the discussion about values that are printed out. This printing is performed only at the users discretion.

4.2.1 Section 1: (Lines 10 - 190)

This section is the definition of terms and variables. Its purpose is to aid the user in understanding the specific purposes of any variable used in the program.

4.2.2 Section 2: (Lines 200 - 270)

This section is devoted to dimensioning all dimensionable variables. This is an initializing stage of the program which only needs to be performed once during any specific case-run.

4.2.3 Section 3: (Lines 280 - 370)

The purpose of this section is to initialize every variable used in the program. This is done to assure the value of any variable upon its initial use.

4.2.4 Section 4: (A: Lines 380 - 460; B: Lines 680 - 850)

This section is divided by data input section 5A into two parts. In part A, the number of system dispatchable units is established and printed out. Also, the data supplied in Section 5A are read in.

In Part B, the data read in part A are displayed and the user is asked to verify the data. This allows the user to change the data points without running the entire program with erroneous data.

4.2.5 Section 5: (A: Lines 480 - 670; B: Lines 1180 - 1510)

This section is divided into two parts. Part A is a listing of the data points as supplied by KPL. They include the minimum power level, maximum power level, the α, β, and γ coefficients for the heat rate equation, i.e.,

$$HTRT(i) = \alpha_i + \beta_i P_i + \gamma_i P_i, \tag{1}$$

where i is the specific unit and P is the unit power level.

In Part B the must-run units and their respective must-run power settings are established and printed out for user verification. When inputting the must run unit numbers and power levels it is extremely important to have a comma to separate each unit value from the preceding power level and every power level from its associated unit number. Even if the values are zero, the commas must be in place. (There should be 27 commas for every data entry line.) In addition, the lambda setting for every system dispatchable unit is established for future reference.

4.2.6 Section 6: (Lines 860 - 1170)

This section has several calculations performed in it which are critical to the performance of the entire program. First, the unit efficiency rate is established. This is followed immediately by the calculation for the maximum and minimum lambda values for each unit as dictated by its inherent heat rate coefficients and minimum and maximum power constraints. By the users discretion, these values are printed out.

Following this sequence, the number of hours for which the program will be run and the peak demand for that day are inputted. Normalizing factors exist in this section so that specific hourly demands can be calculated, if desired. The normalizing factors can be set equal to one so as to allow no change of the inputted system demand value.

4.2.7 Subroutine 1: (Lines 1530 - 2300)

In this subroutine the lambda values for the CLD are calculated. It is divided into four basic sections. Each one will be presented individually.

Part A: (Lines 1600 - 1740)

In this part the product of each units' quadratic term of the heat rate equation and fuel cost are calculated and summed over the set of candidate (dispatchable) units. This term (GPRD) is especially important for the function of parts B and D.

Part B: (Lines 1750 - 1900)

In this part of the subroutine the denominator of the lambda value is calculated. The calculation is performed by taking the GPRD value calculated in part A and dividing it by the product of the individual gamma and fuel cost for each unit. This value is termed GTRM(i). The GTRM values are summed over all candidate units, which equals the denominator value termed DEN. This DEN term is used specifically in part D.

The next series of statements (lines 1910 - 2050) was written to provide for the situation that the algorithm might reach this point and have no dispatchable units that can be used to supply the power to meet demand. The first FOR-NEXT loop (lines 1940-1980) are designed to find the least expensive dispatchable unit (minimum lambda) and to keep a record of this unit and of any other equally inexpensive unit with the variable LMST. The next FOR-NEXT loop (lines 2000-2030) is designed to determine, between units that are seemingly equally inexpensive, which unit will be dispatched first. These steps are completed by returning to the beginning of the subroutine.

Part C: (Lines 2060 - 2180)

In this part of the subroutine the calculation takes place for the second term (STRM) of the numerator for the system lambda equation. This calculation is done by multiplying the GTRM(i) value, described in part B, the linear term of the heat rate equation (Bet(i): given data), and the fuel cost (Cst(i): given data) for each unit. These individual values are summed over all candidate units which equals the value for STRM. This term is used specifically in part D.

Part D: (Lines 2190 - 2290)

This part of the subroutine takes the previously explained variable values (GPRD - part A; DEN - part B; STRM - part C) and the given power demand value (PDMD) to calculate the system incremental cost for the next unit of power (LAMBVAL(j)). This calculation is performed by multiplying twice the demand and GPRD, adding the product to STRM, and then dividing the sum by DEN. This gives the system lambda value used in further analysis. This completes the process of subroutine 1.

4.2.8 Section 7: (Lines 2310 - 2410)

These are the initial steps of the program logic. Initially ordering the units by minimum lambda values in ascending order is done by going to subroutine 6 (line 5910) which will be described in more detail later. Next, a marker is given a value indicating the process has passed this point followed by setting all candidate unit power values equal to zero for the "free-run" lambda dispatch.

4.2.9 Section 8: (Lines 2430 - 2740)

In this section the process of determining the maximum and minimum system production levels is completed. First, the minimum power level of all the dispatchable units is found (MNMN). If any unit is found to be a must-run unit (UNUSD(I,J) $\neq$ 0), the must-run power level (PLUSD(I,J)) is the value for MNMN. If several units are must-run units, the sum of the must-run power levels is the MNMN value. A marker is set indicating the process has been to this point (TRK=1) and the maximum possible system production level (PTOTMX) is calculated by summing all the maximum production levels (PMX(I)) of each individual unit.

Comparisons are made between the power demand and MNMN as well as the power demand and PTOTMX. If the power demand is less than MNMN, the unit with the lowest minimum power setting is set to that value or all the must-run units are set to their must-run production levels and a message stating that power must be sold is printed completing the case run. If the power demand exceeds the PTOTMX value all units are set at maximum and a message is printed that power must be purchased. This completes the case run. When the power demand lies between MNMN and PTOTMX, CLD is to be used (Section 9).

4.2.10 Section 9: (Lines 2750 - 2860)

This section reinitializes unit power settings to 0 or to the must-run levels when it is determined that CLD is to be used. The total system demand is also reset and renamed the original power demand (PDORIG). The CLD process proceeds from here.

4.2.11 Subroutine 2: (Lines 2870 - 2980)

In this subroutine new power production values and unit lambda value calculations are performed for candidate units. First, the candidate value is checked (CAND(I) = 0: candidate unit). If this unit is not a candidate, the next unit is brought on and checked. If the unit is a candidate the next unit is brought on and checked. If the unit is a candidate the system lambda value, subroutine 1: part D, is used together with its BET(i), CST(i), and GAM(i) coefficients to calculate the power production settings (P(I,J)). This is followed by calculating the lambda value (UNLVL(i)) by summing the products of BET(i) and CST(i) with the product of twice CST(i), GAM(i), and P(I,J). Candidate values are set to two, which indicates the possibility for future dispatch, if the P(I,J) value lies below the minimum power setting or must-run setting for the unit. These calculated values are then printed out and the algorithm proceeds to the next subroutine (subroutine 3).

4.2.12 Subroutine 3: (Lines 3010 - 3170)

In this subroutine the power settings, calculated in subroutine 2, are checked and reset when necessary. The first logic step is to compare the power setting, P(I,J), to the power maximum, PMX(i), for each unit. When PMX(i) is equaled or exceeded, that value is subtracted from the power demanded, PDMD, P(I,J) is set equal to PMX(i), and the candidate value is set to one.

If P(I,J) is less than PMX(i) then P(I,J) is compared to the minimum power level, PMN(i). If P(I,J) is less than PMN(i) the power setting is zero and the candidate setting is two. If P(I,J) is greater than PMN(i) then P(I,J) is compared to the must-run power value of the unit, PLUSD(I,J), which is zero if the unit is not a must-run unit. P(I,J) is set equal to PLUSD(I,J) if P(I,J) is less than or equal to PLUSD(I,J). The algorithm then proceeds to subroutine 4.

4.2.13 Subroutine 4: (Lines 3200 - 3340)

In this subroutine the total production of all candidate units is summed. The variable assigned to this value is PVAL. PVAL is first reset to zero and then all P(I,J) values are summed, which is the new PVAL value. The power demanded is then reset by subtracting PVAL from the original demand (PDORIG). System characteristics as well as specific unit characteristics are printed out and then the difference between PVAL and PDORIG is evaluated by subtracting PDORIG from PVAL. The variable assigned to this value is EVAL. This variable's value is used in testing conditions immediately following this subroutine. Subroutine 4 has now been completed.

The next series of statements (lines 3350-3450) tests the EVAL value to determine the algorithmic procedure to be followed. If EVAL is greater than five, the procedure continues with Section 11. If EVAL is less than negative five, the procedure continues with Section 10. If EVAL is equal to or in-between five and negative five, then EVAL is tested to determine whether redispatching is necessary. If EVAL lies between or is equal to negative one and/or one, then the case run is completed. If this is not the case, the candidate unit(s) is (are) found and resetting of respective P(I,J) values is performed. This is followed by redispatching which then completes the case run.

4.2.14 Section 10: (Lines 3470 - 3570)

This section determines whether the system lambda lies outside the minimum to maximum region of every unit. If this is the case, variable XX is set to zero and the process continues with subroutine 5. If this is not the case, XX is set to any value not equal to zero and the process continues with Section 11.

4.2.15 Section 11: (Lines 3580-3850)

This section determines whether units that are must-run units have a minimum lambda value that exceeds the value of the system lambda. If this is the case and the must-run power level is greater than the demand, it is known that units on at maximum capacity must have their generation level lowered. This section continues by appropriately assigning candidate values and P(I,J) values, readjusting the PDMD value, printing out the unit characteristics, and redispatching (X=2). This process continues with Section 12.

4.2.16 Section 12: (Lines 3880 - 4060)

This section works in conjunction with Section 11 in the manner that after redispatch is completed the units on at maximum and must-run units are found and the power demanded is readjusted. The unit and system characteristics are printed and the process continues by redispatching, if noted as necessary in Section 11, or by directly proceeding to subroutine 5.

4.2.17 Subroutine 5: (Lines 4120 - 4890)

The purpose of this subroutine is to recheck whether the system lambda value lies in a region that is not covered by any maximum to minimum lambda area of any unit, called the forbidden lambda zone. This scenario is forbidden so appropriate action must be taken. The appropriate action in this subroutine is divided into four parts, to be explained individually.

Part A: (Lines 4130 - 4320)

This part is only to determine the units that can be dispatched by selecting the unit with the minimum lambda value not set at maximum power (MOCMN). This process is as follows. Every dispatchable unit's minimum lambda (CAND(i) $\neq$ 1) is compared to the MOCMN value, which is initialized at a value of 100. If the value compared to MOCMN is smaller, then MOCMN takes on the lesser value. After this process is completed the chosen unit is printed out and its candidate value is three. This is so that, if necessary (part B), this process can be redone before redispatching and this unit will still be a candidate unit selected for the redispatch. The process continues with part B.

Part B: (Lines 4330 - 4490)

This part begins by initializing the variable MNTOT, the total of the unit power minima for all the units selected for redispatch. MNTOT is incremented by the minimum power level values of these selected units. When all have been considered the MNTOT value is compared to the power demand. If MNTOT is less than the power demand, MNTOT is reset to zero and the unit with the next lowest minimum lambda becomes a member of the selected units. This process is continued until either all units have been used and PDMD still exceeds MNTOT (subroutine 7), or MNTOT equals or exceeds PDMD (part C).

The situation in which all the units have been selected yet the PDMD has not been reached is signified by the variable NOCAND equaling zero. All units are checked for their CAND(i) values. If they are all one or three then NOCAND equals zero. If NOCAND does not equal zero then the process continues with part C.

Part C: (Lines 4500 - 4670)

This part determines what to do when only one unit's PMN(i) value is enough to exceed the PDMD, i.e., only one selected unit is necessary to meet demand. It begins by initializing a marking variable, TRKR, and a variable used in part D, MOCMN, to zero. Then the test is performed to make sure only one unit has been selected for redispatch. If TRKR equals one this situation holds true. PDMD is reset to PDORIG and the entire logic is started by going back to Section 7. If TRKR is not equal to one then this situation does not hold true. Hence, the process continues with part D.

Part D: (Lines 4690 - 4880)

This part is where the calculations are performed when PDMD still exceeds MNTOT but all units have been selected. The situation must be looked at with respect to maximum power values, PMX(i). The unit with the lowest PMX(i) is selected first and PMS(i) is set equal to MOCMX. MOCMX is compared to PDMD and, if PDMD is exceeded or equaled, this unit is selected as the candidate unit for redispatching. If MOCMX is less than PDMD then the unit with the next lowest PMX(i) value is selected. MOCMX is incremented by this value and compared again to PDMD. This process is continued until PDMD is equaled or exceeded, at which point redispatch is performed. This concludes the use of subroutine 5.

4.2.18 Section 13: (Lines 4900 - 5060)

This section begins by redispatching and recalculating the P(I,J) value for each unit (subroutine 1, subroutine 2). Unit numbers and associated candidate values are then printed out and a marking variable (MRK) and a variable used in Section 14 (MXMX) are initialized to zero. Then a process is followed to determine if a P(I,J) value which lies between zero and PMN(i) has been calculated for any unit. If it has then MRK equals one and the process continues by going to subroutine 3. If this is not the situation then the process continues by going to Section 15.

4.2.19 Section 14: (Lines 5070 - 5330)

This section is for commenting purposes only. Even though the last five statements are functional, they are exactly the same ones used in Section 13. Hence, no logic explanation is required for this section.

4.2.20 Section 15: (Lines 5340 - 5660) - [Section 16: Imbedded]

This is where the last selected unit that makes MOCMN exceed PDMD is taken off the selected unit list. The first thing that is done is the selected candidate unit with the highest PMN(i) is found and marked with the variable UNLVL. The process continues in Section 16, an imbedded section.

After completing Section 16, the value of MRK is tested. If it is zero the process continues by going to subroutine 3. If it is not equal to zero, then another marker variable, THRU - which indicates the process, having reached this point, is set equal to five. This process is then redone starting from Section 13. However, when Section 15 is reached again the process goes directly to subroutine 3 because of the new THRU value.

This section is completed with a series of statements that do nothing more than check that P(I,J) values are at allowable levels. The algorithm continues with the printing section, Section 17. Discussion continues with Section 16.

4.2.21 <u>Section 16</u>: (Lines 5410 - 5540)

This series of steps tests whether a must-run unit has its respective minimum lambda value exceeded without increasing its P(I,J) value. If this is the case then another redispatching should be done with this unit considered a candidate. The variable HELP(i) is introduced to help the necessary units be recognized that are overlooked in the previous dispatches. When HELP(i) equals one the unit i should be a candidate unit and redispatch should be performed. When HELP(i) does not equal one unit i is not a candidate. If HELP(i) does not equal one for any unit then redispatching need not be done. The process continues by returning to Section 15. Discussion continues with Section 17.

4.2.22 <u>Section 17</u>: (Lines 5760 - 6220)

This section is where two things happen. First, the heat rates and operating costs are calculated. Then, all results compared thus far are printed out in table form. When this particular section has been reached the entire case run has been completed.

4.2.23 <u>Subroutine 6</u>: (Lines 6230 - 6470)

This subroutine is where the units are ordered by their minimum lambda value and subsequently printed out. This process introduces the use of five new variables, K: an incrementing variable, ORDR(K): the minimum lambda value for the Kth cycle, UNT(K): the unit number selected for the Kth cycle, TKN: indicates a unit already selected, and MNCAND1: a variable used to store the value of the selected minimum lambda values.

To start the process K is set to one, UNT(K) and TKN to zero, and ORDR(K) and MNCAND1 to 100. The minimum lambda values for each unit (LAMBMN(i)) is compared to ORDR(K). Every time a value less than ORDR(K) is found ORDR(K) takes on that value. When the lowest LAMBMN(i) value is found it is stored in ORDR(K) and the respective unit is given a candidate value of four. This prohibits this unit from being selected again. This process is followed until all the units have been ordered and is concluded when the units are all printed.

4.2.24 Subroutine 7: (Lines 6480 - 8010)

The purpose of this subroutine is to handle the situation where no candidate units were found in subroutine 5, part B. This subroutine is also always preceded by the use of subroutine 6. This subroutine begins with documentation and variable initialization or resetting. The new variables introduced are CRUISE, FRSTRN, and EINMAL. They are all marker variables and are all set to zero. This subroutine is divided into 13 separate parts. Each will be presented individually.

Part A: (Lines 6630 - 6700)

This part is where the initial unit is selected for comparison in the following parts. If it is the first run of this process, the units minimum lambda value is less than the stored LAMBMN value, or the candidate of the selected unit (TKN) is three or one then the process will go to the next unit on the list estsblished in subroutine 6 as the comparative unit. The original power demand is also set to a dummy variable so that it may change values yet have its old value recalled.

Part B: (Lines 6720 - 6810)

This part is used solely to reset the power demand value so that redispatch will be properly performed. Because this part may be reached without the proper power setting being calculated the power demand is reset by adding the must-run power setting of every unit and then subtracting the actual power setting.

Part C: (Lines 6820 - 6870)

This part is where all the unit power settings are stored in another arrayed variable (USET(I,J). This way values can change yet be recalled for later processes.

Part D: (Lines 6880 - 7070)

This part is where the LAMBMN(i) value of the unit i that was the last unit selected as being a possible candidate for redispatching, to calculate the power settings of the units already selected as candidate units. The sum is taken of the PMN(i) value of the comparison unit, and the derived P(I,J) values of the other candidate units after the derived P(I,J) values have been checked so as not to exceed the unit's maximum and minimum power levels. After this has been completed the sum is subtracted from the incremental power demand (PDMDDMY). If the value of PDMDDMY is in the range of one to negative one then the case run is complete. If PDMDDMY is greater than one then the preceding process is followed again by going back to the beginning of part D. If PDMDDMY is less than negative one then the algorithm continues with Part E.

Part E: (Lines 7080 - 7190)

This part is where it is determined whether all selected units are set at maximum, yet redispatching needs to be performed because PDMD has not been met. This process is started by initializing variable YES to one, MC to the candidate value of the compared unit, MP to the power value of the compared unit, the candidate value of the compared unit, CAND(TAKN) to zero, and the power value of the compared unit, P(TAKN,J) to .001.

If any unit is set at less than its power maximum then the process continues by proceeding to Part F. This is indicated by the variable YES being decremented to zero. If all candidate units equal their maximum power level then YES retains its value of one and the process goes to Part K.

Part F: (Lines 7200 - 7260)

This part is used to reset candidate unit power levels to their previous levels when it is determined that these are the desired quantities.

Part G: (Lines 7270 - 7320)

This part is used to reset the incremental power demand when the situation stated in Part F holds true.

Part H: (Lines 7330 - 7400)

This part prints out the independent unit characteristics when the candidate units have been determined for redispatching.

Part I: (Lines 7410 - 7510)

This series of statements has no bearing on the logic followed by this program. Hence, no explanation will be given except to say that these lines are comment statements.

Part J: (Lines 7520 - 7610)

When this series of statements is reached, the case run is completed for all practical purposes. This is indicated by variable values, i.e., I = UQNT, Z = UQNT, and PDMDDMY, being reset to PDMD. The only lines which really have a bearing on the logic flow are the last five.

Part K: (Lines 7620 - 7810)

This is where the candidate values and power values are set to dummy variables CSET1(i) and USET1(i) respectively. If the pre-established value of YES (Part E) is zero or the value for PDMDDMY is equal to or greater than PMN(TAKN) then resetting of the variables is done without any further action. If these two conditions are not true then resetting of several other variables takes place before the resetting of values stated initially. These resettings are listed on lines 7660 and 7670 of the program (Appendix 2).

After resetting these variable values, the situation is tested as to whether further checking for candidate units is necessary. If YES equals one then the process is redone starting with Part F. If YES is zero then the process continues in Part C.

Part L: (Lines 7820 - 7860)

No further checking for candidate units is necessary when this part is reached. The power values are reset and the process continues into Part M.

Part M: (Lines 7870 - 8010)

This is where the values for the candidate units selected for redispatch are set for the actual redispatching process. It is a checking process making sure that the values have been properly set. If they have not been, they are readjusted accordingly. These values are printed out, if desired, so that the user can verify their settings. This completes the use of subroutine 7.

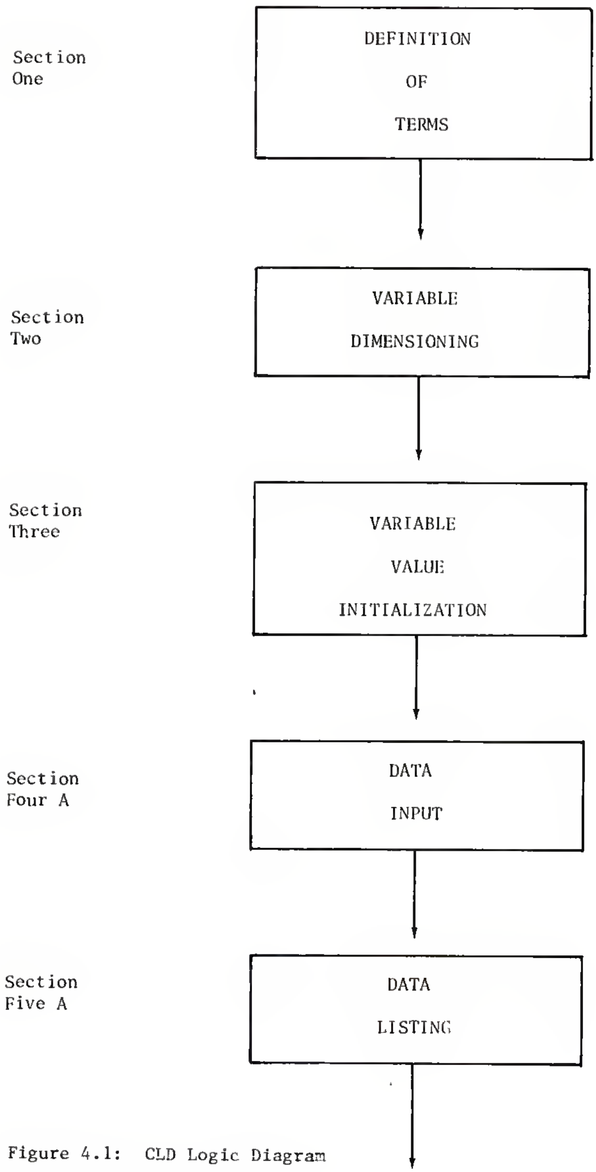

Figure 4.1: CLD Logic Diagram

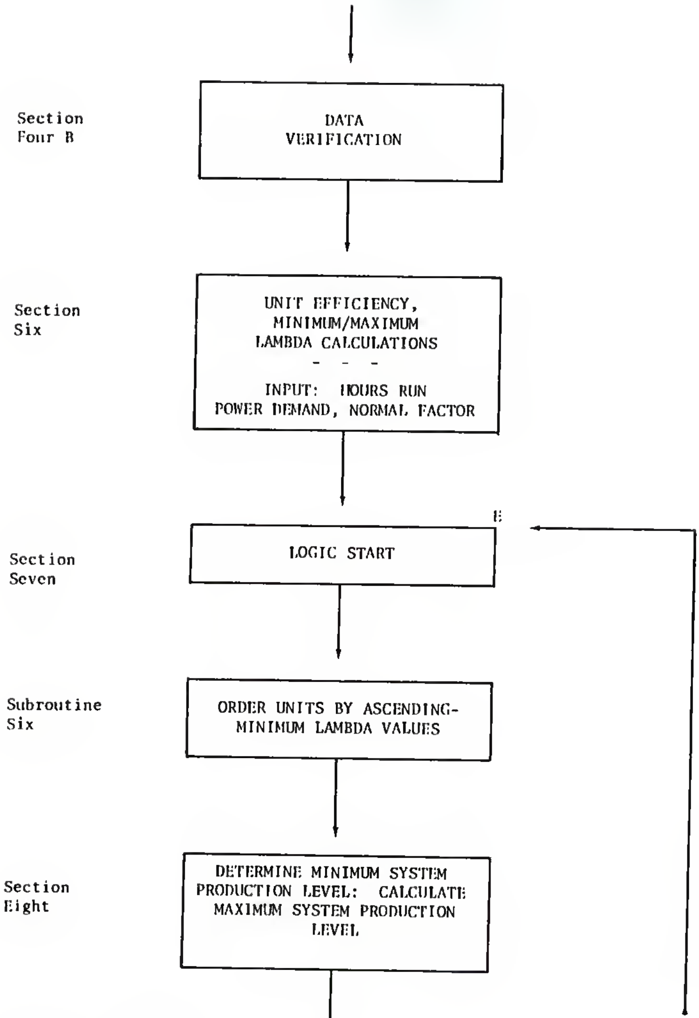

Figure 4.1 (Continued)

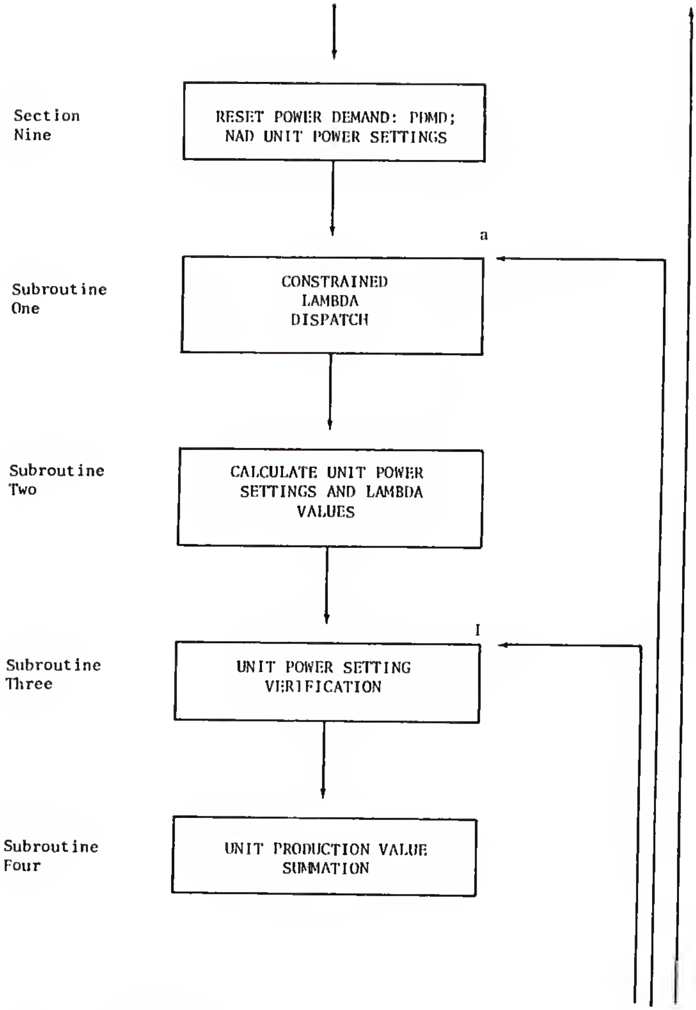

Figure 4.1 (Continued)

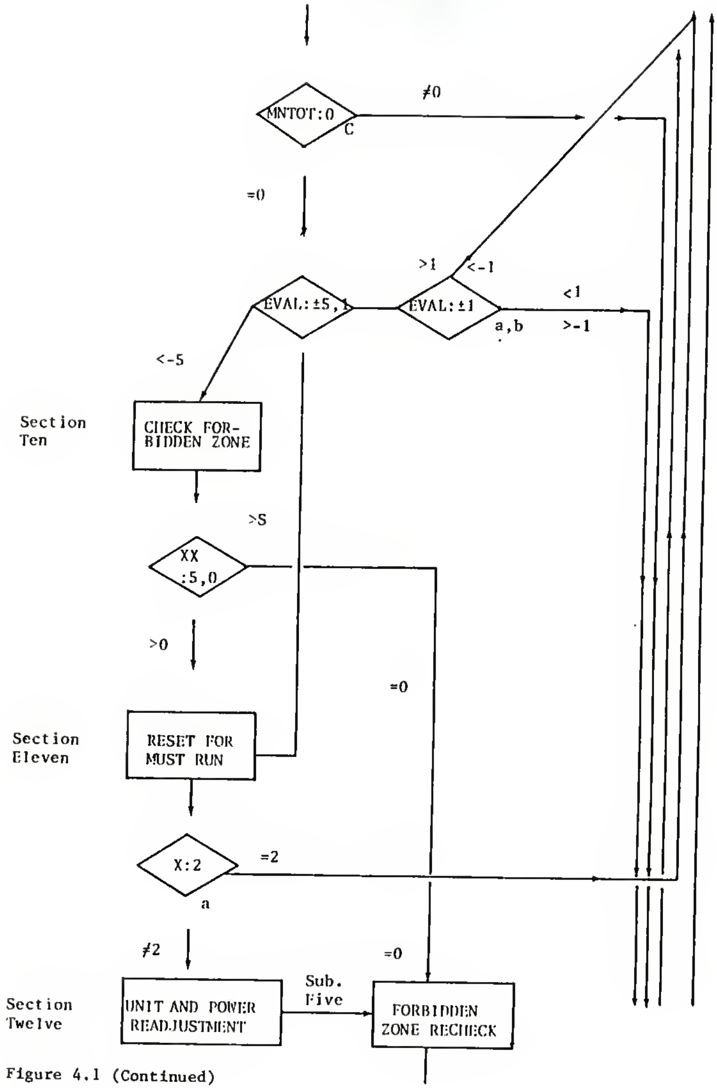

Figure 4.1 (Continued)

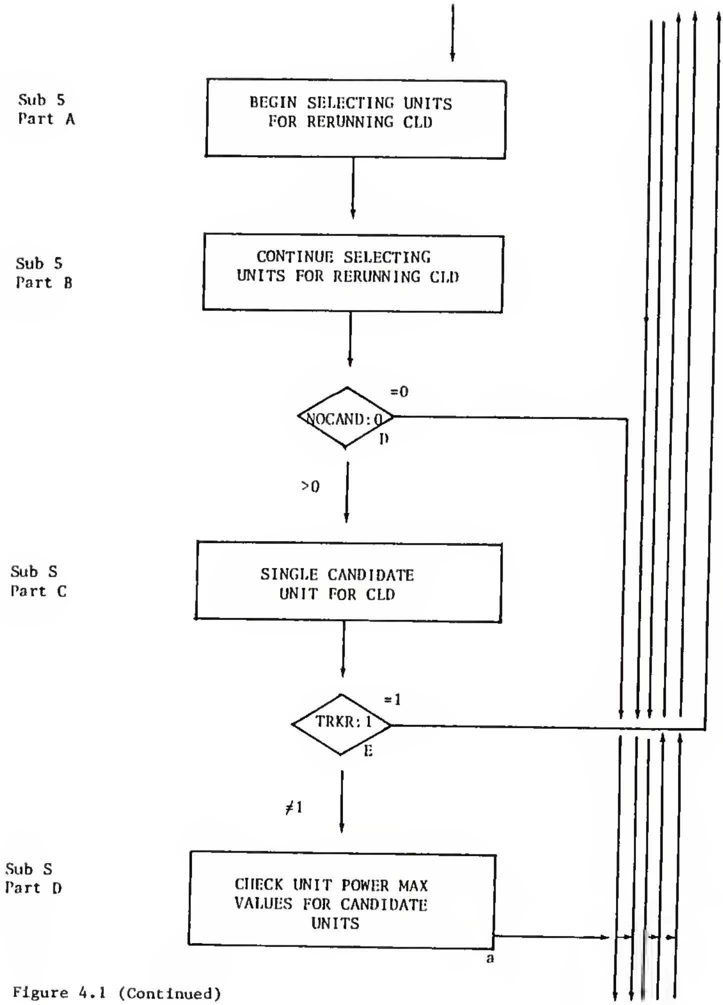

Figure 4.1 (Continued)

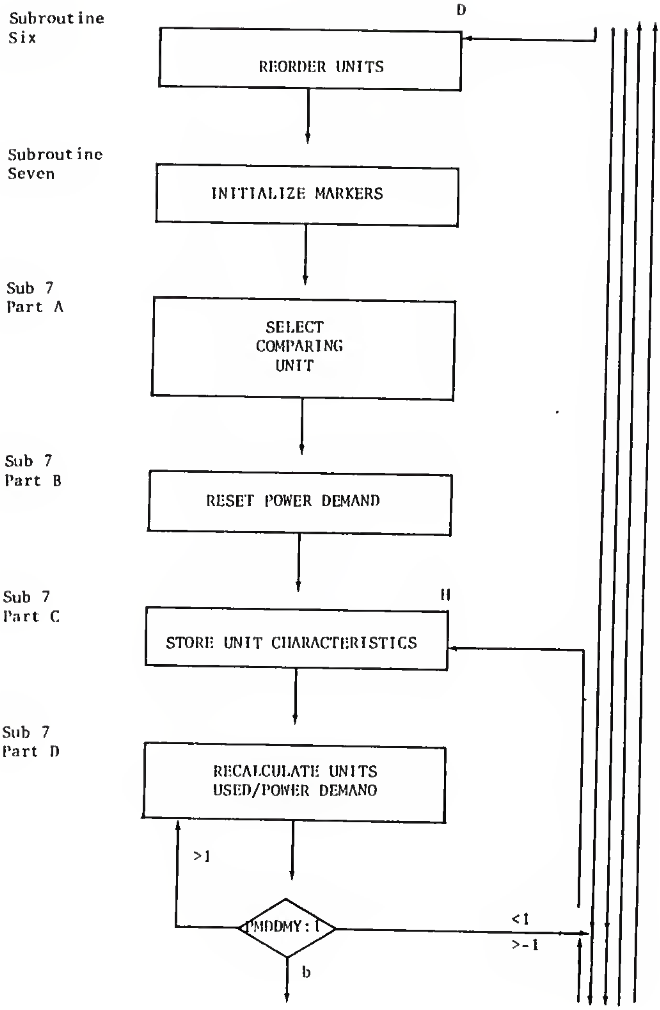

Figure 4.1 (Continued)

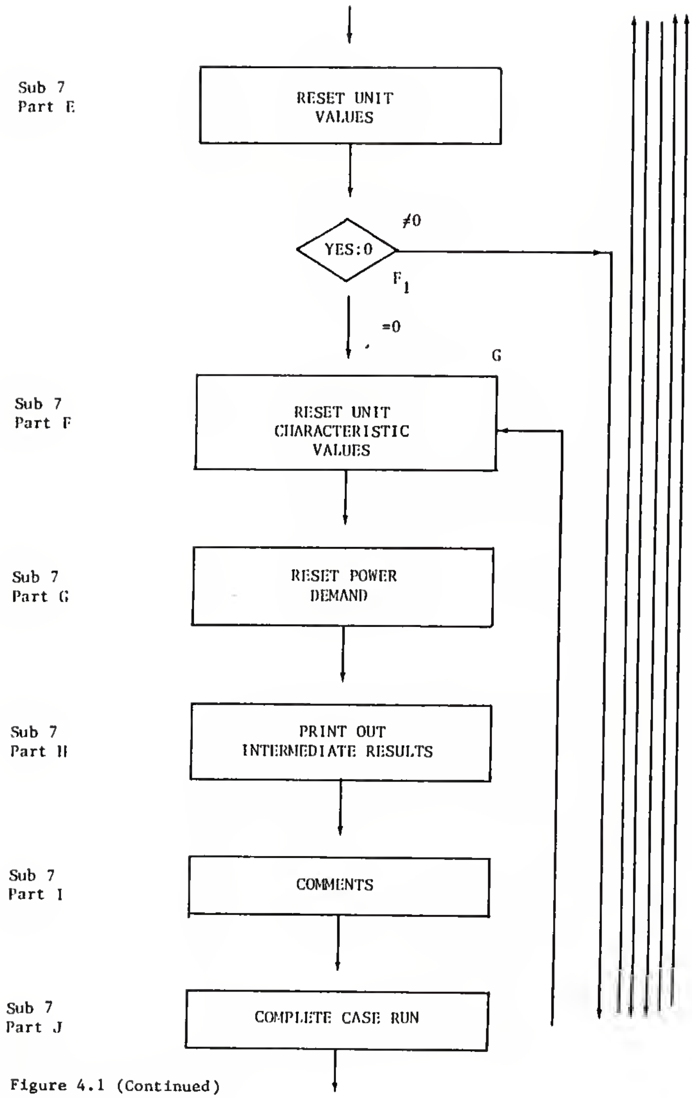

Figure 4.1 (Continued)

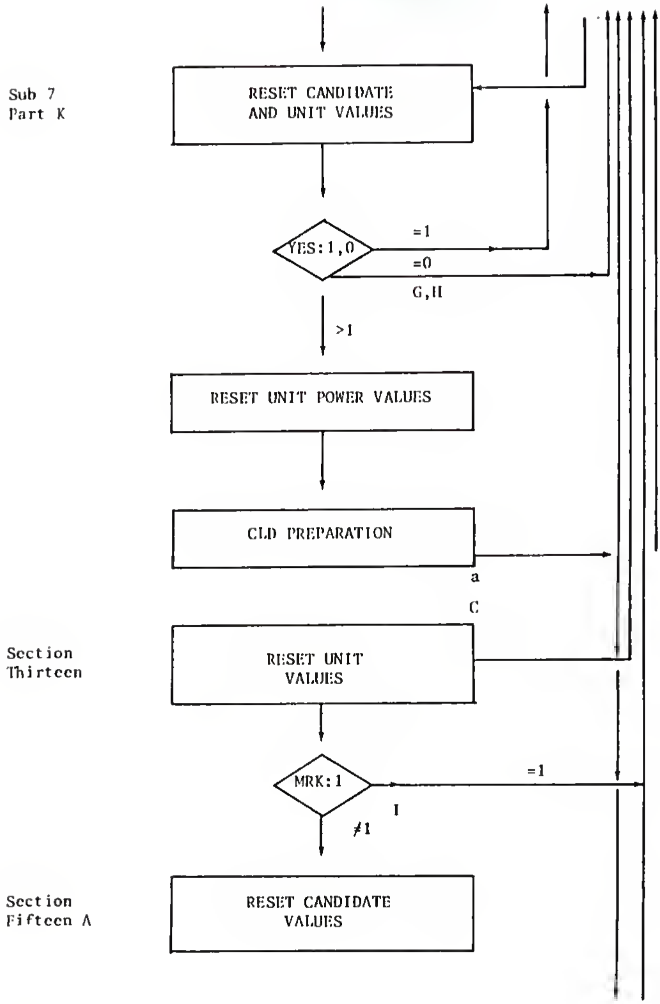

Figure 4.1 (Continued)

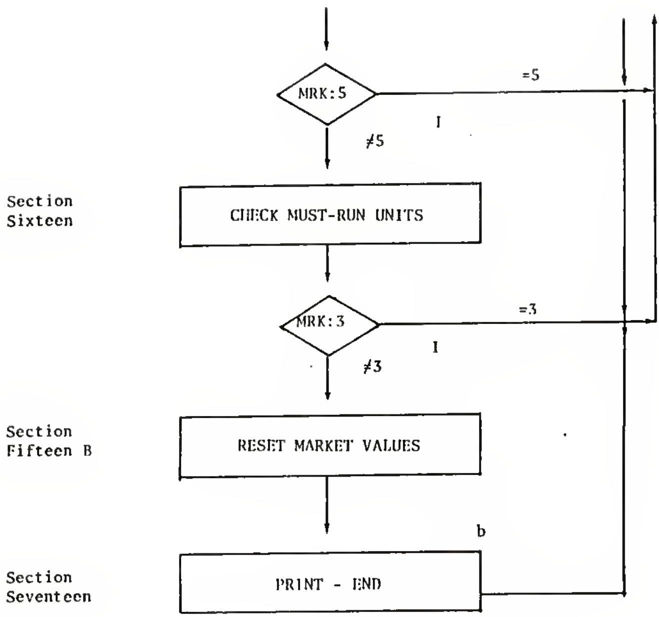

Figure 4.1 (Continued)

5.0 Comparison and Analysis of CLD Results

This chapter is a discussion of results. The discussion will be divided into three major categories. In the first, an explanation of the results of the lambda dispatch computer program used to compare generator settings to KPL data will be given. The second will extend this discussion to other cases not encountered in using KPL data. The third will be a comparative analysis of the lambda dispatch settings, KPL data, and EPRI settings. In explaining the processes that are followed for each of the separate cases, the program algorithm will be described.

5.1 Program Results Using KPL Data

The data shown in Tables 5.1 and 5.2 are generator settings used on January 1 and 2, 1985 by KPL. The data were supplied by Robert Fackler of KPL.[27] The readings are for each individual hour of the two day period. The total system demand is given by the hour as is the power production level of each of six units: Jeffrey 1, 2, and 3, Lawrence 4 and 5, and Tecumseh 7. Each of these units are must-run units for all 48 hours. This means that they must be on at least at a minimum production level during the entire period.

$$P_i = P_i^{min} \text{ for } i = 1, 2, \ldots, N, \tag{5.1}$$

where N is the number of units (6) and i is a specific unit.

There are four types of cases considered for each demand reading. This can be seen in the listing of results in Tables 5.3 through 5.10. These four different sets of results stem from: 1) an optimal free-run dispatch (Tables 5.3 and 5.4); a situation in which there are no must-run units, thus, the algorithm dispatches over the least expensive units until demand is met; 2) A lambda dispatch constrained by must-run units (described in the previous paragraph) (Tables 5.5 and 5.6); and 3) Actual readings (Tables 5.7 and 5.8). Hourly readings from the EPRI computer program were not obtained. However Tables 5.9 and 5.10 show the cumulative comparative values from each set of results (including EPRI) in terms of actual cost ($) and incremental cost ($/MWH)). These tables will be referred to extensively throughout this discussion.

The analysis will be performed in the following manner. Starting with hour one of January 1, the process followed by the CLD program to solve this problem will be explained. The final results are all shown in the comparative tables (Tables 5.9 and 5.10) . After the analysis for the first hour has been completed, the analysis for the second hour will be performed, etc. If the case of any particular hour being similar to any previously explained case, reference will be made to that previously explained case. The analysis will then continue with the discussion of the next hour's case. It must not be forgotten: all six units used must be on at least at a minimum power production level. These values are given in the Given section of each case. The common given data used in these cases is supplied in Table 5.11. A graphical illustration of these data points is supplied in Figure 5.1.

5.1 Case 1

Constrained Lambda Dispatch (CLD) with six must-run units, dispatch over the first three units.

Given: Jef 1 >= 165 MW   Jef 2 >= 165 MW

Jef 3 >= 165 MW   Law 4 >= 30 MW

Law 5 >= 120 MW   Tec 7 >= 20 MW

$P^{tot} = 1157$ MW

Conclusion: Jeffrey 1 produces 268.1 MW, Jeffrey 2 produces 323.44 MW, Jeffrey 3 produces 295.5 MW, Lawrence 4, 5, and Tecumseh 7 all produce at their respective minimums (given).

Discussion: Immediately after the input data have been established a free-run lambda dispatch is performed. As shown in Table 5.12, negative (or less than must-run) power values, which are not allowable, are calculated for units 4, 5 and 6. Since this is a situation that cannot exist, the program algorithm is directed to recognize which units have to be on at least at their minimum power output.

A new system incremental demand (SID) is calculated by subtracting all the must-run minimum power levels from the total system power demand.

$$P^{tot} - \sum_{i=1}^{N} P_i^{min} = SID \ , \qquad (5.2)$$

where P^{tot} is the total system demand, and $\sum_{i=1}^{N} P_i^{min}$ is the sum of all the must-run units which operate at a minimum power level.

The units which are dispatchable to meet that demand are now established as Jeffreys 1 through 3. These units are called candidate units,

$$\text{Cand}(i) = 0 \text{ for all candidate units} \qquad (5.3)$$

If these units are must-run units the respective power levels are set to zero and the SID is increased by their respective must-run levels.

$$P_i = 0 \text{ for all candidate units,} \qquad (5.4)$$

$$SID = SID + P_i^{min} \qquad (5.5)$$

where P_i^{min} is the power minima of the candidate units.

A lambda dispatch is recalculated with the new candidate units and SID. This iteration will give the exact power levels at which the candidate units should be set to meet the SID with the other must-run units being set at their minimum levels. The results are shown in Table 5.13.

Every hour during the first 48 hours of 1985 followed the same scenario except for hours 19 and 20 of January 2. These two hours are the bases for Cases 2 and 3.

5.3 Case 2

CLD with six must-run units and dispatching over 2 units leaving one must-run on at minimum and three Jeffrey units on at maximum power levels.

Given: Jef 1 >= 165 MW   Jef 2 >= 165 MW

Jef 3 >= 165 MW   Law 4 >= 30 MW

Law 5 >= 120 MW   Tec 7 >= 20 MW

$P^{tot} = 1457$ MW

Conclusion: Jeffreys 1 through 3 produce 405 MW each, Lawrence 5 produces 180.5 MW, Lawrence 4 produces 41.6 MW, and Tecumseh 7 produces 20 MW of energy.

Discussion: As was the situation with Case 1 the program algorithm performs a free-run lambda dispatch, determines that negative power production values are present for certain units, sets the units with negative values to zero or at the must-run level, and recalculates the SID. The candidate units are established. If they are must-run units, they are set at 0 MW production level, the SID is adjusted by adding the must-run production level to the present SID (Eqn. 5.5), and a lambda dispatch is performed again.

In this case, the power production maximum of every candidate unit is exceeded. If the power production maximum were exceeded for only a few of the candidate units the algorithm would recognize this and do the following. The candidate units for which the power maximum was exceeded would be set at the power maximum and this value would be subtracted from the SID, i.e.,

$$\text{when Cand(i)} = 0 \text{ and } P_i > P_i^{max}, \tag{5.6}$$

$$P_i = P_i^{max} \text{ and Cand(i)} = 1, \tag{5.7}$$

$$\text{SID} = \text{SID} - P_i^{max}. \tag{5.8}$$

The remaining candidate(s) would then be set to zero and a lambda dispatch would be repeated. However, since all candidate units' power maximum limits are exceeded the algorithm directs the computer program flow as follows.

First, every candidate unit is set to maximum power production and these values are subtracted from the SID (Eqns. 5.6 - 5.8). A search is now made for the unit with the lowest minimum lambda value for which the associated unit is not set at its maximum power level. Since the units have been set in an ascending priority order, according to the minimum lambda values, the next unit on the list is used. If it is a must-run unit, no further checking need be done. It is the new candidate unit. If it is not a must-run unit, then the minimum power production level must be compared to the SID. If the SID is less than the minimum power level then another unit with a higher minimum lambda, but which is either a must-run unit or for which the minimum power level is less than the SID, will be the new candidate unit. The completed cycle of using units that are not must-run units will be covered in later discussion of the program algorithm.

In this case the next unit considered was a must-run unit. Thus, the SID was increased by the minimum production level of the new candidate unit. The power level for this unit was set to zero and a lambda dispatch was executed again. However, even this is not a completed solution. In order to assure optimality one must check whether the lambda level at which the candidate unit is now set is not greater than the minimum lambda value of any unit that is generating power (but not at maximum power). If no minimum value is violated then the solution is complete. However, this was not true with this case.

In this case, the minimum lambda value of one must-run unit, not set at maximum power level, is exceeded by the lambda setting of the candidate unit. Since this is the case, this "exceeded value" unit now becomes a candidate unit in addition to the previous candidate unit, i.e.,

$$\text{when } \lambda^{SID} > \lambda_i^{min} \text{ and } 0 < P_i < P_i^{max}, \tag{5.9}$$

where λ^{SID} is the present system lambda.

$$\text{Then } Cand(i) = 0 : SID = SID + P_i ,$$
$$\text{and } P_i \text{ becomes zero.} \tag{5.10}$$

Both were set to zero MW production level and the SID was increased by adding on the minimum power generation level of the new candidate unit

(Eqn. 5.10). A lambda dispatch was performed and sum of the given power levels of the candidate units in addition to the units on at maximum and must-run levels will equal the demand. No further checking of whether lambda minima were violated was necessary because only one unit was exceeded in the previous run. The results are as shown in Table 5.14.

5.4 Case 3

CLD is done with six must-run units, dispatching over one unit (Lawrence 5), Jeffrey 1 through 3 are left on at maximum, and Lawrence 4 and Tecumseh 7 are left on at minimum.

Given: Jef 1 >= 165 MW　　Jef 2 >= 165 MW

Jef 3 >= 165 MW　　Law 4 >= 30 MW

Law 5 >= 120 MW　　Tec 7 >= 20 MW

$P^{tot} = 1399$ MW

Conclusion: Jeffrey 1 through 3 produces 405 MW each, Lawrence 5 produces 134 MW, Lawrence 4 produces 30 MW, and Tecumseh 7 produces 20 MW of power.

Discussion: This case is very similar to Case 2. The difference lies in the last two steps of completing the algorithm. To review the algorithmic procedure that led to the final two steps, the following listing is offered.

1) The given data are listed and necessary changes are made.
2) The free-run lambda dispatch was run. If any negative power values appear it is known that the units associated with these values are not candidate units.
3) Must-run candidates were designated, the SID was set, and lambda dispatch was run again.

4) The power production maximum of candidate units were exceeded. The candidate units were set to the maximum production level, new candidate unit(s) were found, SID was reset, and a lambda dispatch was run again.

After the last run of the lambda dispatch the lambda setting produced from lambda dispatching the candidate units was compared to the minimum λ value of other units that may be used and were not at maximum power capacity. This time, however, no minima were exceeded. Hence, this is the solution for Case 3 (Table 5.15).

As can be seen from the three cases presented, the results of the lambda dispatch, the lambda dispatch program performed on the IBM:PC yields satisfactory results. However, these three cases do not thoroughly test the algorithmic procedure followed by this program. In order to further the understanding of how the computer algorithm works several more cases were run and the processes of obtaining the results are explained step-by-step.

The data points provided for the remaining cases are different than those used for Cases 1 through 3. A graphical view of the changes made is supplied in Figure 5.2 and the actual data points used can be seen in Table 5.16. The reason for the data differences is because at the time that the remaining cases were developed, the data of Table 5.11 were the only data supplied.

5.5 Case Four

Determination of whether power needs to be sold, bought, or whether constrained lambda dispatch (CLD) should be used.

Given: Any of a group of units that must be on or off. Whether the units are on or off the process is basically the same.

Conclusion: Must sell, must buy, or the CLD should be used.

Discussion: In the process one first compares the sum of all the maximum power production level of all the units to the power demand.

$$\sum_{i=1}^{N} P_i^{max} : P^{tot} \tag{5.11}$$

If the power demanded is equal to or exceeds the sum of the maximum power production levels then one can tell immediately without any dispatch, that all the units should be turned on at maximum power and the difference must be purchased. Table 5.17 shows the results of such a case where the power demand equals 2500 MW. If this is not the case, then one must check the minimum power level against the power demand, i.e.,

$$\text{when } P^{tot} >= \sum_{i=1}^{N} P_i^{max}, \text{ no dispatch necessary,} \tag{5.12}$$

$$\text{and for all units, } P_i^{prd} = P_i^{max} \tag{5.13}$$

$$\text{Power which must be bought} = P^{tot} - \sum_{i=1}^{N} P_i^{max} \tag{5.14}$$

In this situation, when no units must be on, the power demand is compared to the least amount of power that can be produced by a single unit. If the power demand is less than or equal to this amount it becomes clear that this unit should be turned on to its minimum power level, with no other units on, and the difference should be sold.

$$\text{Given, } P_i^{req} = 0,$$

$$\text{when } \min(P_i^{min}) >= P^{tot} \tag{5.15}$$

$$P^{prd} = \min(P_i^{min}), \tag{5.16}$$

$$\text{then we must sell} = P^{prd} - P^{tot} \tag{5.17}$$

This minimum level changes when units must be on. For this case the power level of all the units that must be on at a particular power level are summed and the resulting power level is the minimum power level. Thus, the power demand is compared to this sum and the appropriate action is taken. A sample solution for a case in which 13 units must be on but the power demand is below the sum of the minimum power level of each unit is shown in Table 18. If power demand exceed the power minimum, then one know the dispatch solution will be used.

Given, P_i^{req} <> 0,

$$\text{when} \sum_{i=1}^{N} P_i^{req} >= P^{tot}, \tag{5.18}$$

$$P^{prd} = \sum_{i=1}^{N} P_i^{req}, \tag{5.19}$$

$$\text{hence we must sell} = \sum_{i=1}^{N} P_i^{req} - P^{tot} \tag{5.20}$$

5.6 Case Five

Dispatch between all Jeffreys units with one additional unit on at minimum.

Given: Jef 1 >= 165 MW Jef 2 >= 165 MW

Jef 3 >= 165 MW Law 5 >= 120 MW

P^{tot} = 1000 MW

Conclusion: Jeffrey 1 through 3 produce 293.33 MW each, Lawrence 5 produces 120 MW.

Discussion: After determining that the lambda dispatch solution would be used, as shown in the calculations below,

$$\sum_{i=1}^{N} P_i^{req} = (165+165+165+120)\ \text{MW} = 615\ \text{MW}$$

$$\sum_{i=1}^{N} P_i^{max} = (395+370+395+270+55+45+110+65+140+30+19+65+25+15)\ \text{MW}$$
$$= 2018\ \text{MW}$$

$$\sum_{i=1}^{N} P_i^{req} < P^{tot} < \sum_{i=1}^{N} P_i^{max} \; : \; (615 < 1000 < 2018)$$

all the units are used in calculating a free run lambda value. As discussed before, this lambda value is the system incremental cost that would be brought on if one more unit of power were produced by any unit. It is called free run because the units are brought on with no regard to power minimum or power maximum constraints. Because of this, after the units are dispatched, one must go back and discard those units for which constraints are violated. Then the following process is used.

Generally speaking, if a unit must be on, the free run lambda value is compared to the unit's lambda value at which the power level the unit must produce. If the system lambda is lower than the units lambda value then the unit is left on and the power level at which it is on is subtracted from the power demanded. The remaining power is then dispatched between 1) those units for which constraints are not violated by the free run lambda value and 2) the units that must be on for which the unit's lambda value is less than or equal to the free run lambda value. This case is referred to as the simple case because it need only be done once in order to obtain the conclusion stated above.

5.7 Case 6

Dispatching between two Jeffrey units with one Jeffrey unit on at maximum with no additional must-run unit.

Given: Jef 1 >= 165 MW   Jef 2 >= 165 MW

Jef 3 >= 165 MW   Law 5 >= 120 MW

$P^{tot} = 1270$ MW

Conclusion: Jeffrey 1 and 3 produce 387.5 MW each, Jeffrey 2 produces 370 MW, and Lawrence 5 produces 120 MW.

Discussion: Similar to Case 5, it was determined that dispatch between the Jeffrey units will be necessary. After following the dispatch routine it was found that the maximum production level of Jeffrey 2 has been violated, i.e.,

$$P_{Jef\ 2} > P^{max}_{Jef\ 2}.$$

Thus, Jeffrey 2 is set to its power maximum, the SID is adjusted and the lambda dispatch is performed again with only Jeffrey 1 and 3 as candidate units. The solution then comes out as stated above and shown in Table 5.19.

5.8 Case 7

Dispatch between all Jeffrey units, having 13 units set as must-run units (minimum power capacity).

Given:

Jef 1 >= 165 MW	Law 4 >= 5 MW	Hut 3 >= 15 MW
Jef 2 >= 165 MW	Tec 7 >= 20 MW	Hut 2 >= 10 MW
Jef 3 >= 165 MW	Law 3 >= 20 MW	Hut 1 >= 10 MW
Law 5 >= 120 MW	Hut 4 >= 55 MW	MCP 2 >= 15 MW
Tec 8 >= 40 MW	$P^{tot} = 1390$ MW	

Conclusion: Jeffrey 1, 2, and 3 are all equally dispatched at 357 MW, the remaining must-run units are set at the must-run power levels.

Discussion: In this case the printing of the input data was followed by the free-run lambda dispatch which, in turn, gave negative power level settings for some units. When these units were set at their must-run settings and summed, the total production was found to exceed the power demanded.

$$P_i > P^{tot} \tag{5.21}$$

So the units setting above their minimum power level minimum (the three Jeffrey units) were considered candidate units. The SID was set by subtracting power production levels of all the other units from the total power demand and a lambda dispatch was run again. This time the candidate units were set at allowable levels which, when summed, equaled the SID.

$$P_i = P^{tot} \tag{5.22}$$

The output is shown in Table 5.20.

5.9 Case 8

Dispatch between Jeffrey 1 and 3 with Jeffrey 2 on at maximum and ten of the remaining units at must-run power production levels.

Given:	Jef 1 >= 165 MW	Law 4 >= 5 MW	Hut 3 >= 15 MW
	Jef 2 >= 165 MW	Tec 7 >= 20 MW	Hut 2 >= 10 MW
	Jef 3 >= 165 MW	Law 3 >= 20 MW	Hut 1 >= 10 MW
	Law 5 >= 120 MW	Hut 4 >= 55 MW	MCP 2 >= 15 MW
	Tec 8 >= 40 MW	P^{tot} = 1445 MW	

Conclusion: Jeffrey 1 and 3 are equally dispatched at 377.5 MW, Jeffrey 2 was set at maximum setting (370 MW), the remaining units are set at must-run levels.

Discussion: As is standard, the input listing and free-run dispatch were followed by a redispatching of candidate units because too much power was being produced (Eqn. 5.21). The three candidate units (Jeffrey 1, 2, and 3) were then dispatched only to have the power maximum of Jeffrey 2 exceeded. This meant setting Jeffrey 2 at maximum and redispatching between Jeffrey 1 and 3. This dispatch led to the solution (Eqn. 5.22). The output is shown in Table 5.21.

5.10 Case 9

Dispatch between units Lawrence 5, Tecumseh 8, and Tecumseh 7 with all Jeffrey units on at maximum and all other units at must-run power production levels.

Given:

Jef 1 >= 165 MW	Law 4 >= 5 MW	Hut 3 >= 15 MW
Jef 2 >= 165 MW	Law 7 >= 20 MW	Hut 2 >= 10 MW
Jef 3 >= 165 MW	Law 3 >= 20 MW	Hut 1 >= 10 MW
Law 5 >= 120 MW	Hut 4 >= 55 MW	MCP 2 >= 15 MW
Tec 8 >= 40 MW	P^{tot} = 1591 MW	

Conclusion: Jeffrey 1, 2, and 3 are on at maximum generating capacity; dispatchable demand is distributed between Lawrence 5 (181 MW), Tecumseh 8 (42 MW), and Lawrence 3 (53 MW); the remaining must-run units are set at their generating minimum; the other units are set at 0 MW.

Discussion: After the initial steps of input data printing, free-run dispatch, followed by recognition of negative power production values of units, and a realization that the units with positive power production levels equal to or above the minimum production value sum to a value less than enough to meet power demand, a redispatch was done. The redispatching sets all the Jeffrey units above maximum power level constraints. So the unit with the next lowest lambda value was found. If the power minimum level exceeds the SID then the unit was set to a minimum power level and the units that are producing at maximum power capacity will have their production levels lowered. Otherwise the process is to find the second lowest lambda minimum unit, add this P^{min} value to the P^{min} value of the first candidate unit. Continue this process until the summed minima surpass the SID. At this time, all the units brought on except the last one selected are considered candidate units. Redispatch is then computed and, in this case, the candidate units' lambda setting exceeded the minimum setting of another unit that could be used, i.e.,

$$\lambda^{sys} > P_i^{min}, \tag{5.23}$$

where P_i^{min} is the minimum power setting of unit i.

Since enough power demand existed to bring on this new unit it was considered a candidate unit

$$SID > P_i^{min}, \tag{5.24}$$

with the previous two and redispatching was performed. It is this redispatching which gives the final results as shown in Table 5.22.

5.11 Case 10

Dispatch between Lawrence 5 and Tecumseh 7, all Jeffrey units are set at maximum and the rest of nine units are set at minimum power production level.

Given:

Jef 1 >= 165 MW	Law 5 >= 120 MW	Tec 7 >= 20 MW
Jef 2 >= 165 MW	Tec 8 >= 40 MW	Law 3 >= 20 MW
Jef 3 >= 165 MW	Law 4 >= 5 MW	Hut 4 >= 55 MW

$P^{tot} = 1445$ MW

Conclusion: All Jeffrey units are set at maximum power levels, dispatching was performed between Lawrence 5 and Tecumseh 8. Lawrence 4, Tecumseh 7, Lawrence 3, and Hutchinson 4 were all set at must-run levels.

Discussion: This case follows the same initial steps as most cases have to this point. When it is realized that the sum of the unit power levels was less than the total power demand, the Jeffrey units were set as candidate units, the SID is reset and redispatching is done. After this series of events all of the Jeffrey units were set at maximum power production levels because they were exceeded by the lambda dispatch. Thus, the SID is reset and the unit with the lowest minimum unit lambda was found and selected as the candidate unit. Redispatch was carried out only to find that one must-run units's minimum lambda was exceeded by the λ value found for the system when redispatching. The SID was reset once more and redispatch was again carried out. This time the results are the final solution shown on Table 5.23.

5.12 Case 11

Set all Jeffrey units on at maximum and the rest of the nine units on at must-run levels.

Given:

Jef 1 >= 165 MW	Law 5 >= 120 MW	Tec 7 >= 20 MW
Jef 2 >= 165 MW	Law 8 >= 40 MW	Law 3 >= 20 MW
Jef 3 >= 165 MW	Law 4 >= 5 MW	Hut 4 >= 55 MW

$P^{tot} = 1421$ MW

Conclusion: All Jeffrey units are on at maximum power level, the rest of nine units are set at must-run settings.

Discussion: This case also follows the same initial steps. However, its procedure is simplified because when it is found that the redispatched Jeffrey units are set at maximum (because the power maxima were exceeded) the the power produced is summed over all units, the production level is only one megawatt away from the exact solution.

$$P_i = P^{tot} \pm 1, \qquad (5.25)$$

Plus or minus one megawatt is a tolerated difference, thus the system solution has been found. The results are shown in Table 5.24.

There are many more cases which have been run by this program and not discussed here. This program is written in BASICA and it lends itself readily to modifications if they are found necessary.

5.13 Case Results Comparison

A listing of all the case types and their respective hour-by-hour and cumulative results is shown in Tables 5.9 and 5.10. This discussion should lead one to the conclusion that the CLD is the proper and justified method of solving the unit dispatch problem as faced by KPL and even possibly other Kansas utilities.

All the lambda values shown in Tables 5.9 and 5.10 were taken directly from the computer output as they were developed by the computer algorithms of the various case types, with the exception of the values shown for KPL data. By the strictest definition of incremental cost the KPL data do not have a system incremental cost value (lambda) because lambda dispatch was not used to obtain the cost value shown or to determine generator settings.

The process used to obtain the lambda value given for the KPL data in Tables 5.9 and 5.10 is as follows. In every hour of the first two days of 1985 Jeffrey 1 was never turned on at maximum. In fact, it was always at a lower setting than the other two Jeffrey units. Hence, the cheapest next unit of power produced would be that unit of power produced by Jeffery 1. So Jeffrey 1 was used with its respective power setting for each hour to calculate the lambda value shown in the tables.

The relationship revealed from the results in Tables 5.9 and 5.10 shows that they are inversely related - as the cost increases the lambda value decreases, and vice-versa. This does not seem correct, because the use of lambda dispatch is supposed to save money. However, the reason for the relationship being as it is is really quite sound and logical. When one dispatches using the least expensive first then all of the least expensive fuels will be in use leaving the cost for the next unit of power equal to the cost of producing power from one of the more expensive units. It must not be forgotten: the definition of system incremental cost is the cost for the next unit of power produced. However, when dispatching is constrained by must-run units or is not used at all then it becomes very likely that all the least expensive units will not be used first. When demand has been met, some of the lesser expensive units will still not be running at maximum capacity because other units had to be used. Because of this the cost for the next unit of fuel will be equal to the cost found by using the lesser expensive unit. Thus, using free-run CLD will generally give you a lower operation cost and higher lambda value. One should realize, of course, that there will always be a few of those exceptions to the rule, but this is the general conclusion.

The hourly production costs and the resulting cumulative two day costs for the free-run CLD were lower than any other case type as shown in Tables 5.9 and 5.10. With the exception of the free-run CLD, CLD must-run dispatches yielded lower operating costs than the KPL actual dispatches and the EPRI dispatches.

The must-run CLD results yielded a 200 to 500 dollars per hour savings over the actual dispatches made by KPL. As shown in the calculations completed in Table 5.25, if these values are taken as the average financial savings for every hour over a period of one month, the calculated monthly savings is approximately $120,000. Extrapolate this into annual terms and the savings are in the range of 1.25 to 1.50 million dollars.

The free-run CLD results show an average daily savings of about $40,000 over the actual dispatches made by KPL. Using these values as average financial savings for every day over a period of one month, the calculated monthly savings is approximately 1.14 million dollars. Extrapolate this into annual terms and the savings are approximately 13.7 million dollars (Table 5.26). This translates into approximately 8 percent of the dollars presently spent by KPL over a year's period.

The savings may be more or less than those given above, because the two days for which KPL provided data were winter days. The seasonal variations in electrical demand were not accounted for. The savings during the summer peak demand days will probably not be nearly as great as for the days when KPL has much idle generating capacity. The conclusion is though that even during the summer peak there will be some savings when CLD is used over the current method KPL uses to dispatch its generation to meet its demand.

These power settings and the subsequent cost values given by CLD are unrealistic because of contracts that must be kept, operating system security, unexpected shut downs, etc. This could be very true. However, what CLD provides is a valuable result that, in essence, states the cost of deciding to keep some units on all the time versus turning them on and off, the cost of making or keeping contracts versus doing what may be more profitable to the utilities. It is having access to these types of results that can sometimes lead to a much wiser decision than would have otherwise been made in terms of which units should be used. The conclusion of this work is also that even if the savings are not as significant as they were stated above, they would still be very significant and CLD is a useful tool for studying alternatives in dispatching electrical generators.

As shown by the calculations and values supplied in Table 5.27 the financial savings of the CLD with must-run units over the EPRI program were modest, especially in comparison to the savings the free-run CLD shows over the EPRI program results. If the values supplied for days one and two of 1985 are average for the year then the bi-daily savings of the must-run CLD over the EPRI results turns into a savings of about 400,000 dollars in the course of a year. On the other hand, if the 87,400 dollars savings per day of the free-run CLD results over the EPRI results are considered as average savings then the annual savings turns into approximately 14.7 million dollars.

In addition to these financial production savings there is the convenience factor to be considered. With respect to the time sharing program presently being used by KPL, the IBM:PC is much easier to use. One could take a PC with them if it were necessary, could use the PC to run these programs in as much or less time than it takes to run the EPRI program, could use the PC when they wanted and not worry about the mainframe system being down, and the direct cost of using the PC versus the time sharing setup is much less expensive over the long run.

In view of the advantages and cost savings provided in the preceding discussion, one can see that the use of the IBM:PC and the CLD program is not only justified, but also very sensible.

Table 5.1: Generator Settings and Hourly Settings.

KP&L GENERATING UNIT LOADING
WEDNESDAY 1/1/85

Hour Ending	Total Load to Generators (MW)	JEC 1 (MW)	JEC 2 (MW)	JEC 3 (MW)	LAW 4 (MW)	LAW 5 (MW)	TEC 7 (MW)
0100	1157	243	326	295	35	124	34
0200	998	222	305	274	37	124	36
0300	956	201	301	263	36	122	33
0400	922	179	294	253	37	123	36
0500	911	166	293	249	36	133	34
0600	940	184	299	254	37	132	34
0700	998	221	303	272	36	131	35
0800	1051	246	318	285	38	129	35
0900	1097	262	332	302	37	129	35
1000	1160	282	350	326	37	129	36
1100	1204	293	366	343	37	130	35
1200	1198	300	358	342	36	128	34
1300	1185	294	350	339	36	130	36
1400	1159	279	352	326	38	128	36
1500	1128	263	343	320	38	129	35
1600	1102	263	338	303	37	127	34
1700	1053	233	313	305	36	128	38
1800	1171	309	367	293	38	130	34
1900	1253	329	386	344	37	127	30
2000	1222	315	367	346	36	127	31
2100	1169	293	358	330	36	129	32
2200	1125	278	343	320	36	128	20
2300	1119	264	340	324	37	126	28
2400	1047	226	330	297	37	128	29

Table 5.2: Generator Settings and Hourly Demand.

KP&L GENERATING UNIT LOADING
WEDNESDAY 1/2/85

Hour Ending	Total Load to Generators (MW)	JEC 1 (MW)	JEC 2 (MW)	JEC 3 (MW)	LAW 4 (MW)	LAW 5 (MW)	TEC 7 (MW)
0100	949	202	325	228	37	130	27
0200	883	144	297	249	36	128	29
0300	878	142	291	250	36	129	30
0400	894	154	291	254	38	127	30
0500	925	183	290	256	36	131	29
0600	1007	226	308	275	37	134	27
0700	1156	288	348	323	38	132	27
0800	1351	362	402	393	38	129	27
0900	1331	351	400	386	37	131	26
1000	1348	357	400	396	37	131	27
1100	1332	348	388	401	37	130	28
1200	1349	365	400	393	36	128	27
1300	1321	344	400	381	35	129	32
1400	1339	354	404	384	35	128	34
1500	1292	338	392	365	34	127	36
1600	1382	349	403	384	35	176	35
1700	1326	335	390	369	33	158	41
1800	1375	370	404	391	35	134	41
1900	1457	394	399	401	49	172	42
2000	1399	388	402	396	37	138	38
2100	1363	384	397	376	36	130	40
2200	1289	382	393	309	36	127	42
2300	1172	368	394	211	36	125	38
2400	1008	280	347	186	35	123	37

Table 5.3: Free Run Results (1/1/85 Data)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	TOTAL ($/HR)
01	MW	361.99	405.00	390.00	0	
	$/HR	5866.68	6392.06	6229.48	0	18488.22
02	MW	305.05	360.47	332.48	0	
	$/HR	5015.21	5727.89	5369.50	0	16112.60
03	MW	291.19	346.33	318.48	0	
	$/HR	4810.63	5519.18	5162.87	0	15492.68
04	MW	279.97	334.88	307.15	0	
	$/HR	4645.78	5350.99	4996.37	0	14993.15
05	MW	276.34	331.18	303.48	0	
	$/HR	4592.59	5296.74	4942.65	0	14831.98
06	MW	285.91	340.94	313.15	0	
	$/HR	4732.96	5439.95	5084.43	0	15257.35
07	MW	305.05	360.47	332.48	0	
	$/HR	5015.21	5727.89	5369.50	0	16112.60
08	MW	322.54	378.31	350.15		
	$/HR	5274.85	5992.78	5631.74	0	16899.36
09	MW	337.72	393.80	365.48	0	
	$/HR	5501.53	6224.04	5860.69	0	17586.26
10	MW	363.49	405.00	391.51	0	
	$/HR	5889.23	6392.06	6252.26	0	18533.54
11	MW	394.00	405.00	405.00	0	
	$/HR	6352.81	6392.06	6457.62	0	19201.49
12	MW	388.00	405.00	405.00	0	
	$/HR	6261.25	6392.06	6456.62	0	19109.92
13	MW	375.93	405.00	404.07	0	
	$/HR	6077.61	6392.06	6442.53	0	18912.19
14	MW	362.99	405.00	391.01	0	
	$/HR	5881.71	6392.06	6244.66	0	18518.43
15	MW	347.95	404.24	375.81	0	
	$/HR	5655.00	6380.61	6015.69	0	18051.30

Table 5.3 (Cont.)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	TOTAL ($/HR)
16	MW	339.37	395.48	367.15	0	
	$/HR	5526.25	6249.26	5885.65	0	17661.15
17	MW	323.19	378.99	350.81	0	
	$/HR	5284.68	6002.81	5641.67	0	16929.15
18	MW	368.96	405.00	397.04	0	
	$/HR	5972.01	6392.06	6335.87	0	18699.94
19	MW	323.00	405.00	405.00	120.00	
	$/HR	5281.73	6392.06	6456.62	3284.61	21415.02
20	MW	292.00	405.00	405.00	120.00	
	$/HR	4822.60	6392.06	6456.62	3284.61	20955.89
21	MW	367.97	405.00	396.03	0	
	$/HR	5956.96	6392.06	6320.66	0	18669.67
22	MW	346.96	403.23	374.81	0	
	$/HR	5640.12	6365.43	6000.66	0	18006.21
23	MW	344.98	401.41	372.81	0	
	$/HR	5610.39	6335.09	5970.63	0	17916.11
24	MW	321.22	376.97	348.81	0	
	$/HR	5255.19	5972.73	5611.89	0	16839.81

TOTAL $ FOR FREE RUN = 425194.02

Table 5.4: Free Run Results (1/2/85)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	LAW4	LAW3	TOTAL ($/HR)
01	MW	288.88	343.97	316.15	0	0	0	
	$/HR	4776.63	5484.49	5128.54	0	0	0	15389.66
02	MW	267.10	321.75	294.15	0	0	0	
	$/HR	4457.51	5158.93	4806.23	0	0	0	14422.67
03	MW	265.45	320.07	292.48	0	0	0	
	$/HR	4433.44	5134.38	4781.92	0	0	0	14349.73
04	MW	270.73	325.46	297.82	0	0	0	
	$/HR	4510.52	5213.01	4859.77	0	0	0	14583.31
05	MW	280.96	335.89	308.15	0	0	0	
	$/HR	4660.29	5365.81	5011.04	0	0	0	15037.14
06	MW	308.02	363.50	335.48	0	0	0	
	$/HR	5059.18	5772.76	5413.91	0	0	0	16245.85
07	MW	361.50	405.00	389.49	0	0	0	
	$/HR	5859.16	6392.06	6221.89	0	0	0	18473.11
08	MW	405.00	405.00	405.00	136.00	0	0	
	$/HR	6524.18	6392.06	6456.62	3585.95	0	0	22955.81
09	MW	405.00	405.00	405.00	116.00	0	0	
	$/HR	6521.18	6392.06	6456.62	3210.20	0	0	22580.06
10	MW	405.00	405.00	405.00	133.00	0	0	
	$/HR	6521.18	6392.06	6456.62	3529.00	0	0	22898.86
11	MW	405.00	405.00	405.00	117.00	0	0	
	$/HR	6521.18	6392.06	6456.62	3228.77	0	0	22598.63
12	MW	405.00	405.00	405.00	134.00	0	0	
	$/HR	6521.18	6392.06	6456.62	3547.96	0	0	22917.82
13	MW	391.00	405.00	405.00	120.00	0	0	
	$/HR	6307.00	6392.06	6456.62	3284.61	0	0	22440.29
14	MW	405.00	405.00	405.00	124.00	0	0	
	$/HR	6521.18	6392.06	6456.62	3359.39	0	0	22729.26
15	MW	405.00	405.00	405.00	0	55.00	22.00	
	$/HR	6521.19	6392.06	6456.62	0	1438.59	686.20	21494.66

Table 5.4 (Cont.)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	LAW4	LAW3	TOTAL ($/HR)
16	MW	405.00	405.00	405.00	167.00	0	0	
	$/HR	6521.18	6392.06	6456.62	4186.53	0	0	23556.39
17	MW	396.00	405.00	405.00	120.00	0	0	
	$/HR	6383.37	6392.06	6456.62	3284.61	0	0	22516.67
18	MW	405.00	405.00	405.00	160.00	0	0	
	$/HR	6521.18	6392.06	6456.62	4048.99	0	0	23418.85
19	MW	405.00	405.00	405.00	187.00	55.00	0	
	$/HR	6521.18	6392.06	6456.62	4585.73	1438.56	0	25394.18
20	MW	405.00	405.00	405.00	184.00	0	0	
	$/HR	6521.18	6392.06	6456.62	4526.26	0	0	23895.12
21	MW	405.00	405.00	405.00	148.00	0	0	
	$/HR	6521.18	5392.06	6456.62	3815.81	0	0	23185.67
22	MW	405.00	405.00	405.00	0	55.00	0	
	$/HR	6521.19	6392.06	6456.62	0	1438.56	0	21494.65
23	MW	369.46	405.00	397.54	0	0	0	
	$/HR	5979.55	6392.06	6343.48	0	0	0	18715.08
24	MW	308.35	363.84	335.81	0	0	0	
	$/HR	5064.07	5777.74	6518.85	0	0	0	16260.66

TOTAL $ FOR FREE RUN = 487558.13

Table 5.5: CLD with Must-Run Units Results (1/1/85 Data)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	LAW4	TEC7	TOTAL ($/HR)
01	MW	295.47	323.44	268.10	120.00	30.00	20.00	
	$/HR	4824.99	5182.71	4471.97	3283.58	936.27	645.26	19344.78
02	MW	275.80	303.56	248.64	120.00	30.00	20.00	
	$/HR	4538.96	4893.53	4189.01	3283.58	936.27	645.26	18486.62
03	MW	261.80	289.40	234.79	120.00	30.00	20.00	
	$/HR	4336.60	4688.95	3988.83	3283.58	936.27	645.26	17879.49
04	MW	250.47	277.95	223.58	120.00	30.00	20.00	
	$/HR	4173.55	4524.11	3827.53	3283.58	936.27	645.26	17390.31
05	MW	246.80	274.24	219.96	120.00	30.00	20.00	
	$/HR	4120.95	4470.93	3775.49	3283.58	936.27	645.26	17232.49
06	MW	256.47	284.01	229.52	120.00	30.00	20.00	
	$/HR	4259.78	4611.29	3912.84	3283.58	936.27	645.26	17649.03
07	MW	275.80	303.56	248.64	120.00	30.00	20.00	
	$/HR	4538.96	4893.53	4189.01	3283.58	936.27	645.26	18486.62
08	MW	293.47	321.42	266.12	120.00	30.00	20.00	
	$/HR	4795.81	5153.20	4443.10	3283.58	936.27	645.26	19257.23
09	MW	308.80	336.92	281.28	120.00	30.00	20.00	
	$/HR	5020.08	5379.94	4664.96	3283.58	936.27	645.26	19930.11
10	MW	329.80	358.15	302.06	120.00	30.00	20.00	
	$/HR	5329.28	5692.54	4970.83	3283.58	936.27	645.26	20857.77
11	MW	344.46	372.97	316.56	120.00	30.00	20.00	
	$/HR	5546.62	5912.27	5185.84	3283.58	936.27	645.26	21509.84
12	MW	342.46	370.95	314.59	120.00	30.00	20.00	
	$/HR	5516.91	5882.24	5156.45	3283.58	936.27	645.26	21420.72
13	MW	338.13	366.57	310.30	120.00	30.00	20.00	
	$/HR	5452.63	5817.24	5092.86	3283.58	936.27	645.26	21227.84
14	MW	329.46	357.81	301.73	120.00	30.00	20.00	
	$/HR	5324.35	5687.56	4965.96	3283.58	936.27	645.26	20842.99
15	MW	319.13	347.36	291.51	120.00	30.00	20.00	
	$/HR	5171.93	5533.46	4815.18	3283.58	936.27	645.26	20385.69

Table 5.5 (Cont.)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	LAW4	TEC7	TOTAL ($/HR)
16	MW	310.46	338.60	282.93	120.00	30.00	20.00	
	$/HR	5044.54	5404.67	4689.15	3283.58	936.27	645.26	20003.47
17	MW	294.13	322.09	266.78	120.00	30.00	20.00	
	$/HR	4805.53	5163.04	4452.82	3283.58	936.27	645.26	19286.40
18	MW	333.46	361.85	305.68	120.00	30.00	20.00	
	$/HR	5383.51	5747.36	5024.48	3283.58	936.27	645.26	21020.46
19	MW	360.79	389.48	332.72	120.00	30.00	20.00	
	$/HR	5790.01	6158.33	5426.61	3283.58	936.27	645.26	22240.07
20	MW	350.46	379.04	322.50	120.00	30.00	20.00	
	$/HR	5635.86	6002.49	5174.12	3283.58	936.27	645.26	2177.60
21	MW	332.80	361.18	305.02	120.00	30.00	20.00	
	$/HR	5373.64	5737.39	5014.72	3282.58	936.27	645.26	20990.87
22	MW	318.13	346.35	290.52	120.00	30.00	20.00	
	$/HR	5157.21	5518.58	4800.62	3283.58	936.27	645.26	20341.53
23	MW	316.13	344.33	288.54	120.00	30.00	20.00	
	$/HR	5127.79	5488.83	4771.51	3283.58	936.27	645.26	20253.25
24	MW	292.13	320.07	264.80	120.00	30.00	20.00	
	$/HR	4776.36	5133.55	4423.86	3285.58	936.27	645.26	19198.89

TOTAL $ FOR OPT = 477014.08

Table 5.6: KPL Results (1/1/85 Data)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	LAW4	TEC7	TOTAL ($/HR)
01	MW	295.00	326.00	243.00	124.00	35.00	34.00	
	$/HR	4818.18	5220.12	4107.33	3358.33	1034.28	937.45	19475.69
02	MW	274.00	305.00	222.00	124.00	37.00	36.00	
	$/HR	4512.86	4914.45	3804.81	3358.33	1073.81	980.00	18644.26
03	MW	263.00	301.00	201.00	122.00	36.00	33.00	
	$/HR	4353.87	4856.49	3504.67	3320.91	1054.02	916.26	18006.22
04	MW	253.00	294.00	179.00	123.00	37.00	36.00	
	$/HR	4209.90	4755.27	3192.79	3339.61	1073.81	980.00	17551.37
05	MW	249.00	293.00	166.00	133.00	36.00	34.00	
	$/HR	4152.45	4740.83	3009.73	3527.85	1054.02	937.45	17422.33
06	MW	254.00	299.00	184.00	132.00	37.00	34.00	
	$/HR	4224.27	4827.54	3263.45	3508.92	1073.81	937.45	17835.44
07	MW	272.00	303.00	221.00	131.00	36.00	35.00	
	$/HR	4483.91	4885.46	3790.47	3490.01	1054.02	958.70	18662.57
08	MW	285.00	318.00	246.00	129.00	38.00	35.00	
	$/HR	4672.50	5103.40	4150.74	3452.28	1093.64	958.70	19431.26
09	MW	302.00	332.00	262.00	129.00	37.00	35.00	
	$/HR	4920.48	5307.88	4383.09	3452.28	1073.81	958.70	20096.23
10	MW	326.00	350.00	282.00	129.00	37.00	36.00	
	$/HR	5273.20	5572.31	4675.46	3452.28	1073.81	980.00	21027.05
11	MW	343.00	366.00	293.00	130.00	37.00	35.00	
	$/HR	5524.90	5808.79	4837.18	3471.13	1073.81	958.70	21674.51
12	MW	342.00	358.00	300.00	128.00	36.00	34.00	
	$/HR	5510.05	5690.38	4940.44	3433.44	1054.02	937.45	21565.78
13	MW	339.00	350.00	294.00	130.00	36.00	36.00	
	$/HR	5465.54	5572.31	4851.92	3471.13	1054.02	980.00	21394.91
14	MW	326.00	352.00	279.00	128.00	38.00	36.00	
	$/HR	5273.20	5601.79	4631.47	3433.44	1093.64	980.00	21013.54
15	MW	320.00	343.00	263.00	129.00	38.00	35.00	
	$/HR	5184.73	5469.27	4397.65	3452.28	1093.64	958.70	20556.27

Table 5.6 (Cont.)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	LAW4	TEC7	TOTAL ($/HR)
16	MW	303.00	338.00	263.00	127.00	37.00	34.00	
	$/HR	4935.12	5395.83	4397.65	3414.63	1073.81	937.45	20154.49
17	MW	305.00	313.00	233.00	128.00	36.00	38.00	
	$/HR	4964.40	5030.62	3962.98	3433.44	1054.02	1022.75	19468.21
18	MW	293.00	367.00	309.00	130.00	38.00	34.00	
	$/HR	4789.01	5823.61	5073.59	3471.13	1093.64	937.45	21188.43
19	MW	344.00	386.00	329.00	127.00	37.00	30.00	
	$/HR	5539.75	6106.29	5371.03	3414.63	1073.81	852.96	22358.47
20	MW	346.00	367.00	315.00	127.00	36.00	31.00	
	$/HR	5569.48	5823.61	5162.59	3414.63	1054.02	874.01	21898.34
21	MW	330.00	358.00	296.00	129.00	36.00	23.00	
	$/HR	5332.28	5690.38	4837.18	3452.28	1054.02	707.04	21073.18
22	MW	320.00	343.00	278.00	128.00	36.00	20.00	
	$/HR	5184.73	5469.27	4616.81	3433.44	1054.02	645.26	20403.54
23	MW	324.00	340.00	264.00	126.00	37.00	28.00	
	$/HR	5243.69	5425.19	4412.23	3395.84	1073.81	811.02	20361.77
24	MW	297.00	330.00	226.00	128.00	37.00	29.00	
	$/HR	4847.39	5278.61	3862.25	3433.44	1073.81	831.97	19327.46

TOTAL $ FOR KPL = 480591.33

Table 5.7: CLD with Must-Run Unit Results (1/2/85 Data)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	LAW4	TEC7	TOTAL ($/HR)
01	MW	259.47	287.05	232.49	120.00	30.00	20.00	
	$/HR	4302.97	4654.96	3955.56	3283.58	936.27	645.26	17778.61
02	MW	237.47	264.81	210.72	120.00	30.00	20.00	
	$/HR	3987.37	4335.88	3643.35	3283.58	936.27	645.26	16834.72
03	MW	235.80	263.12	209.08	120.00	30.00	20.00	
	$/HR	3963.56	4311.82	3619.80	3283.58	936.27	645.26	16760.30
04	MW	241.14	268.51	214.35	120.00	30.00	20.00	
	$/HR	4039.79	4388.88	3695.21	3283.58	936.27	645.26	16989.00
05	MW	251.47	278.96	224.57	120.00	30.00	20.00	
	$/HR	4187.91	4538.63	3841.74	3283.58	936.27	645.26	17433.39
06	MW	278.80	306.59	251.61	120.00	30.00	20.00	
	$/HR	4582.46	4937.51	4232.04	3283.58	936.27	645.26	18617.12
07	MW	328.46	356.80	300.74	120.00	30.00	20.00	
	$/HR	5309.58	5672.62	4951.34	3283.58	936.27	645.26	20798.66
08	MW	402.26	405.00	373.74	120.00	30.00	20.00	
	$/HR	6414.34	6390.87	6044.23	3283.58	936.27	645.26	23714.56
09	MW	392.21	405.00	363.79	120.00	30.00	20.00	
	$/HR	6262.12	6390.87	5893.64	3283.58	936.27	645.26	23411.75
10	MW	400.75	405.00	372.25	120.00	30.00	20.00	
	$/HR	6391.48	6390.87	6021.61	3283.58	936.27	645.26	23669.07
11	MW	392.71	405.00	364.29	120.00	30.00	20.00	
	$/HR	6269.72	6390.87	5901.16	3283.58	936.27	645.26	23426.86
12	MW	401.25	405.00	372.75	120.00	30.00	20.00	
	$/HR	6399.10	6390.87	6029.15	3283.58	936.27	645.26	23684.23
13	MW	387.18	405.00	358.82	120.00	30.00	20.00	
	$/HR	6186.21	6380.87	5818.55	3283.58	936.27	645.26	23260.74
14	MW	396.23	405.00	367.77	120.00	30.00	20.00	
	$/HR	6322.94	6390.87	5953.81	3283.58	936.27	645.26	23532.74
15	MW	373.79	402.63	345.58	120.00	30.00	20.00	
	$/HR	5984.75	6355.21	5619.25	3283.58	936.27	645.26	22824.33

Table 5.7 (Cont.)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	LAW4	TEC7	TOTAL ($/HR)
16	MW	405.00	405.00	402.00	120.00	30.00	20.00	
	$/HR	6455.93	6390.87	6475.01	3283.58	936.27	645.26	24186.93
17	MW	389.69	405.00	361.31	120.00	30.00	20.00	
	$/HR	6224.15	6390.87	5856.08	3283.58	936.27	645.26	23336.21
18	MW	405.00	405.00	395.00	120.00	30.00	20.00	
	$/HR	6455.93	6390.87	6367.90	3283.58	936.27	645.26	24079.82
19	MW	405.00	405.00	405.00	180.38	41.62	20.00	
	$/HR	6455.93	6390.87	6520.99	4450.95	1165.83	645.26	25629.83
20	MW	405.00	405.00	405.00	134.00	30.00	20.00	
	$/HR	6455.93	6390.87	6520.99	3546.80	936.27	645.26	24496.12
21	MW	405.00	405.00	383.00	120.00	30.00	20.00	
	$/HR	6455.93	6390.87	6184.90	3283.58	936.27	645.26	23896.82
22	MW	372.79	401.62	344.59	120.00	30.00	20.00	
	$/HR	5969.73	6340.03	5604.40	3283.58	936.27	645.26	22779.29
23	MW	333.80	362.19	306.01	120.00	30.00	20.00	
	$/HR	5388.44	5752.35	5029.36	3283.58	936.27	645.26	21035.27
24	MW	279.13	306.93	251.94	120.00	30.00	20.00	
	$/HR	4587.29	4942.40	4236.82	3283.58	936.27	645.26	18631.63

TOTAL $ FOR OPT = 520805.02

Table 5.8: KPL Results (1/2/85 Data)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	LAW4	TEC7	TOTAL ($/HR)
01	MW	228.00	325.00	202.00	130.00	37.00	27.00	
	$/HR	3852.29	5205.51	3518.91	3471.13	1073.81	790.13	17911.78
02	MW	249.00	297.00	144.00	128.00	36.00	29.00	
	$/HR	4152.45	4798.62	2995.68	3433.44	1054.02	831.97	17266.18
03	MW	250.00	291.00	142.00	129.00	36.00	30.00	
	$/HR	4166.81	4711.96	2995.68	3452.28	1054.02	852.96	17233.71
04	MW	254.00	291.00	154.00	127.00	38.00	30.00	
	$/HR	4224.27	4711.96	2995.68	3414.63	1093.64	852.96	17293.15
05	MW	256.00	290.00	183.00	131.00	36.00	29.00	
	$/HR	4253.03	4697.54	3249.30	3490.01	1054.02	831.97	17575.88
06	MW	275.00	308.00	226.00	134.00	37.00	27.00	
	$/HR	4527.35	4957.97	3862.25	3546.80	1073.81	790.13	18758.31
07	MW	323.00	348.00	288.00	132.00	38.00	27.00	
	$/HR	5228.94	5542.84	4763.59	3508.92	1093.64	790.13	20928.06
08	MW	393.00	402.00	362.00	129.00	38.00	27.00	
	$/HR	6274.13	6345.81	5866.53	3452.28	1093.64	790.13	23822.51
09	MW	386.00	400.00	351.00	131.00	37.00	26.00	
	$/HR	6168.43	6315.79	5700.71	3490.01	1073.81	769.28	23518.04
10	MW	396.00	400.00	357.00	131.00	37.00	27.00	
	$/HR	6319.51	6315.79	5791.08	3490.01	1073.81	790.13	23780.33
11	MW	401.00	388.00	348.00	131.00	37.00	28.00	
	$/HR	6395.25	6236.15	5655.60	3490.01	1073.81	811.02	23561.84
12	MW	393.00	400.00	365.00	128.00	36.00	27.00	
	$/HR	6274.13	6315.79	5911.87	3433.44	1054.02	790.13	23779.38
13	MW	381.00	400.00	344.00	129.00	35.00	32.00	
	$/HR	6093.10	6315.79	5595.53	3452.28	1034.28	895.11	23386.08
14	MW	384.00	404.00	354.00	128.00	35.00	34.00	
	$/HR	6138.28	6375.84	5745.87	3433.44	1034.28	937.45	23665.17
15	MW	365.00	392.00	338.00	127.00	34.00	36.00	
	$/HR	5852.92	6195.95	5505.58	3414.63	1014.59	980.00	22963.66

Table 5.8 (Cont.)

HR	UNIT	JEC3	JEC2	JEC1	LAW5	LAW4	TEC7	TOTAL ($/HR)
16	MW	384.00	403.00	349.00	176.00	35.00	35.00	
	$/HR	6138.28	6360.82	5671.63	4363.45	1034.28	958.70	24526.16
17	MW	369.00	390.00	335.00	158.00	33.00	41.00	
	$/HR	5912.83	6166.04	5460.48	4008.49	994.94	1087.24	23630.23
18	MW	391.00	404.00	370.00	134.00	35.00	41.00	
	$/HR	6243.90	6375.84	5987.53	3546.80	1034.28	1087.24	24275.60
19	MW	401.00	399.00	394.00	172.00	49.00	42.00	
	$/HR	6395.25	6300.79	6352.62	4283.92	1314.91	1108.84	25756.34
20	MW	396.00	402.00	388.00	138.00	37.00	38.00	
	$/HR	6319.51	6345.81	6261.06	3622.83	1073.81	1022.75	24645.76
21	MW	376.00	397.00	384.00	130.00	36.00	40.00	
	$/HR	6017.89	6270.81	6200.12	3471.13	1054.02	1065.69	24079.68
22	MW	309.00	393.00	382.00	127.00	36.00	42.00	
	$/HR	5023.04	6210.91	6169.69	3414.63	1054.02	1108.84	22981.13
23	MW	211.00	394.00	368.00	125.00	36.00	38.00	
	$/HR	3611.02	6225.88	5957.25	3377.07	1054.02	1022.75	21247.99
24	MW	186.00	347.00	280.00	123.00	35.00	37.00	
	$/HR	3257.02	5528.12	4646.12	3339.61	1034.28	1001.35	18808.49

TOTAL $ FOR KPL = 525395.46

Table 5.9: Comparative Cost and Lambda Values of Various Case Types (1/1/85 Data)

HR	CASE TYPE	(CLD) FREE RUN	(CLD) MUST-RUN OPTIMAL	(KPL) ACTUAL	EPRI
01	COST ($)	18488.22	19344.78	19475.69	10830
	LAMBDA	15.10	14.60	14.46	14.78
02	COST ($)	16112.60	18486.62	18644.26	18489
	LAMBDA	14.80	14.49	14.35	14.49
03	COST ($)	15492.68	17879.49	18006.22	17880
	LAMBDA	14.72	14.42	14.24	14.42
04	COST ($)	14993.15	17390.31	17551.37	17392
	LAMBDA	14.66	14.36	14.12	14.36
05	COST ($)	14831.98	17232.49	17422.33	17392
	LAMBDA	14.64	14.34	14.05	14.36
06	COST ($)	15257.35	17649.03	17835.44	17392
	LAMBDA	14.69	14.39	14.14	14.36
07	COST ($)	16112.60	18486.62	18662.57	18489
	LAMBDA	14.80	14.49	14.34	14.49
08	COST ($)	16899.36	19257.23	19431.26	19260
	LAMBDA	14.89	14.59	14.48	14.59
09	COST ($)	17586.26	19930.11	20096.23	19932
	LAMBDA	14.97	14.67	14.57	14.67
10	COST ($)	18533.54	20857.77	21027.05	20909
	LAMBDA	15.11	14.78	14.67	14.79
11	COST ($)	19201.49	21509.84	21674.51	21526
	LAMBDA	15.28	14.86	14.73	14.86
12	COST ($)	19109.92	21420.72	21565.78	21526
	LAMBDA	15.24	14.85	14.77	14.86
13	COST ($)	18912.19	21227.84	21396.91	21526
	LAMBDA	15.18	14.83	14.74	14.86
14	COST ($)	18518.43	20842.99	21013.54	20844
	LAMBDA	15.11	14.78	14.66	14.78
15	COST ($)	18051.30	20385.69	20556.27	20387
	LAMBDA	15.03	14.72	14.57	14.72

Table 5.9 (Cont.)

HR	CASE TYPE	(CLD) FREE RUN	(CLD) MUST-RUN OPTIMAL	(KPL) ACTUAL	EPRI
16	COST ($)	17661.15	20003.47	20154.49	20006
	LAMBDA	14.98	14.68	14.57	14.68
17	COST ($)	16929.15	19286.40	19468.21	19289
	LAMBDA	14.89	14.59	14.41	14.59
18	COST ($)	18699.94	21020.46	21188.43	21023
	LAMBDA	15.14	14.80	14.82	14.80
19	COST ($)	21415.02	22240.07	22358.47	22242
	LAMBDA	14.89	14.95	14.93	14.94
20	COST ($)	20955.89	21777.60	21898.34	21781
	LAMBDA	14.73	14.89	14.85	14.89
21	COST ($)	18669.67	20990.87	21073.18	20992
	LAMBDA	15.14	14.80	14.73	14.79
22	COST ($)	18006.21	20341.54	20403.54	20343
	LAMBDA	15.02	14.72	14.65	14.72
23	COST ($)	17913.11	20253.25	20361.77	20343
	LAMBDA	15.01	14.71	14.58	14.72
24	COST ($)	16839.81	19198.89	19327.46	19201
	LAMBDA	14.88	14.58	14.58	14.58
TOTAL COST ($)		425194.02	477014.08	480591.33	478991

Table 5.10: Comparative Cost and Lambda Values of Various Case Types (1/2/85 Data)

HR	CASE TYPE	(CLD) FREE RUN	(CLD) MUST-RUN OPTIMAL	(KPL) ACTUAL	EPRI
01	COST ($)	15389.66	17778.61	17911.78	17671
	LAMBDA	14.71	14.41	14.24	14.39
02	COST ($)	14422.67	16831.72	17266.18	16770
	LAMBDA	14.59	14.29	13.93	14.28
03	COST ($)	14349.73	16760.30	17233.71	16770
	LAMBDA	14.58	14.29	13.92	14.28
04	COST ($)	14583.31	16989.00	17293.15	16770
	LAMBDA	14.61	14.31	13.98	14.28
05	COST ($)	15037.14	17433.39	17575.88	17436
	LAMBDA	14.67	14.36	14.14	14.36
06	COST ($)	16245.85	18617.12	18758.31	18618
	LAMBDA	14.81	14.51	14.37	14.51
07	COST ($)	18473.11	10798.66	20928.06	20801
	LAMBDA	15.10	14.77	14.71	14.77
08	COST ($)	22955.81	23714.56	23822.51	23992
	LAMBDA	19.02	15.17	15.10	15.25
09	COST ($)	22580.06	23411.75	23518.04	23495
	LAMBDA	18.56	15.11	15.05	15.13
10	COST ($)	22898.86	23669.07	23780.33	23495
	LAMBDA	18.98	15.16	15.08	15.13
11	COST ($)	22598.63	23426.86	23561.84	23495
	LAMBDA	18.58	15.12	15.03	15.13
12	COST ($)	22917.82	23684.23	23779.38	23495
	LAMBDA	18.97	15.16	15.12	15.13
13	COST ($)	22440.29	23260.74	23386.08	23495
	LAMBDA	15.26	15.09	15.00	15.13
14	COST ($)	22729.26	23532.74	23665.17	23495
	LAMBDA	18.74	15.14	15.06	15.13
15	COST ($)	21494.66	22824.33	22963.66	22984
	LAMBDA	20.61	15.02	14.98	15.04

Table 5.10 (Cont.)

HR	CASE TYPE	(CLD) FREE RUN	(CLD) MUST-RUN OPTIMAL	(KPL) ACTUAL	EPRI
16	COST ($)	23556.39	24186.93	24526.16	24235
	LAMBDA	19.73	15.32	15.03	15.49
17	COST ($)	22516.67	23336.21	23630.23	23403
	LAMBDA	15.29	15.10	14.96	15.11
18	COST ($)	23418.85	24079.82	24275.60	24235
	LAMBDA	19.57	15.28	15.15	15.37
19	COST ($)	25394.18	25629.83	25756.34	25282
	LAMBDA	20.19	20.03	15.28	19.77
20	COST ($)	23895.12	24496.12	24645.76	24638
	LAMBDA	20.12	18.96	15.24	19.14
21	COST ($)	23185.67	23896.82	24079.68	24037
	LAMBDA	19.29	15.22	15.22	15.27
22	COST ($)	21498.65	22779.29	22981.13	22884
	LAMBDA	20.87	15.01	15.21	15.02
23	COST ($)	18715.08	21035.27	21247.99	21038
	LAMBDA	15.14	14.80	15.14	14.80
24	COST ($)	16260.66	18631.63	18808.49	18633
	LAMBDA	14.82	14.51	14.66	14.51
TOTAL COST ($)		487558.13	520805.02	525395.46	521162

Table 5.11. Original Data Setting Values for Case Runs

Unit	α (MBtu/H)	β (MBtu/MW^{2}H	$\gamma(10^3)$ (MBtu/MW^{2}H)	C ($/MBtu)	Power MIN (MW)	Power MAX (MW)	Lambda MIN $/MWH	Lambda MAX $/MWH
J1	513.66	8.98	1.84	1.45	165	395	13.901	15.129
J2	513.66	8.98	1.84	1.45	165	370	13.901	14.995
J3	513.66	8.98	1.84	1.45	165	395	13.901	15.129
L5	550.97	7.22	5.22	2.20	120	270	18.640	22.062
T8	201.45	7.19	1.73	2.22	40	110	19.034	24.411
L3	78.40	9.02	14.50	2.20	20	45	21.133	22.745
T7	110.33	8.79	11.32	2.22	20	65	20.519	22.781
L4	169.43	8.22	10.60	2.20	5	55	18.317	20.652
H4	152.679	8.449	8.60	2.90	55	140	27.247	31.487
H3	32.258	11.906	24.20	2.90	15	30	36.636	38.744
H2	15.514	12.120	58.50	2.90	10	19	38.541	41.594
H1	15.514	12.120	58.50	2.90	10	19	38.541	41.594
MCD2	34.699	9.031	89.00	2.90	15	25	33.832	38.927
ABILE CT	32.100	8.453	4.30	2.99	25	65	25.917	26.946

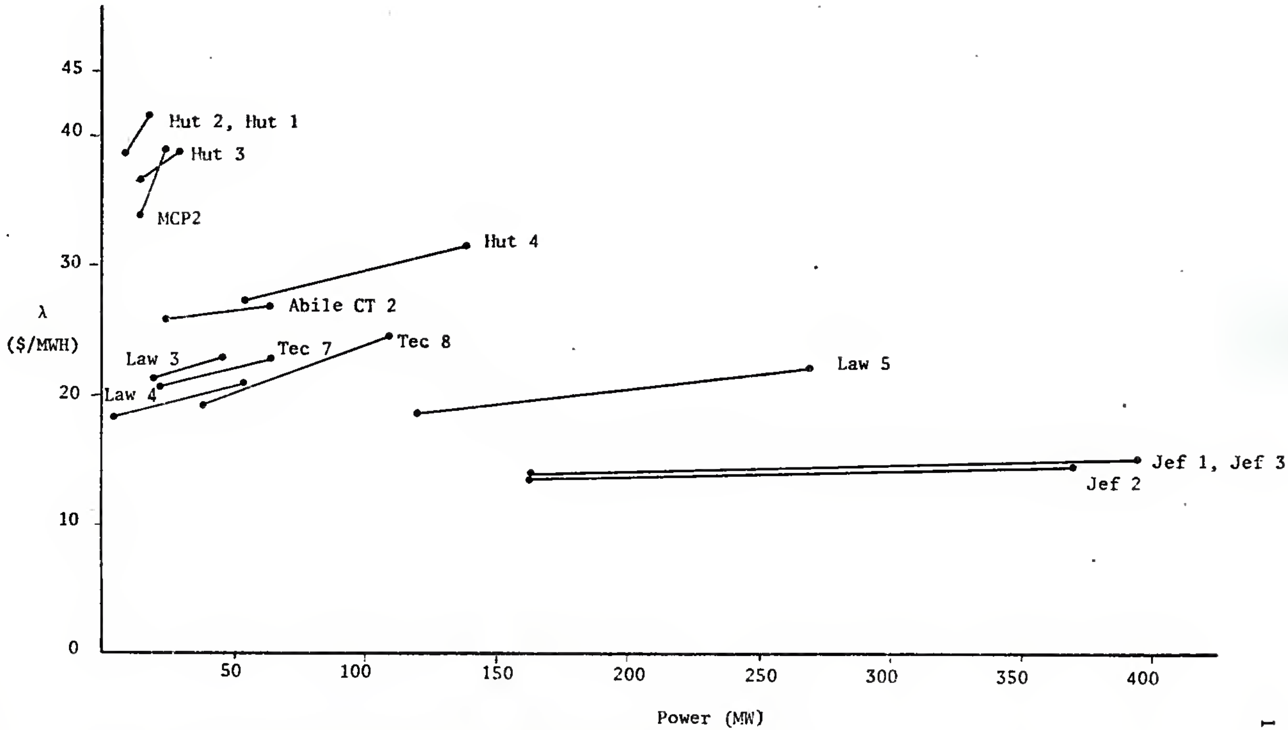

Fig. 5.1 Incremental Costs and Power Production Relationship

Table 5.12: Initial Free Run Results

UNIT #	UNIT NAME	POWER LEVEL (MW)	CANDIDATE VALUES
1	Jef1	449.0936	0
2	Jef2	449.0936	0
3	Jef3	449.0936	0
4	LAW5	-20.51863	2
5	LAW4	-57.28005	2
6	LAW3	-68.60578	2
7	TEC8	-6.96666	2
8	TEC7	-81.25187	2
9	Hut4	-182.1314	2
10	Hut3	-135.9568	2
11	Hut2	-58.15734	2
12	Hut1	-58.15734	2
13	MCP2	-383.3954	2
14	ABILE CT	-21.13727	2

Table 5.13: Case One Results

UNIT	UNIT NAME	INC CST ($/MWH)	POWER LEVEL (MW)	OPR CST ($)
1	JEF1	14.77778	301.4172	4961.527
2	JEF2	14.77778	356.767	5673.129
3	JEF3	14.77778	328.8154	5315.283
4	LAW5	14.77778	120	3284.613
5	LAW4	14.77778	20	743.9843
6	LAW3	14.77778	0	0
7	TEC7	14.77778	0	0
8	HUT4	14.77778	30	1200.339
9	HUT3	14.77778	0	0
10	HUT2	14.77778	0	0
11	HUT1	14.77778	0	0
12	MCP2	14.77778	0	0
13	ABILE CT	14.77778	0	0
TOTALS:		1157	21178.87	

Table 5.14: Case Two Results

UNIT	UNIT NAME	INC CST ($/MWH)	POWER LEVEL (MW)	OPR CST ($)
1	JEF1	20.18953	405	6521.186
2	JEF2	20.18953	405	6392.056
3	JEF3	20.18953	405	6456.62
4	LAW5	20.18953	187	4585.725
5	LAW4	20.18953	55	1438.595
6	LAW3	20.18952	0	0
7	TEC7	20.18953	0	0
8	HUT4	20.18953	0	0
9	HUT3	20.18953	0	0
10	HUT2	20.18953	0	0
11	HUT1	20.18953	0	0
12	MCP2	20.18953	0	0
13	ABILE CT	20.18953	0	0
TOTALS		1457	25394.18	

Table 5.15: Case Three Results

UNIT	UNIT NAME	INC CST ($/MWH)	POWER LEVEL (MW)	OPR CST ($)
1	JEF1	18.97142	405	6521.186
2	JEF2	18.97142	405	6392.056
3	JEF3	18.97142	405	6456.62
4	LAW5	18.97142	134	3547.961
5	LAW4	18.97142	30	936.6074
6	LAW3	18.97142	0	0
7	TEC7	18.97142	0	0
8	HUT4	18.97142	20	942.8276
9	HUT3	18.97142	0	0
10	HUT2	18.97142	0	0
11	HUT1	18.97142	0	0
12	MCP2	18.97142	0	0
13	ABILE CT	18.97142	0	0
TOTALS		1399	24797.26	

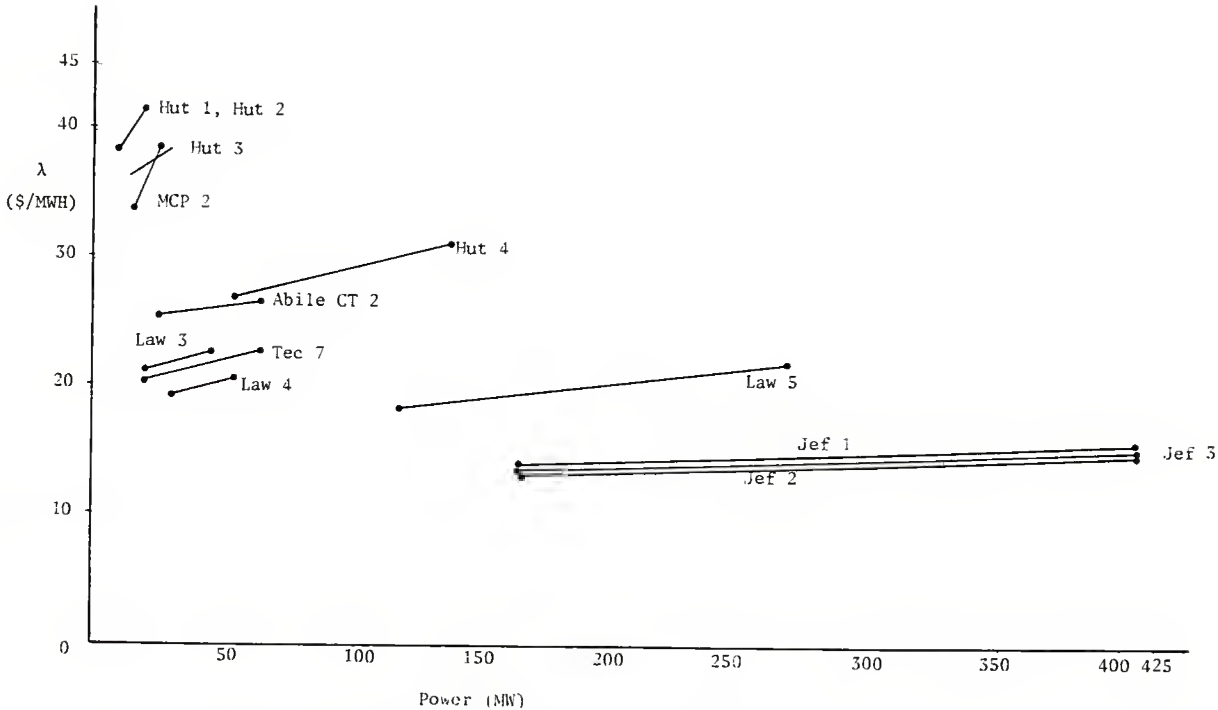

Fig. 5.2 Incremental Costs and Power Production Relationship

Table 5.16: Given KPL Data (From Reference 26)

Unit	α (MBtu/H)	β (MBtu/MW^{2}H)	$\gamma(10^3)$ (MBtu/MW^{2}H)	C ($/MBtu)	Power MIN (MW)	Power MAX (MW)	Lambda MIN $/MWH	Lambda MAX $/MWH
J1	518.798	9.070	1.86	1.45	165	405	14.042	15.336
J2	508.525	8.891	1.82	1.45	165	405	13.764	15.033
J3	513.662	8.981	1.84	1.45	165	405	13.903	15.185
L5	550.969	7.223	5.22	2.20	120	270	18.650	22.097
L3	78.404	9.019	14.50	2.20	20	45	21.131	22.742
T7	110.335	8.786	11.32	2.22	20	65	20.510	22.771
L4	169.431	8.225	10.60	2.20	30	55	19.496	20.663
H4	152.679	8.449	8.60	2.90	55	140	27.247	31.487
H3	32.258	11.906	24.20	2.90	15	30	36.636	38.744
H2	15.514	12.120	58.50	2.90	10	19	38.541	41.594
H1	15.514	12.120	58.50	2.90	10	19	38.541	41.594
MCP2	34.699	9.031	89.00	2.90	15	25	33.832	38.927
ABILE CT 2	232.100	8.453	4.30	2.99	25	65	25.917	26.946

Table 5.17: Power Demanded Exceeds Maximum Power Production

With all units MX CAP. one must buy 497 MW of power.

UNIT	UNIT NAME	INC CST ($/MWH)	POWER (MW)
1	JEF1	0	395
2	JEF2	0	370
3	JEF3	0	395
4	LAW5	0	270
5	LAW4	0	55
6	LAW3	0	45
7	TEC8	0	110
8	TEC7	0	65
9	HUT4	0	140
10	HUT3	0	30
11	HUT2	0	19
12	HUT1	0	19
13	MCP2	0	65
14	ABILE CT	0	25

Total Generaged Power = 2003 MW

Note: 2003 MW + 497 MW = 2500 MW

2500 MW = P^{tot}

Table 5.18: Power Sell Situation

With all units at MW CAP. one must sell 45 MW of Power.

UNIT (#)	UNIT NAME	INC CST ($/MBtu)	POWER (MW)	HT RATE (BTU/KWH)	OPR CST ($)
1	JEF1	0	165	12521.50	2995.77
2	JEF2	0	165	12273.55	2936.448
3	JEF3	0	165	12397.53	2966.109
4	LAW5	0	120	12441.72	3284.613
5	LAW4	0	5	42164.12	463.8053
6	LAW3	0	20	13232.04	582.2057
7	TEC7	0	40	11996.89	1065.324
8	HUT4	0	20	16255.65	942.8276
9	HUT3	0	55	13825.26	2205.13
10	HUT2	0	15	14031.73	610.3801
11	HUT1	0	10	14256.43	413.4364
12	MCP2	0	10	31706.1	948.0122
13	ABILE CT	0	25	12614.87	914.5779

Total generated power = 185 MW

Note: 815 MW - 45 MW = 770 MW

770 MW = P^{tot}

Table 5.19: Case Six Results

UNIT (#)	UNIT NAME	INC CST ($/MWH)	POWER (MW)
1	JEF1	15.10452	390.0001
2	JEF2	15.10452	370
3	JEF3	15.10452	390.0001
4	LAW5	15.10452	120
5	LAW4	15.10452	0
6	LAW3	15.10452	0
7	TEC8	15.10452	0
8	TEC7	15.10452	0
9	HUT4	15.10452	0
10	HUT3	15.10452	0
11	HUT2	15.10452	0
12	HUT1	15.10452	0
13	MCP2	15.10452	0
14	ABILE CT	15.10452	0

Total generated power = 1270 MW

Table 5.20: Case Seven Results

UNIT (#)	UNIT NAME	INC CST ($/MWH)	POWER (MW)
1	JEF1	14.92651	356.6669
2	JEF2	14.92651	356.6669
3	JEF3	14.92651	356.6669
4	LAW5	14.92651	120
5	LAW4	14.92651	5
6	LAW3	14.92651	20
7	TEC8	14.92651	40
8	TEC7	14.92651	20
9	HUT4	14.92651	55
10	HUT3	14.92651	15
11	HUT2	14.92651	10
12	HUT1	14.92651	10
13	MCP2	14.92651	25
14	ABILE CT	14.92651	0
Total generated power = 1390.001 MW			

Table 5.21: Case Eight Results

UNIT (#)	UNIT NAME	INC CST ($/MWH)	POWER (MW)
1	JEF1	15.03776	377.5002
2	JEF2	15.03776	370
3	JEF3	15.03776	377.5002
4	LAW5	15.03776	120
5	LAW4	15.03776	5
6	LAW3	15.03776	20
7	TEC8	15.03776	40
8	TEC7	15.03776	20
9	HUT4	15.03776	55
10	HUT3	15.03776	15
11	HUT2	15.03776	10
12	HUT1	15.03776	10
13	MCP2	15.03776	25
14	ABILE CT	15.03776	0

Total generated power = 1445 MW

Table 5.22: Case Nine Results

UNIT (#)	UNIT NAME	INC CST ($/MWH)	POWER (MW)
1	JEF1	20.04855	395
2	JEF2	20.04855	370
3	JEF3	20.04855	395
4	LAW5	20.04855	180.8659
5	LAW4	20.04855	41.83691
6	LAW3	20.04855	20
7	TEC8	20.04855	53.29732
8	TEC7	20.04855	20
9	HUT4	20.04855	55
10	HUT3	20.04855	15
11	HUT2	20.04855	10
12	HUT1	20.04855	10
13	MCP2	20.04855	25
14	ABILE CT	20.04855	0
Total generated power = 1591 MW			

Table 5.23: Case Ten Results

UNIT (#)	UNIT NAME	INC CST ($/MWH)	POWER (MW)
1	JEF1	18.92875	395
2	JEF2	18.92875	370
3	JEF3	18.92875	395
4	LAW5	18.92875	132.1433
5	LAW4	18.92875	17.85674
6	LAW3	18.92875	20
7	TEC8	18.92875	40
8	TEC7	18.92875	20
9	HUT4	18.92875	55
10	HUT3	18.92875	0
11	HUT2	18.92875	0
12	HUT1	18.92875	0
13	MCP2	18.92875	0
14	ABILE CT	18.92875	0
Total generated power = 1445 MW			

Table 5.24: Case Eleven Results

UNIT (#)	UNIT NAME	INC CST ($/MWH)	POWER (MW)
1	JEF1	16.12561	395
2	JEF2	16.12561	370
3	JEF3	16.12561	395
4	LAW5	16.12561	120
5	LAW4	16.12561	5
6	LAW3	16.12561	20
7	TEC8	16.12561	40
8	TEC7	16.12561	20
9	HUT4	16.12561	55
10	HUT3	16.12561	0
11	HUT2	16.12561	0
12	HUT1	16.12561	0
13	MCP2	16.12561	0
14	ABILE CT	16.12561	0

Total generated power = 1420 MW

Table 5.25: Savings of CLD with Must-Run Units Over Present KPL Technique

A) Average Daily Savings $\cong$ \$4000/day
(From Tables 5.9 and 5.10)

B) Monthly Savings $\cong$ \$4000(30) $\cong$ \$120,000/mo.

C) Annual Savings $\cong$ \$120,000 (12) $\cong$ 1.44 million

Table 5.26: Savings Using CLD with no Must-Run Units Over Present KPL Technique

A) Average Daily Savings ≅ 38000/day
(From Tables 5.9 and 5.10)

B) Monthly Savings ≅ \$38000 x 30 = 1,140,000/mo

C) Annual Savings ≅ 1.1400000 x 12 = \$13,680,000/yr

Table 5.27: EPRI, CLD (Must-Run), CLD (Free-Run) Comparative Results ($)

1/1/85	1/2/85
EPRI-CLD (Must-Run)	EPRI-CLD (Must-Run)
478994.00 - 477014.08 = 1979.92	521167.00 - 520805.02 = 361.98
EPRI - CLD (Free-Run)	EPRI-CLD (Free-Run)
478994.00 - 425194.02 ≅ 53800	521167.00 - 487558.13 ≅ 33600

Savings (dollar)

	CLD (Must-Run) Over EPRI	CLD (Free-Run) Over EPRI
bi-day	2350	87,400
week	8200	305,900
month	32800	1,223,600
year	393500	14,683,200

6.0 Areas for Extended Research

There are several other areas which could be studied to enhance the lambda dispatch method for use by Kansas Electric Utilities. This chapter is dedicated to the discussion of these areas - specifically the areas of hydro power, nuclear power, start-up costs, and transmission losses.

6.1 Hydro Power

In looking at the fuel types used in the data for this research, one will notice that hydro power was seemingly overlooked. Kansas Power and Light nor any Kansas electric utility use hydro power as an electrical power supplier to any major extent. There are some contractual obligations with the U.S. Corps of Engineers for a small amount of hydro power. However, this amounts to only a small portion of the Kansas electric utilities' power supply.

Other states do use sizeable quantities, e.g., Colorado and other mountain states. For this reason, one suggested branch for further research is hydro power and its affect on the lambda dispatch program developed in this research. The use of hydro power could effect the fuel cost and efficiency (heat rate) scenario in terms of restructuring the order of candidate units as well as the candidate plot. This kind of restructuring could subsequently effect which units should be used at given power demands. These thoughts alone warrant further research into the inclusion of hydro power.

6.2 Nuclear Power

As with hydro power, nuclear power was not considered as a fuel type in this research. The reason was also that KPL nor any other Kansas utility presently uses nuclear power or nuclear power units to meet electrical energy demand other than that purchased from surplus capacity in Nebraska and Arkansas. However, other states, e.g., California, Iowa, and Pennsylvania, have used electricity produced from nuclear power. Even the state of Kansas is to have a nuclear facility that is available for commercial production by summer 1985.

Nuclear fuel is, in general, much less expensive than any other fuel type, e.g., natural gas or coal. Thus, a nuclear power plant in a system with coal and gas-fired units would run almost constantly with the fossil units used to meet peak or heavy demands. As was suggested when considering hydro power, the use of nuclear power could change the entire scenario of which units to use to meet specific power demands. Further research is suggested here.

6.3 Start-Up Costs

Start-up costs were not considered in this research because of the assumption stated in Chapter 4, i.e., no unit is ever shut down completely if it is a viable candidate. Realizing the unrealistic nature of this assumption with the possibilities of equipment failure and routine maintenance one can realize that start-up costs may have a definite affect on which decisions should be made, i.e., which units should be used, and in what order should units be brought up. Since these types of decisions, especially when considering start-up costs, have a direct affect on the cost, one can easily see the importance of this area. Again, further research is suggested.

6.4 Transmission Losses

This area is, if not the most important, one of the more important areas that was not considered. As discussed earlier (Chapter 4) making the same assumption that KPL does (thus the assumption used in this research) that all system power losses can be matched by producing eight percent above the actual power demand takes away realism that may be desired.

As discussed in Chapter 1, significant in-roads to transmission losses and their effect upon overall systems have been made since the 1930's. The presence of transmission losses does make the lambda dispatch problem much more difficult. One has to consider which unit is the source, where the demand is with respect to the source, efficiency of the transmission line in addition to the unit efficiency, the type of line being used, and the impedance of the line just to name a few of the variables.

In view of its complexities, study of this area and its effect on the IBM:PC version of the lambda dispatch may prove time consuming and cumbersome, but it could also prove very beneficial in adding to the realistic nature of the computer program.

Acknowledgements

I would like to take this opportunity to thank all the people involved in helping me with this thesis work. To the members of my committee (Dr. Gale Simons, Dr. Frank Tillman, and Dr. Stanley Lee) I give my thanks. Dr. Simons for serving on my committee, Dr. Tillman for his suggestions and help, and Dr. Lee for his continued support as my major professor. A special thanks to Dr. Lee for the patience and concern he expressed throughout the duration of my completing this work.

Also, a note of thanks to Robert Fackler of KPL. His supply of actual data made it possible to make decisions from comparable results dealing with real data. It was appreciated.

Finally, a very special thank you to one fantastic person, Dr. N. Dean Eckhoff (Dear Dad). The support he gave, the ideas he presented as well as work done, opportunity for discussion of problems and suggestions he made were not just deeply appreciated, these things were essential to complete this work in the time frame given. The feelings of gratitude run deeper than can be expressed here so I simply say thanks -- from the bottom of my computer output and scratched up outlines. I hope that this work meets with his approval.

References

1) H. H. Happ, "Optimal Power Dispatch - A Comprehensive Study", I.E.E.E. Trans. on Power App. Syst., Vol. PAS-96, No. 3, May/June, 1977, pp. 841-850.

2) M. J. Steinberg and T. H. Smith, "Economy Loading of Power Plants and Electric Systems", John Wiley & Sons, Inc., 1943.

3) E. E. Ceorge, "Intrasystem Transmission Losses", A.I.E.E. Trans., Vol. 62, March 1943, pp. 153-158.

4) J. B. Ward, J. R. Eaton, and H. W. Hale, "Total and Incremental Losses in Power Transmission Networks", A.I.E.E. Trans., Vol. 69, Part I, 1950, pp. 626-631.

5) E. E. Ceorge, H. W. Page, and J. B. Ward, "Coordination of Fuel Cost and Transmission Loss by Use of the Network Analyzer to Determine Plant Loading Schedules", A.I.E.E. Trans., Vol. 68, Part II, 1949, pp. 1152-1160.

6) R. H. Travers, D. C. Harker, R. W. Long, and E. L. Harder, "Loss Evaluation - Part III", A.I.E.E. Trans., Vol. 73, Part III, 1954, pp. 1091-1104.

7) L. K. Kirchmayer and C. W. Stagg, "Evaluation of Methods of Coordinating Incremental Fuel Costs and Incremental Transmission Losses", A.I.E.E., Trans., Vol. 71, Part III, 1952, pp. 513-520.

8) C. Kron, "Tensorial Analysis of Integrated Transmission Systems", Part I, Part II, A.I.E.E. Trans., Vol. 70, Part I, 1951, pp. 1239-1248; Vol. 71, Part III, 1952, pp. 502-512.

9) J. B. Ward, "Economy Loading Simplified", A.I.E.E. Trans., Vol. 72, Part III, 1953, pp. 1306-1311.

10) P. G. Lowery, "Generating Unit Commitment by Dynamic Programming", I.E.E.E. Trans. on PAS., Vol. 85, No. 5, May 1966, pp. 422-426.

11) H. P. Van Meeteren, "Scheduling of Ceneration and Allocation of Fuel, Using Dynamic and Linear Programming", I.E.E.E. Trans. on PAS, Vol. 103, No. 7, July 1984, pp. 1562-1568.

12) H. W. Dommel and W. F. Tinney, "Optimal Power Flow Solutions", I.E.E.E. Trans. on PAS Vol. 87, 1968, 1866-1876.

13) O. Alsac and B. Stott, "Optimal Load Flow with Steady-State Security", I.E.E.E. Trans. Power App. Syst., Vol. PAS-93, May/June 1974, pp. 745-751.

14) I. Megahed, N. Abou-Taleb, M. Iskandrani, and A. Moussa, "A Modified Method for Solving the Economic Dispatching Problem", I.E.E.E. Trans. Power App. Syst., Vol. PAS-96, No. 1, Jan/Feb 1977, pp. 124-133.

15) H. Nicholson and M. J. H. Sterling, "Optimum Dispatch of Active and Reactive Generation by Quadratic Programming", I.E.E.E. Trans. Power App. Syst., Vol. PAS-92, Mar/Apr 1973, pp. 644-654.

16) G. F. Reid and L. Hasdorff, "Economic Dispatch Using Quadratic Programming", I.E.E.E. Trans. Power App. Syst., Vol. PAS-92, Nov/Dec 1973, pp. 2015-2023.

17) A. M. Sasson and H. M. Merrill, "Some Applications of Optimization Techniques to Power Systems Problems", Proc. of the I.E.E.E., Vol. 62, No. 7, 1974, pp. 959-972.

18) R. Lugtu, "Security Constrained Dispatch," I.E.E.E. Trans. on Power App. Syst., Vol. PAS-98, No. 1, Jan/Feb 1979, pp. 270-275.

19) S. A. Arafeh and A. P. Sage, "Hierarchial Estimation Identification and Power Systems Engineering", M. Singh, eds., Large Scale Syst. Eng. Appl., North-Holland Pub. Company, 1979, pp. 421-449.

20) F. J. Trefny and K. Y. Lee, "Economic Fuel Dispatch", I.E.E.E. Trans. on Power App. Syst., Vol. PAS-100, No. 7, July 1981, pp. 3468-3477.

21) L. Roy and N. D. Rao, "A New Algorithm for Real-Time Optimal Dispatch of Active and Reactive Power Generation Retaining Nonlinearity", I.E.E.E. Trans. on Power App. Syst., Vol. PAS-102, No. 4, April 1983, pp. 832-842.

22) A. I. Cohen and M. Yoshimura, "A Branch-and-Bound Algorithm for Unit Commitment", I.E.E.E. Trans. on Power App. Syst., PAS-102, No. 2, February 1983, pp. 444-451.

23) S. Vemuri, A. B. Ranjit Kumar, D. F. Hackett, J. Eisenhauer, and R. Lugtu, "Fuel Resource Scheduling, Part I - Overview of an Energy Management Problem", I.E.E.E. Trans. on Power App. Syst., Vol. PAS-103, No. 7, July 1984, pp. 1542-1548.

24) S. Vemuri, A. B. Ranjit Kumar, D. F. Hackett, J. Eisenhauer, and R. Lugtu, "Fuel Resource Scheduling, Part 2...", I.E.E.E. Trans. on Power App. Syst., Vol. PAS-103, No. 7, July 1984.

25) M. E. El-Hawary and G. S. Christenson, Optimal Economic Operation of Electric Power Systems, Academic Press, Inc. 1979, pg. 124.

26) L. K. Kirchmayer, Economic Operation of Power Systems, John Wiley & Sons, 1958.

27) R. Fackler, System Engineer, KPL, Personal Conversation and Letter, January 3 and 4, 1985

28) I. S. Sokolnikoff, R. M. Redheffer, Mathematics of Physics and Modern Engineering, McGraw-Hill Inc. 1966, pp. 341-342.

Appendix 1: Lagrangian Theory (Reference 28)

Assume that a multi-variable function (u) exists for which the minimum or maximum extreme is desired such that:

$$u = f(x,y,z), \tag{1}$$

and this function is limited by an equation constraint ω such that:

$$\omega = g(x,y,z) = 0 \tag{2}$$

(Assume that the extreme values for function u satisfy ω as well.) Solve ω for one variable, i.e., z, in terms of the other variables. Thus,

$$z = h(x,y), \tag{3}$$

and substitute into function u.

$$u = f(x,y,h(x,y)) = F(x,y) \tag{4}$$

This new function may be difficult to handle or the partial derivative with respect to x and/or y may be unwieldy. Therefore, Lagrange developed a theroem (technique) called Lagrange multipliers which develops the function u extreme by assuming variable values to be such that the total differential of u vanishes.

$$df = \frac{\partial f}{\partial x}\,dx + \frac{\partial f}{\partial y}\,dy + \frac{\partial f}{\partial z}\,dz, \tag{5}$$

and with $\omega = 0$ expression (6) is developed.

$$dg = \frac{\partial g}{\partial x}\,dx + \frac{\partial g}{\partial y}\,dy + \frac{\partial f}{\partial z}\,dz = 0 \tag{6}$$

Multiply dg by a Lagrange multiplier (λ) and add to df,

$$df + \lambda dg = \left(\frac{\partial f}{\partial x} + \lambda \frac{\partial g}{\partial x}\right) dx + \left(\frac{\partial f}{\partial y} + \lambda \frac{\partial g}{\partial y}\right) dg + \left(\frac{\partial f}{\partial z} + \lambda \frac{\partial g}{\partial z}\right) dz = 0 \qquad (7)$$

Assume x and y to be independent variables and

$$\frac{\partial g}{\partial z} \neq 0 \qquad (8)$$

at function u extremes. Thus, find a λ value such that

$$\frac{\partial f}{\partial z} + \lambda \frac{\partial g}{\partial z} = 0 \qquad (9)$$

Therefore,

$$\left(\frac{\partial f}{\partial x} + \lambda \frac{\partial g}{\partial x}\right) dx + \left(\frac{\partial f}{\partial y} + \lambda \frac{\partial g}{\partial y}\right) dy = 0 \qquad (10)$$

Hence,

$$\frac{\partial f}{\partial x} + \lambda \frac{\partial g}{\partial x} = 0 \qquad (11)$$

and

$$\frac{\partial f}{\partial y} + \lambda \frac{\partial g}{\partial y} = 0 \qquad (12)$$

Equations (2), (9), (11), and (12) form a series of four equations and four unknowns (x,y,z,λ). The solution of this set causes Eqn. (1) to be an extreme constrained by Eqn. (2).

Appendix 2: CLD Computer Listing*

```
10 CLS
20 REM ################################################################   Section 1
30 REM THIS IS A SCALEO DDWN MDDEL DF THE LAMBDA DISPATCH PRDCESS USED
40 REM BY KP&L.  THIS WILL INITIALLY DETERMINE WHICH UNITS ARE USED AND
50 REM WHICH PDWER VALUES ARE PRDDUCED. FURTHER ADDITIDNS WILL BE MADE.
60 REM ################################################################
70 REM DEFINITIDN STATEMENTS:
80 REM UQNT: TDTAL AMT. DF UNITS        ALPH(I): CDNST. TERM DF EQ FDR UNIT I
90 REM I: UNIT NUMBER                   BET(I): LIN. TERM DF EQ FDR UNIT I
100 REM J: HDUR NUMBER                  GAM(I): QUAD. TERM DF EQ FOR UNIT I
110 REM PMX(I): MAX POWER GEN.          CST(I): CDST CDEFFICIENT FDR UNIT I
120 REM P(I,j): PDWER OF UNIT I IN HR J PDEM: POWER DEMANDEO
130 REM LAMBVAL(J):PRES INCR CST VAL    EFF(I): PLANT EFFICIENCY RATE
140 REM LAMBMX(I): MX. UNIT I LMB. VAL  NDRM(I): NDRMALIZING PEAK VALUES
150 REM LAMBMN(I): MN. UNIT I LMB. VAL  PK: PEAK VALUES FDR EACH DAY (GIVEN)
160 REM HRS: # DF HOURS MDDELED         PDIF: PDWER SOLD
170 REM K: INCREMENTING VALUE           UNAM(I): UNIT NAME
180 REM Y: YES  N: ND                   ANS: ANSWER (Y/N)
190 REM LAMB(I): LAMBDA VAL.DF UNIT I
200 REM ################################################################   Section 2
210 REM DIMENSIDN VARIABLES: 26 UNITS IN STATE DF KANSAS
220 DIM PMX(26),PMN(26),PD(3),P(26,3),EFF(26),NDRM(3),PDMD(3),CAND(15)
230 DIM LAMBMX(26),LAMBMN(26),ALPH(26),BET(26),GAM(26),CST(26),DUMY(14)
240 DIM W(26),UNUSD(26,3),PLUSD(26,3),ULBVAL(26,3),NTMXM(26),NOTMXM(26)
250 DIM LAMBVAL(26),GTRM(26),MNM(26),MATCH(26),PVAL(3),PMXOMY(26),THRU(15)
260 DIM UNLVL(14),HELP(14),DRDR(15),UNT(15),USET(15),USET1(15),CSET1(15)
270 DIM HTRT(19),HRTCST(19)
280 REM ################################################################   Section 3
290 REM INITIALIZE VARIABLES:
300 I = 1   :   J = 0    :        K = 0    :       PK = 0
310 PMX(I) = 0 :PMN(I) = 0  :   P(I,J)=0 :       PDIF = 0
320 HRS = 0    :LAMBVAL(J)=0:  LAMBMX(I) = 0 : LAMBMN(I) = 0
330 NDRM(I) = 0:EFF(I) = 0  :   PDEM = 0      : PVAL = 0
340 CST(I) = 0 :GAM(I) = 0  :   BET(I) = 0    : ALPH(I) = 0
350 HRS = 0    :UNAM(I) = 0 :   Y = 0    :       N = 0
360 ANS = 0    :UQNT = 0    :   LAMB(I) = 0   : ULBVAL(I,J) = 0
370 TRYAGN = 0 :PIN(I,J) = 0:   TOTHCST = 0
380 REM ################################################################   Section 4A
390 REM INPUTTING DATA:
400 READ UQNT
410 DATA 13
420 PRINT "THERE ARE",UQNT,"UNITS AVAILABLE FDR USE."
430 'PRINT "?",13
440     FDR I = 1 TO UQNT
450             READ PMN(I),PMX(I),ALPH(I),BET(I),GAM(I),CST(I)
460     NEXT I
470 REM ACTUAL DATA: ----------------------------------------------    Section 5A
480 DATA    165,   405,518.7981,9.070396,.00185984,1.45
490 DATA 165,405,508.5249,8.890785,.00182302,1.45
500 DATA 165,405,513.6615,8.980591,.00184143,1.45
510 DATA 120,270,550.9698,7.223485,.00522346,2.20
520 DATA 30,55,169.4305,8.224950,.01061296,2.20
530 DATA 20,45,78.4038,9.018903,.01464727,2.20
540 'DATA 40,110,201.4463,7.187002,.01729802,2.22
550 DATA 20,65,110.3347,8.785682,.01132096,2.22
560 DATA 55,140,152.6788,8.449719,.00859942,2.90
570 DATA 15,30,32.2580,11.906040,.02423113,2.90
580 DATA 10,19,15.5141,12.120150,.05848686,2.90
590 DATA 10,19,15.5141,12.120150,.05848686,2.90
```

*See Appendix 3 for supplementary variable definitions.

```
600 DATA 25,65,232.0997,8.453138,.00429833,2.99
610 'DATA 20,45,161.6832,9.654569,-.0039202,2.90
620 'DATA 20,45,161.6832,9.654569,-.0039202,2.90
630 'DATA 20,45,161.6832,9.654569,-.0039202,2.90
640 DATA 15,25,34.6990,9.030818,.0878436O,2.90
650 'DATA 20,45,161.6832,9.654569,-.0039202,2.90
660 'DATA 20,45,161.6832,9.654569,-.0039202,2.90
670 REM ****************************************************************
680 REM ****************************************************************          Section 4B
690 REM DATA CHECK:
700 PRINT "LAST INPUT"
710 PRINT "****************************************************************"
720 PRINT "       Power   Power                                      Fuel"
730 PRINT "Unit   Max     Min     Alph      Beta    Gamma       Cost"
740 PRINT "****************************************************************"
750     FOR I = 1 TO UCNT
760             PRINT I;PMN(I);PMX(I);ALPH(I);BET(I);GAM(I);CST(I)
770     NEXT I
780 ANS=0
790 INPUT "ANY CHANGES: Y=1;N=0";ANS
800 IF (ANS = 0) THEN GOTO 830
810 PRINT "TYPE: LIST 480-660 THEN HIT RETURN TO MAKE APPROPRIATE CHANGES"
820 PRINT "DEPRESS KEY F2 AFTER CHANGES HAVE BEEN MADE TO RERUN THE PROGRAM"
825 STOP
830 ANS = 0
840 INPUT "IS A LIST OF UNIT EFF. RT, MIN AND MAX LAMBDA VALUES DESIRED: Y=1,N=0
";ANS
850 REM ****************************************************************          Section 6
860 REM SUBROUTINE EFFICIENT
870 I = 1
880 IF (ANS = 0) THEN GOTO 920
890 PRINT "UNIT          UNIT                UNIT LAMBDA"
900 PRINT "NUM          EFFRATE       MAXIMUM           MINIMUM"
910 PRINT "-------------------------------------------------"
920     FOR I = 1 TO UCNT
930             EFF(I) = 3413/(BET(I) * 1000)
940             LAMBMX(I)=(CST(I)*BET(I))+(2*CST(I)*GAM(I)*PMX(I))
950             LAMBMN(I)=(CST(I)*BET(I))+(2*CST(I)*GAM(I)*PMN(I))
960             IF (LAMBMN(I) > LAMBMX(I)) THEN LAMBDMY = LAMBMX(I)
970             IF (LAMBMN(I) > LAMBMX(I)) THEN LAMBMX(I) = LAMBMN(I)
980             IF (LAMBMX(I) = LAMBMN(I)) THEN LAMBMN(I) = LAMBDMY
990             IF (ANS = 1) THEN PRINT I,EFF(I),LAMBMX(I),LAMBMN(I)
1000    NEXT I
1010 REM ****************************************************************
1020 REM SUBROUTINE SCHEMOEVEL: TIME PER. IS CHOSEN & PRED. DEM. VAL. ARE CALC.
1030 INPUT "HOW MANY HOURS WILL THIS MODEL CONTAIN";HRS
1040 'INPUT "HOW MANY HOURS ARE BEING RUN";HRS
1050 'DATA 3
1060 'PRINT "?",HRS
1070 'READ PK
1080 INPUT"What is the total system demand in MW";PK
1090    PRINT "NRM FCTR       PK VL      PRD DEM        HRS RN"
1100    PRINT "**********************************************"
1110 FOR J = 1 TO HRS
1120    READ NORM(J)
1130    PD(J) = NORM(J) * PK
1140    PRINT NORM(J), PK, PD(J), HRS
1150 NEXT J
1160 DATA 1.00
```

Section 5B

```
1170 PRINT "****************************************************"
1180 REM ***************************************************
1190 REM MAIN: MAIN BDOY DF PRDGRAM-DETERMINES NHICH INTERATIVE PRDCESS
1200 REM       TD FDLLDW - DONE FOR EVERY hour of every day: pd(j)
1210 REM ***************************************************
1220 FDR J = 1 TD HRS
1230 REM DETERMINE WHICH UNITS ARE DN-LINE & AT WHAT PROD. LEVEL
1240     PRINT "--------  MUST RUN UNITS AND PDWER VALUES  ------"
1250 '   PRINT "PLEASE INPUT IN PRIORITY DRDER"
1260      FDR I = 1 TD UQNT
1270          READ UNUSD(I,J),PLUSO(I,J)
1280          IF (UNUSD(I,J) > 0) THEN P(I,J) = PLUSO(I,J)
1290          PRINT I,J,UNUSD(I,J),PLUSD(I,J)
1300          IF (PLUSD(I,J) > 0) THEN P(I,J) = PLUSD(I,J)
1310          IF (P(I,J) > 0) THEN PVAL(J) = PVAL(J) + P(I,J)
1320      NEXT I
1330 J = 3
1340 NEXT J
1350 ANS = 0
1360 INPUT "ANY CHANGES DF MUST-RUN UNIT VALUES: Y=1, N=0";ANS
1370 IF (ANS = 0) THEN GDTO 1430
1380 PRINT"TYPE: LIST 1390-1420, MAKE NECESSARY CHANGES AND HIT F2 FDR RERUN"
1385 STDP
1390 'DATA 1,165,2,165,3,165,4,120,5,5,6,20,7,40,8,20,9,55,,,,,,,,,,
1400 'DATA I,165,2,165,3,165,,,,,,,,,,,,,,,,,
1410 'DATA 1,165,2,165,3,165,4,120,5,5,6,20,7,40,8,20,9,55,10,I5,I1,I0,I2,10,13,
25,,
1420 DATA I,165,2,I65,3,I65,4,120,5,20,,,,,8,30,,,,,,,,,,,,
1430 FDR J = 1 TD HRS
I440 REM DETERMINE THE UNIT LAMBDA FRDM THE INPUTTED UNIT PDWER GEN. LEVEL
1450     FDR I = 1 TD UQNT
1460          IF (UNUSO(1,J) = 0) THEN GOTO I480
I470     ULBVAL(1,J) = (BET(1) * CST(1)) + (2 * CST(I) * GAM(1) * PLUSO(1,J))
1480     NEXT I
I490 NEXT J
1500 GDTD 2300
I510 REM ***************************************************************
I520 'NEXT J
I530 REM ***************************************************************
1540 REM ----------- SUBRDUTINE SYSTEM INCREMENTAL LAMBDA ---------------
1550 FOR J = 1 TO HRS
1560 REM GPRD: SUM OF GAMMA-CDST FUNC.     DEN: THE DENDMINATDR OF SYS.INC LAM.
1570 REM GTRM: DVDND OF GPRD & GM-CST FNC. STRM: PRD. DF GTRM & BET-CST FNCTIDN
1580 IF (1 = 15) THEN GDTO 1590
1590 GPRD = 1: GTRM = 0: DEN = 0: STRM = 0
1600 REM CALCULATE SYS. INC. LAMBDA (LAMBVAL(J))
1610 REM FIND THE SUM DF ALL THE GAMMA-COST FUNCTIDNS
1620 IF (X = 1) THEN GDTD 1630
1630     GPRD = 1
1640     FOR I = 1 TD UQNT
1650 IF (X = 2) THEN GOTD 1700
1660     IF (TRYAGN <> 1) THEN GOTD 1700
1670       IF (P(I,J) <= PLUSD(I,J)) THEN PMN(1) = PLUSD(I,J)
1680       IF (P(I,J) <= PMN(1)) THEN GOTD I730
1690      GDTD 1720
1700        IF (CAND(I) > 0) THEN GDTO 1730
1710        IF (P(1,J) = 0) THEN GDTD 1730
1720        GPRD = GPRD * GAM(1) * CST(1)
1730     NEXT I
```

Subroutine 1

Part A

```
1740 IF (TRY = 12) THEN PRINT "GPRO",GPRO
1750 REM FIND THE DENOMINATOR TERM OF SYS. INC. LAMBDA EXPRESSION      Part B
1760 DEN = 0
1770      FOR I = 1 TO UQNT
1780 IF (X = 2) THEN GOTO 1830
1790      IF (TRYAGN <> 1) THEN GOTO 1840
1800         IF (P(I,J) <= PLUSO(I,J)) THEN PMN(I) = PLUSO(I,J)
1810         IF (P(I,J) <= PMN(I)) THEN GOTO 1870
1820         GOTO 1850
1830            IF (CAND(I) > 0) THEN GOTO 1870
1840            IF ( P(I,J) = 0) THEN GOTO 1870
1850            GTRM(I) = GPRO / (GAM(I) * CST(I))
1860            DEN = DEN + GTRM(I)
1870      NEXT I
1880 IF (TRY = 12) THEN PRINT "GTRM:DEN",GTRM,DEN
1890 IF (X = 1) THEN GOTO 1900
1900 IF (DEN <> 0) THEN GOTO 2060
1910 REM DO THE FOLLOWING PROC WHEN ALL UNIT POWERS = 0 OR PMX
1920 REM FIND THE CHPST UNIT & KEEP TRACK OF TIES IN COST
1930 LMST = 100
1940 FOR I = 1 TO UQNT
1950    IF (P(I,J) <> 0) THEN GOTO 1980
1960    IF (LAMBMN(I) < LMST) THEN LAMBMN(I) = LMST
1970    N = I
1980 NEXT I
1990 REM FOLLOW THESE STEPS IF COSTS ARE = BETWEEN 2 OR MORE UNITS
2000 FOR I = 1 TO UQNT
2010      IF (P(I,J) <> 0) THEN GOTO 2030
2020      IF (LAMBMN(I) = LMST) THEN P(I,J) = PMN(I)
2030 NEXT I
2040 GOTO 1590
2050 GOTO 1630
2060 REM FIND THE SECOND TERM OF THE NUMERATOR OF SYS. LAM. EXPRESSION     Part C
2070  STRM = 0
2080      FOR I = 1 TO UQNT
2090 IF (X = 2) THEN GOTO 2140
2100      IF (TRYAGN <> 1) THEN GOTO 2150
2110         IF (P(I,J) <= PLUSO(I,J)) THEN PMN(I) = PLUSO(I,J)
2120         IF (P(I,J) <= PMN(I)) THEN GOTO 2170
2130         GOTO 2160
2140            IF (CAND(I) > 0) THEN GOTO 2170
2150            IF (P(I,J) = 0) THEN GOTO 2170
2160            STRM = STRM + (BET(I) * CST(I) * GTRM(I))
2170      NEXT I
2180 IF (TRY = 12) THEN PRINT "STRM=",STRM:STOP
2190 REM CALCULATE THE VALUE OF THE SYSTEM INCREMENTAL LAMBDA
2200      IF (POMD = 0) THEN POMD = PD(J)                                  Part D
2210      LAMBVAL(J) = ((2 * POMD * GPRO) + STRM) / DEN
2220 'PRINT " THE VALUE OF THE SYSTEM INCREMENTAL LAMBDA IS",LAMBVAL(J),"AT HOUR
",J,POMD
2230 IF (X = 3) THEN X = 2
2240 IF (TRYAGN <> 1) THEN GOTO 2250
2250 IF (X <> 0) OR (NOCAND = 0) THEN GOTO 2280
2260 NEXT J
2270 TRYAGN = 2
2280 RETURN
2290 REM ----------------------------------------------------------------
2300 REM ****************************************************************     Section 7
2310 FOR J = 1 TO HRS
```

```
2320 ANS = 0
2330 INPUT "IS AN OROEREO LISTING OF UNITS 8Y MIN LAM8OAS OESIREO: Y=1, N=0";ANS

2340 IF (ANS = 0) THEN GOTO 2360
2350 GOSU8 6220: STOP
2360 X = 1: ANS = 0
2370 CLS: PRINT "RUNNING ...."
2380 'make sure all candidate values are set to zero
2390 FOR I = 1 TO UQNT
2400     CANO(I) = 0
2410 NEXT I
2420 'DO INIT SUMMING OF MNMNS ANO PNMXS
2430 TRK = 0                                                        Section 8
2440 FOR I = 1 TO UQNT
2450    IF (PLUSO(I,J) > 0) OR (TRK = 1) THEN GOTO 2510
2460    MNMN = 100
2470     FOR I = 1 TO UQNT
2480         IF (PMN(I) < MNMN) THEN MNMN = PMN(I)
2490     NEXT I
2500    GOTO 2550
2510   IF (TRK = 0) THEN MNMN = PLUSO(I,J)
2520   IF (TRK = 0) THEN GOTO 2540
2530   MNMN = MNMN + PLUSO(I,J)
2540   TRK = 1
2550  PTOTMX = PTOTMX + PMX(I)
2560 NEXT I
2570 IF (MNMN < PO(J)) ANO (PTOTMX > PO(J)) THEN GOTO 2800
2580 IF (MNMN > PO(J)) THEN GOTO 2650
2590 IF (MNMN = PO(J)) THEN GOTO 2670
2600 IF (PTOTMX < PO(J)) THEN PRINT "NITH ALL UNITS AT MX CAP. ONE MUST 8UY",PO(
J)-PTOTMX,"MN OF PONER"
2610 FOR I = 1 TO UQNT
2620     P(I,J) = PMX(I)
2630 NEXT I
2640 GOTO 5710
2650 PRINT "NITH ALL UNITS AT MN CAP. ONE MUST SELL",MNMN-PO(J),"MN OF PONER
2660 TRK = 0
2670 FOR I = 1 TO UQNT
2680     IF (PLUSO(I,J) > 0) THEN TRK = I
2690 NEXT I
2700 IF (TRK = 1) THEN GOTO 2750
2710 FOR I = 1 TO UQNT
2720     IF (PMN(I) = MNMN) THEN P(I,J) = MNMN
2730 NEXT I
2740 GOTO 5710                                                      Section 9
2750 FOR I = 1 TO UQNT
2760     P(I,J) = 0
2770     IF (PLUSO(I,J) > 0) THEN P(I,J) = PLUSO(I,J)
2780 NEXT I
2790 GOTO 5710
2800 GOSU8 1590
2810 IF (X = 1) THEN POORIG = PO(J)
2820 X = 2
2830 REM find the new power settings (new lambda value)
2840 GOSU8 2880
2850 GOTO 2990
2860 '############################################################
2870 '----------------- SU8ROUTINE PONER CALC. -------------------      Subroutine 2
2880 FOR I = 1 TO UQNT
```

```
2890 IF (CAND(I) > 0) THEN GOTO 2950
2900      P(I,J) = (LAMBVAL(J) - (BET(I) * CST(I))) / (2 * CST(I) * GAM(I))
2910      UNLVL(I) = (BET(I) * CST(I)) + (2 * CST(I) * GAM(I) * P(I,J))
2920 IF (P(I,J) < PMN(I)) THEN CAND(I) = 2
2930 IF (P(I,J) < PLUSO(I,J)) THEN CAND(I) = 2
2940 'PRINT I,P(I,J),CAND(I)
2950 NEXT I
2960 RETURN
2970 '-----------------------------------------------------------
2980 '*********************************************************
2990 GOSUB 3020
3000 GOTO 3180
3010 '**********************************************************
3020 '------------------ SUBROUTINE PONER SET ---------------------
3030 'IF (MRK = 1) THEN PRINT "2530: PDMD=",PDMD:STOP
3040 FOR I = 1 TO UQNT
3050      IF (P(I,J) < PMX(I)) OR (CAND(I) = 1) THEN GOTO 3070
3060       PDMD = PDMD - PMX(I) : P(I,J) = PMX(I) : CAND(I) = 1
3070      IF (P(I,J) > PMN(I)) THEN GOTO 3090
3080       P(I,J) = 0 : CAND(I) = 2
3090      IF (P(I,J) < PLUSO(I,J)) THEN P(I,J) = PLUSO(I,J)
3100 IF (MRK = 0) THEN GOTO 3130
3110 'PRINT "2565: PDMD/I/P(I,J)",PDMD,I,P(I,J)
3120 'PRINT "2566: CAND/PMN/PMX",CAND(I),PMN(I),PMX(I):STOP
3130 NEXT I
3140 'IF (MRK = 1) THEN PRINT "2575 - PDMD=",PDMD:STOP
3150 RETURN
3160 '-----------------------------------------------------------
3170 '**********************************************************
3180 GOSUB 3220
3190 GOTO 3350
3200 '*****************************************************************
3210 '--------------------- SUBROUTINE PRODUCTION SET --------------------
3220 PVAL = 0
3230 FOR I = 1 TO UQNT
3240      PVAL = PVAL + P(I,J)
3250 IF (P(I,J) = PMX(I)) THEN CAND(I) = 1
3260 NEXT I
3270 PDMD = PDORIG - PVAL
3280 'PRINT "I    P(I,J)        CAND(I)     PVAL              PDORIG"
3290 FOR I = 1 TO UQNT
3300 '   PRINT I,P(I,J),CAND(I),PVAL,PODRIG
3310 NEXT I
3320 EVAL = PVAL - PDORIG
3330 RETURN
3340 '*****************************************************************
3350  IF (EVAL > 5) THEN GOTO 3570
3360  IF (EVAL < -5) THEN GOTO 3470
3370  IF (EVAL < 1) AND (EVAL > -1) THEN GOTO 3460
3380  PDMD = PDORIG
3390  FOR I = 1 TO UQNT
3400     IF (CAND(I) <> 1) THEN GOTO 3420
3410     PDMD = PDMD - PMX(I): CAND(I) = 1: GOTO 3440
3420     IF (CAND(I) = 0) THEN P(I,J) = .001
3430     IF (PLUSO(I,J) > 0) AND (CAND(I) <> 0) THEN PDMD = PDMD - PLUSO(I,J)
3440 NEXT I
3441 MXM = 0
3442 FOR I = 1 TO UQNT
3443   IF (P(I,J) = 0) THEN GOTO 3445
```

Subroutine 3

Subroutine 4

```
3444    IF (LAMBMX(I) > MXM) THEN MXM = LAMBMX(I)
3445 NEXT I
3446 FDR I = 1 TD UQNT
3447    IF (LAMBMX(I) <> MXM) THEN GDTO 3449
3448    CAND(I) = 0: PDMD = PDMD + P(I,J): P(I,J) = .00I
3449 NEXT I
3450 GDSUB I590: GDSUB 2880
3460 GDTD 5760
3470 REM CHECK TD SEE IF LAMBDA  IS IN FDRBIODEN ZONE *********      Section 10
3480 XX = 0
3490 FDR I = 1 TD UQNT
3500    IF (P(I,J) = PMX(I)) THEN GDTO 3540
3510    IF (ULBVAL(I,J) => LAMBVAL(J)) THEN GDTD 3540
3520    IF (P(I,J) = 0) THEN GDTO 3540
3530    XX = XX + I
3540 NEXT I
3550 'PRINT "************** XX =",XX
3560 IF (XX = 0) THEN GDTD 4050
3570 X = 0
3580 REM DETERMINE IF ANY LAMBVAL (THAT MUST BE DN) > SYS LAMB      Section 11
3590 PDMD = PDDRIG
3600 FDR I = I TO UQNT
3610       IF (CHK = 2) AND (P(I,J) = PMX(I)) THEN CAND(I) = 1
3620       IF (CHK = 2) AND (P(I,J) = PMX(I)) THEN PDMD = PDMD - PMX(I)
3630       IF (CHK = 2) AHD (P(I,J) = PMX(I)) THEN GDTD 3740
3640     IF (P(I,J) = PMX(I)) THEN CAND(I) = 0
3650     IF (P(I,J) = PMX(I)) THEN BOTD 3740
3660     IF (P(I,J) = 0) THEN GDTD 3740
3670     IF (CAND(I) = 1) THEN GDTD 3720
3680     IF (CAND(I) = 2) THEN GDTD 3700
3690     IF (ULBVAL(I,J) < LAMBVAL(J)) THEN GOTO 3740
3700     P(I,J) = PLUSD(I,J)
3710     X = 2
3720     PDMD = PDMD - P(I,J)
3730 'PRINT I,PDMD
3740 NEXT I
3750 'PRINT "I    CAND       P(I,J)       PODRIG          PVAL     PDMD"
3760 FOR I = 1 TD UQNT
3770 '    PRINT I,CANO(I),P(I,J),PDDRIG,PVAL,PDMO
3780 NEXT I
3790 CHK = 2
3800 IF (X = 2) THEN GDSUB I590
3810 IF (X = 2) THEN GDTD 2830
3820 'PRINT " WE MADE IT TO 2869 SCENARID"
3830 X = 0
3840 IF (PVAL <= PODRIG) THEN GDTD 3880
3850 X = 2
3860 GDSUB I590
3870 GDSUB 3220
3880 REM DETERM. IF ANY UNITS W/ LAMB VAL < SYS LAM ARE AT PWR MAX      Section 12
3890 PDMD = POORIG
3900 FDR I = 1 TO UQNT
3910     IF (ULBVAL(I,J) < LAMBVAL(J)) THEN GDTD 3940
3920     IF (P(I,J) = PLUSD(I,J)) THEN PDMD = PDMD - P(I,J)
3930     GOTD 3970
3940     IF (P(I,J) < PMX(I)) THEN GDTD 3980
3950     IF (P(I,J) = PMX(I)) THEN PDMD = PDMD - P(I,J)
3960     X = 2
3970     CANO(I) = I
```

```
3980 NEXT I
3990 'PRINT "I      CANO     P(I,J)       PVAL          POORIG     PONO"
4000 FOR I = I TO UQNT
4010 '    PRINT I,CANO(I),P(I,J),PVAL,POORIG,POMO
4020 NEXT I
4030 IF (X = 2) THEN GOSUB 1590
4040 IF (X = 2) THEN GOTO 2810
4050 GOSUB 4120
4060 IF (NOCANO = I) THEN GOTO 4100
4070 GOSUB 6220
4080 GOSUB 6500
4090 GOTO 2800
4100 GOTO 4940
4110 REN ****************************************************************
4120 REM -------------- SUBROUTINE FORBIDDEN LANBDA ZONE --------------
4130 NOCMN = 100
4140 'PRINT " HELLO  IS IT NE YOURE LOOKING FOR?"
4150 'PRINT "I              CANO(I)"
4160 FOR I = 1 TO UQNT
4170 ' PRINT I,CANO(I)
4180 NEXT I
4190 'STOP
4200 FOR I = 1 TO UQNT
4210      IF (CANO(I) = I) THEN GOTO 4240
4220      IF (CAND(I) = 3) THEN GOTO 4240
4230         IF (LANBNN(I) < NOCNN) THEN NOCMN = LANBNN(I)
4240 NEXT I
4250 FOR I = 1 TO UQNT
4260      IF (CANO(I) = I) THEN GOTO 4320
4270      IF (CANO(I) = 3) THEN GOTO 4320
4280      IF (LANBNN(I) <> NOCNN) THEN GOTO 4320
4290         CANO(I) = 3
4300 '        PRINT I,CANO(I),NOCNN
4310 'STOP
4320 NEXT I
4330 NNTOT = 0
4340 FOR I = 1 TO UQNT
4350      IF (CANO(I) <> 3) THEN GOTO 4390
4360      IF (PMN(I) < PLUSO(I,J)) THEN PMN(I) = PLUSO(I,J)
4370      MNTOT = MNTOT + PNN(I) 'check out - (p(i,j) - ?)
4380       CANO(I) = 3
4390 NEXT I
4400 'PRINT "NNTOT=",NNTOT,"POMO=",PONO:STOP
4410 NOCANO = 0
4420 FOR I = I TO UQNT
4430    IF (CANO(I) = 0) OR (CANO(I) = 2) THEN NOCANO = 1
4440 NEXT I
4450 IF (NOCANO = I) OR (PONO <= (MNTOT + I)) THEN NOCANO = I
4460 IF (NOCANO = I) THEN GOTO 4480
4470 GOTO 4870
4480  IF (PONO > (MNTOT+1)) THEN GOTO 4130
4490 IF (POMO <= (NNTOT+I)) THEN GOTO 4500
4500 MOCMX = 0
4510 TRKR = 0
4520 FOR I = I TO UQNT
4530    IF (CANO(I) <> 3) THEN GOTO 4550
4540    TRKR = TRKR + I
4550 NEXT I
4560 'PRINT "THE HILLS ARE ALIVE...",TRKR:STOP
```

Subroutine 5

Part A

Part B

Part C

```
4570  IF (TRKR > 1) THEN GOTO  4690
4580 'when only one unit is cand, but pmn > pdad
4590 'reset pmn value to plusd value
4600 FOR I = 1 TO UGNT
4610     IF (CAND(I) <> 3) THEN GOTO 4650
4620          PLUSD(I,J) = PMN(I)
4630          P(I,J) = PMN(I)
4640          ULBVAL(I,J) = (BET(I) * CST(I)) + (2*CST(I)*GAM(I)*PLUSD(I,J))
4650 NEXT I
4660 POMO = POORIG
4670 'PRINT "hello again hello"
4680 GOTO 2360
4690 'PRINT "LET ME START BY SAYING - I LOVE YOU!":STOP
4700 FOR I = 1 TO UGNT
4710     IF (CAND(I) <> 3) THEN GOTO 4730
4720     IF (LAMBMN(I) > MOCMX) THEN MOCMX = LAMBMN(I)
4730 NEXT I
4740 FOR I = 1 TO UGNT
4750     IF (CAND(I) <> 3) THEN GOTO 4810
4760     IF (LAMBMN(I) = MOCMX) THEN MNTOT = MNTOT - PMN(I)
4770     IF (LAMBMN(I) = MOCMX) THEN CAND(I) = 2
4780     IF (CAND(I) = 3) THEN CAND(I) = 0
4790     IF (CAND(I) = 0) THEN POMO = POMO + P(I,J)
4800     IF (CAND(I) = 0) THEN P(I,J) = .001
4810 NEXT I
4820 'PRINT "THE CANDIDATES ARE"
4830 FOR I = 1 TO UGNT
4840   IF (CAND(I) > 0) THEN GOTO 4860
4850 ' PRINT I,CAND(I),POMO,P(I,J)
4860 NEXT I
4870 'PRINT "ITS TIME TO MAKE ANOTHER SUBROUTINE -- YEA!":STOP
4880 RETURN
4890 '------------------------------------------------------------------
4900 '******************************************************************
4910 FOR I = 1 TO UGNT
4920  'IF (CAND(I) = 0) THEN P(I,J) .001
4930 NEXT I
4940 GOSUB 1590
4950 GOSUB 2880
4960 FOR I = 1 TO UGNT
4970 '  PRINT "I,CAND(I)",I,CAND(I)
4980 NEXT I
4990 'FIND IF ANY UNIT HAS BEEN SELECTED W/ P(I,J) < PMN(I)
5000 MRK = 0 : MXMN = 0
5010 FOR I = 1 TO UGNT
5020    IF (P(I,J) <> 0) AND (P(I,J) < PMN(I)) AND (CAND(I) <> 1) THEN MRK = 1
5030 NEXT I
5040 'PRINT "MRK=",MRK:STOP
5050 IF (MRK = 1) THEN GOTO 5580
5060 GOTO 5290
5070 'FOR I = 1 TO UGNT
5080 '    IF (PLUSD(I,J) > 0) AND (P(I,J) < PLUSD(I,J)) THEN P(I,J) = PLUSD(I,J)
5090 '    IF (CAND(I) = 1) OR (P(I,J) => PMN(I)) OR (P(I,J) = 0)
          OR (P(I,J) = PLUSD(I,J)) THEN GOTO 4750
5100 '    IF (LAMBMN(I) > MXMN) THEN MXMN = LAMBMN(I)
5110 '      MRK = I
5120 'NEXT I
5130 'PRINT "MXMN=",MXMN,"MRK=",MRK:STOP
5140 'IF (MRK = 0) THEN GOTO 4970
```

Part D

Section 13

Section 14

```
5150 'POMO = 0 ' FIND MX LAMBMN OF DISP UNITS BELOW PMN
5160 'PRINT "I MADE IT TO 4305":STOP
5170 'FOR I = 1 TO UQNT
5180 '    IF (LAMBMN(I) <> MXMN) THEN GOTO 4850
5190 '    CAND(I) = 2 : PDMD = PDMO + P(I,J) : P(I,J) = 0
5200 'PRINT I,P(I,J),CAND(I):STOP
5210 '    I = 14
5220 'NEXT I
5230 'COND. IS TRUE - COMPLETE ADJ. & REDISP
5240 'FOR I = 1 TO UQNT
5250 '   IF (CAND(I) > 0) THEN GOTO 4900
5260 '   POMO = POMO + P(I,J) : P(I,J) = .001
5270 'NEXT I
5280 'GOTO 4570
5290 MRK = 0
5300 FOR I = 1 TO UQNT
5310       IF (P(I,J) <> 0) AND (P(I,J) < PMN(I)) AND (CAND(I) <> 1) THEN MRK = I
5320 NEXT I
5330 IF (MRK = 1) THEN GOTO 5580
5340 IF (THRU = 5) THEN GOTO 5580
5350 FOR I = 1 TO UQNT                                                    Section 15
5360 'FIND UNIT THAT WAS LAST CAND
5370     IF (CAND(I) <> 0) THEN GOTO 5480
5380 'PRINT "I,P(I,J),POMO",I,P(I,J),POMO:STOP
5390    UNLVL = UNLVL(I)
5400       FOR I = I TO UQNT
5410       'FIND IF ANY UNIT THAT MUST BE ON SHOULDVE BEEN DISPATCHED 'MORE'   Section 16
5420       HELP(I) = 0
5430            IF (PLUSO(I,J) > 0) AND (P(I,J) < PMX(I)) AND (CAND(I) <> 0)
                  AND (ULBVAL(I,J) < UNLVL)     THEN HELP(I)=1
5440            IF (HELP(I) = 0) THEN GOTD 5470
5450            CAND(I) = 0 : POMO = PDMO + PLUSO(I,J) : P(I,J) = .001
5460 '          IF (HELP(I) = 1) THEN PRINT I,PDMO,P(I,J),HELP(I):STDP
5470       NEXT I
5480 NEXT I
5490 MRK = 0
5500 FOR I = 1 TO UQNT
5510 'DETER IF ANY HELP(I)S ARE = 1 - IF SO WHICH ONES
5520    IF (HELP(I) <> 0) THEN MRK = I
5530    IF (MRK = 1) THEN PRINT I,CAND(I),POMD,P(I,J):STOP
5540 NEXT I
5550 IF (MRK = 0) THEN GOTO 5580
5560 THRU = 5
5570 GOTO 4940
5580 GOSUB 3030
5590 IF (MRK <> 1) THEN GOTO 5680
5600 PVAL = 0
5610 FOR I = 1 TO UQNT
5620     IF (CAND(I) = 0) THEN GOTD 5650
5630     IF (P(I,J) < PLUSO(I,J)) THEN P(I,J) = PLUSO(I,J)
5640     PVAL = PVAL + P(I,J)
5650 NEXT I
5660 POMO = POORIG - PVAL
5670 GOTO 4900
5680 GOSUB 3220
5690 GOTO 3350
5700 GOSUB 3220
5710 PVAL(J) = 0
5720 POMO(J) = POMO
```

Section 17

```
5730 POMD = 0
5740 TRYAGN = 2
5750 CTR = 0
5760 'PRINT "**********",J + 1,"**************"
5770 GOON = 0
5780 J = 3
5790 IF (TRKR = 1) THEN GOTO 5810
5800 NEXT J
5810 FOR J = 1 TO NRS
5820 FOR I = 1 TO UQNT
5830   IF (P(I,J) = 0) THEN GOTO 5870
5840   NTRT(I) = (ALPN(I) + (BET(I)*P(I,J)) + (GAM(I)*P(I,J)*P(I,J)))
5850   NRTCST(I) = NTRT(I)*CST(I)
5860   TOTNCST = TOTHCST + NRTCST(I)
5870 NEXT I
5880  PRINT "************************************************************"
5890  PRINT "*CUMULATIVE RESULTS FROM REQUIREMENTS AND SYS. INC. LAMBDA:*"
5900  PRINT "************************************************************"
5910  PRINT "                                                          *"
5920  PRINT "UNIT        INC CST          POWER          NTRT CST      *"
5930  PRINT " (#)          ($)             (MW)          ($/MBtu)      *"
5940  PRINT "                                                          *"
5950  PRINT "************************************************************"
5960   PRINT "----------------------------------------------------------"
5970   PVAL = 0
5980     FOR I = I TO UQNT
5990      PRINT I,LAMBVAL(J),P(I,J),HRTCST(I)
6000      PVAL = P(I,J) + PVAL
6010     NEXT I
6020   PRINT "----------------------------------------------------------"
6030   PRINT "TOTALS:              ",PVAL,TOTNCST                       "
6040   PRINT "**********************************************************"
6050 IF (POMD > 1) THEN GOTO 6080
6060 IF (POMD > -1) THEN GOTO 6100
6070 GOTO 6190
6080 DIFF = PO(J) - PVAL
6090 IF (DIFF > 0) THEN GOTO 6160
6100 IF (DIFF < 0) THEN GOTO 6120
6110 GOTO 6190
6120 IF (DIFF => WCOMPAR) THEN GOTO 6180
6130 DIFF = -1 * DIFF
6140 PRINT "CONSTRAINTS VIOLATED-SHOULD SELL",DIFF,"MW"
6150 GOTO 6180
6160 IF (DIFF <= COMPAR) THEN GOTO 6180
6170 PRINT "CONSTAINTS VIOLATED-SHOULD BUY",DIFF,"MW"
6180 PRINT "---------------------------------------------------"
6190 J = 3
6200 NEXT J
6210 BEEP:END
6220 '*************************************************************
6230 '----------------- SUBROUTINE TDMCHPWR ----------------------------
6240 'DETERMINE WHICH UNITS TO USE WHEN FIRST CRITERIA (MIN POWER SUMS)
6250 'IS EXCEEDED - SEND BACK TO LAMBDA DISPATCH SUBROUTINE
6260 'I) ORDER MIN. LAMBDAS IN ASCENDING ORDER
6270 K = 1 : DROR(K) = 100 : UNT(K) = 0
6280 MNCAND1 = 100 : TKN = 0
6290 FOR I = 1 TO UQNT
6300    IF (OROR(K) = 100) THEN GOTO 6320
6310    IF (LAMBMN(I) < OROR(K-1)) OR (I = UNT(K-1)) THEN GOTO 6340
```

Subroutine 6

```
6320    IF (LAMBMN(I) > MNCAND1) OR (CAND(I) = 4) THEN GOTO 6340
6330    MNCAND1 = LAMBMN(I) : TKN = I
6340 NEXT I
6350 ORDR(K) = MNCAND1 : UNT(K) = TKN : K = K + 1 : CAND(TKN) = 4
6360 IF (K < (UQNT+1)) THEN GOTO 6280
6370 PRINT "CAND#     LAMBDA VAL     UNIT #"
6380 FOR I = 1 TO K
6390    TKN = UNT(I)
6400    IF (P(TKN,J) = PMX(TKN)) THEN CAND(TKN) = I
6410    IF (I > UQNT) THEN GOTO 6430
6420    PRINT CAND(TKN),ORDR(I),UNT(I)
6430 NEXT I
6440 'PRINT "WOULD YOU LOOK AT THIS, K=",K:STOP
6450 RETURN
6460 '-----------------------------------------------------------------
6470 '*****************************************************************
6480 '*****************************************************************
6490 '------------------- SUBROUTINE NOCAND ---------------------------
```

Subroutine 7

```
6500 'TAKE 1ST TWO UNITS WITH LOWEST LAMBDA VALUE.  SET THE LAMBDA VALUE
6510 'AT THE LOWEST POINT AT WHICH BOTH UNITS CAN OPERATE, SUM THEIR PROD
6520 'LEVELS AND COMPARE TO THE POWER DEMAND.  IF POWER DEMAND IS NOT MET
6530 'BRING UP UNIT WITH NEXT LOWEST LAMBDA VALUE AND REPEAT THE PROCESS.
6540 'IF ANY PRODUCTION MAXIMUMS ARE EXCEEDED DURING THIS PROCESS, SUBTR.
6550 'THAT VALUE FROM THE POWER DEMAND AND MAKE THE RESPECTIVE UNIT A NON
6560 'CANDIDATE.  WHEN POWER DEMAND IS EXCEEDED READJUST THE POWER DEMAND
6570 'VALUE AND DISPATCH BETWEEN ALL THE CANDIDATE UNITS - WHICH ARE ALL
6580 'THE UNITS BROUGHT ON IN THIS PROCESS EXCEPT FOR the last one BROUGHT
6590 'ON AND THOSE WHOSE POWER MAXIMUMS WERE EXCEEDED.
6600 POMODMY = PDMD : MRK = 0 : CRUISE = 0 : LAMBMN = 0 : TAKN = 0
6610 FRSTRN = 0 : EINMAL = 0 : YES = 0
6620 FOR Z = 1 TO UQNT
6630 '-----------------------------------------------------------------
```

Part A

```
6640 'FIND UNIT WITH COMPARITIVE LAMBDA VALUE - UNIT SET AT MIN LAMBDA
6650    TKN = UNT(Z)
6660     IF (UNT(Z) = 0) OR (ORDR(Z) < LAMBMN) OR (CAND(TKN) = 3)
          OR (CAND(TKN) = 1) THEN GOTO 7890
6670    LAMBMN = ORDR(Z) : CAND(TKN) = 3 : P(TKN,J) = PMN(TKN)
6680    IF (FRSTRN = 0) THEN GOTO 7880
6690 PRINT "TKN,P(TKN,J)",TKN,P(TKN,J):STOP
6700 '-----------------------------------------------------------------
6710 POMODMY = PODRIG :PRINT PODRIG:STOP
6720 '-----------------------------------------------------------------
```

Part B

```
6730 'FIND INCREMENTAL POWER DEMAND FOR THIS SET OF CANDIDATES
6740 FOR I = 1 TO UQNT
6750    POMODMY = POMODMY - PLUSO(I,J)
6760    IF (PLUSO(I,J) = 0) OR (P(I,J) <= PLUSO(I,J)) THEN GOTO 6790
6770    POMODMY = POMODMY + PLUSO(I,J) - P(I,J)
6780 '  PRINT "I,POMODMY,PLUSO(I,J),P(I,J)",I,POMODMY,PLUSO(I,J),P(I,J):STOP
6790 NEXT I
6800 PRINT "POMODMY=",POMODMY:STOP
6810 '-----------------------------------------------------------------
6820 '-----------------------------------------------------------------
```

Part C

```
6830 'SET ALL PWR SETTINGS TO A DUMY VARIABLE SO THEY CAN BE RESET IF NEC
6840 FOR I = 1 TO UQNT
6850   USET(I) = P(I,J)
6860 NEXT I
6870 '-----------------------------------------------------------------
6880 '^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Part D

```
6890 'FIND SET OF CANDIDATE UNITS
```

```
6900     FDR I = I TD UQNT + I
6910          IF (CRUISE = 1) THEN GDTD 6940
6920 '        PDNDDNY = PDNDDNY - PNN(TKN)
6930 '        IF (P(TKN,J) <= PLUSD(TKN,J)) THEN PDNDONY = PDMDDNY + PLUSD(TKN,J)

6940          TAKN = UNT(I)
6950          IF (I = Z) THEN I = UQNT
6960 PRINT "UNT,ORDR,LANBMN,I,CANO",UNT(I),DRDR(I),LAMBNN,I,CAND(TAKN):STOP
6970          IF (UNT(I) = O) DR (DRDR(I) > LANBMN) DR (I = UQNT)
               DR (CAND(TAKN) = I) THEN GDTD 7580
6980          IF (PLUSD(TAKN,J) > 0) THEN PDNDDNY = PDNOONY + PLUSO(TAKN,J)
6990 PRINT "PDNDDNY=",PDNDDNY:STDP
7000          P(TAKN,J)=(LAMBMN-(BET(TAKN)*CST(TAKN)))/(2*CST(TAKN)*GAN(TAKN))
7010 PRINT "TAKN,P(TAKN,J)",TAKN,P(TAKN,J):STOP
7020          IF (P(TAKN,J) => PNX(TAKN)) THEN GDTD 7050
7030          IF (P(TAKN,J) < PLUSD(TAKN,J)) THEN P(TAKN,J) = PLUSD(TAKN,J)
7040          PDMDDNY = PDNDDMY - P(TAKN,J) : CANO(TAKN) = 0 : GDTD 7060
7050           P(TAKN,J) = PNX(TAKN) : PDNOONY = PONODNY - PNX(TAKN)
               : CAND(TAKN) = 5
7060          IF (PDMOOMY = -1) DR ((PDNDDNY > -1) AND (PDNDONY < 1))
               DR (PDMDDNY = I)  THEN GDTD 7570
7070          IF (PDNDDNY > 1) THEN GDTD 7580
7080 '...... WHAT TD DD WHEN UNIT NUST BE DN BUT TDD NUCH PDWER ........
7090 REM FIRST: TEST PRESENT CASE TD DETERNINE IF IT'S CAND LISTING        Part E
7100  YES = I : NC = CAND(TAKN) : CAND(TAKN) = 0 : NP = P(TAKN,J) :
      P(TAKN,J) = .00I
7110           IF (UNT(I+I) <> TKN) THEN YES = 0
7120           FOR L = I TD UQNT
7130               IF (LANBNN(L) => LAMBNN) DR (L = TAKN) GDTD 7160
7140               IF (P(L,J) <> PNX(L)) THEN YES = 0
7150               IF (YES = 0) THEN L = UQNT
7160           NEXT L
7170  PRINT YES,UNT(I+I),TKN,UNT(I),TAKN:STOP
7180           IF (YES = I) THEN GDTD 7640
7190           CAND(TAKN) = NC : P(TAKN,J) = NP
7200 REN RESET UNIT PDWER LEVELS
7210           FOR L = I TD UQNT
7220               P(L,J) = USETI(L) : TKN = UNT(Z-1) : CANO(L) = CSETI(L)      Part F
7230 '             IF (P(L,J) = PMX(L)) DR ((PLUSD(L,J) > 0)
                    AND (LANBMN(L) > LAMBMN(TKN))) THEN CAND(L) = I
7240 '             IF (CAND(L) <> I) DR (P(L,J) = 0) THEN GDTD 6750
7250 '             CANO(L) = 0
7260           NEXT L
7270 REN RESET INCRENENTAL PDWER DENAND
7280         PDNDDMY = PDDRIG : YES = 0                                          Part G
7290           FDR L = I TD UQNT
7300               IF (CAND(L) = 0) THEN P(L,J) = 0
7310               PDMDDMY = PDNDDNY - P(L,J)
7320           NEXT L
7330 REN PRINT DUT CANDIDATE VALUES ANO INC PWR DND
7340         PRINT "UNT#      CAND#       PWR SET"                               Part H
7350           FDR L = I TD UQNT
7360               PRINT L,CANO(L),P(L,J)
7370           NEXT L
7380         PRINT "PDDRIG : PDNDDNY",PDDRIG,PDNDDNY:STDP
7390         GDTD 7570
7400 '******************,......................************************
7410           NINLAN = 0 : MRK = 5                                               Part I
7420           FDR K = 1 TD UQNT
```

```
7430           IF (CAND(K) <> 0) THEN GOTO 7450
7440           IF (LAMBMN(K) > MINLAM) THEN MINLAM = LAMBMN(K)
7450         NEXT K
7460         FOR K = 1 TO UGNT
7470           IF (CAND(K) <> 0) OR (LAMBMN(K) <> MINLAM) GOTO 7490
7480           CAND(K) = 4 : PDMDDMY = PDMDDMY + P(K,J)
7490         NEXT K
7500       IF (PDMDDMY < -1) THEN GOTO 7200
7510 '*****************.......................**********************
7520 '.............................................................       Part J
7530       IF (MRK = 5) THEN GOTO 7570
7540 '       IF (P(Y,J) <> PMX(Y)) THEN GOTO 6330
7550 '       CAND(UNT(Y)) = 1 : PDMD = PDMD - PMX(Y)
7560        IF (PDMDDMY < -1) OR (PDMDDMY > 1) THEN GOTO 7580
7570        I = UGNT : Z = UGNT : PDMD = PDMDDMY
7580 'PRINT "PDMDDMY,I",PDMDDMY,I:STOP
7590      CRUISE = 1 : EINMAL = 1
7600    NEXT I
7610 '^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
7620 '-------------------------------------------------------------
7630        IF (Z = UGNT) THEN GOTO 7810                                  Part K
7640 'STORE NEW POWER VALUES IN CASE THIS IS DISPATCHABLE CASE (PRINT)
7650 IF (YES = 0) THEN GOTO 7690
7660 IF (PDMDDMY => PMN(TKN)) THEN GOTO 7690
7670 P(TKN,J) = PMN(TKN) : CAND(TKN) = 1 : P(TAKN,J) = .001 : CAND(TAKN) = 0
7680 PDMDDMY = PDMDDMY - PMN(TKN) + PMX(TAKN)
7690 PRINT "UNT#        PWR SET         CAND VAL"
7700 FOR L = 1 TO UGNT
7710     IF (CAND(L) = 3) THEN CAND(L) = 0
7720     USETI(L) = P(L,J) : CSETI(L) = CAND(L)
7730   PRINT L,P(L,J),CAND(L)
7740 NEXT L
7750 PRINT "Z : TKN",Z,TKN:STOP
7760 IF (YES = 1) THEN GOTO 7200
7770 '-------------------------------------------------------------
7780    IF (PDMDDMY > 1) THEN PDMDDMY = PDMD
7790 '  IF (PDMDDMY < -1) OR (PDMDDMY > 1) THEN GOTO 6360
7800 '  Z = UGNT
7810  IF (Z = UGNT) OR (FRSTRN = 0) THEN GOTO 7890
7820 '-------------------------------------------------------------
7830 'RESET THE POWER SETTINGS TO THEIR ORIGINAL SETTINGS                  Part L
7840 FOR I = 1 TO UGNT
7850    P(I,J) = USET(I)
7860 NEXT I
7870 '-------------------------------------------------------------
7880  FRSTRN = 1 : CRUISE = 0 : EINMAL = 0                                Part M
7890 NEXT Z
7900 STOP
7910 PRINT "UNT#          CAND#          PDN SET"
7920 FOR I = 1 TO UGNT
7930    IF (CAND(I) > 0) THEN CAND(I) = 1
7940    IF (CAND(I) = 1) AND (P(I,J) = .001) THEN P(I,J) = 0
7950    IF (CAND(I) = 0) THEN P(I,J) = .001
7960    PRINT I,CAND(I),P(I,J)
7970 NEXT I
7980 PRINT "I FINISHED THIS PUPPY OF A SUBROUTINE:PDMD=",PDMD:STOP
7990 RETURN
8000 -------------------------------------------------------------
8010 ***************************************************************
```

Appendix 3: Variable Definition Supplement

PMN(I) = minimum power setting for unit J

UNUSD(I,J) = unit i used in hour J case

PLUSD(I,J) = power level unit i (used) is set at in hour J.

PVAL(J) = sum of power produced over all units

ULBVAL(I,J) = setting of λ for unit i used in hour J

GPRD = sum of gamma-cost function

DEN = denominator of system incremental lambda

GTRM = result of multiplying GPRD and GAM(I)

STRM = prod of GTRM and BET(I)

TRYAGN = locator variable

PDSAVE = any variable for PDMD + PLUSD(I,J) for when must run units are candidate: must reset PDMD

TRK = tracking variable: MNMN will be sum of must run units

PTOTMX = total of all unit max

MNMN = the minimum of all unit power minima or sum of all must run units at their must-run level

PD(J) = total power demanded for hour J

PDORIG = total power demanded for hour J

MRK = indicates where process has been (not first time through)

PDMD = power demanded (generally increment)

PDMDDMY = dummy variable for PDMD

EVAL = evaluation term: difference between power production and power demanded

XX = another marker - check to make sure no candidate limits exist: 0 = NO 0 ≠ YES

X = variable marker - candidates exist (=2), candidates don't exist (≠2)

ULBVAL = value of lambda determined for unit from dispatch

MOCMN = used to find minimum LAMBMN (i) and save the value

MCDMY = dummy variable for MOCMN so value can change yet be recalled

CAND(I): dispatchable or not
  0 = dispatchable
  $\neq$ 0, cannot dispatch
  1 non dispatchable
  2,3 possible to dispatch

MNTOT = check to see if total of candidate units minimum power settings equal or exceed PDMD

NOCAND = marker - if/when no candidates exist

TRKR = variable marker - only 1 candidate unit

MXMN = the maximum value of the minimum power settings

MRK = indicates whether unit with P(I,J) < PMN(I) has been selected

THRU = variable marker of where process has been

UNLVL = stores UNLVL(I) value

UNLVL(I) = independent unit variable

HELP(I) = independent unit variable

USET(I) = dummy unit power setting so P(I,J) can change yet be recalled later

Z = incrementing variable

TKN = marks variable which is supposed to have lambda value to be compared against

TAKN = variance compared to TKN

ORDR(K) = orders minimum lambda in ascending order

MNCAND1: stores ORDR(K) value

UNT(K) = unit with minimum lambda

TAKN = the TAKN unit, stores UNT(K) value

Cruise = marker of what process has been completed in NOCAND subroutine

FRSTRN = marks when algorithm is in first run of NOCAND subroutine

EINMAL = variable marker

YES = answer variable

MC = dummy variable for CAND (TAKN)

MP = dummy variable for P(TAKN,J)

L = incrementing variable

UNIT COMMITMENT USING CONSTRAINED LAMBDA DISPATCH WITH THE IBM:PC

by

Bradley Dean Eckhoff

B.S., Kansas State University, 1983

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the requirements

for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1985

ABSTRACT

The idea of unit commitment and the desire to use equal incremental cost (lambda dispatch) to generate optimal unit settings has been around for several decades. This thesis dedicates the first three chapters to a literature research of this field and a comprehensive summary of the works that have been done to date. The combination of techniques that give the most comprehensive background of related works is the traditional lambda dispatch, the branch-and-bound method, dynamic programming, dynamic programming with linear programming, cartesian coordinate formulation, load flow analysis with transmission losses considered, and an economic dispatch program developed by the Electric Power Research Institute (EPRI).

The purpose of this research was two-fold: One, to develop the CLD program code to work on an IBM:PC and two, to obtain results that are better or equivalent to those obtained by the EPRI program.

CLD was selected over other techniques because of the simplified nature of the dispatch problem created by KPL assumptions used in their dispatch scenario. All 14 units make up one bus instead of 14 buses, units are never shut down entirely, and transmission losses being accounted for by producing eight percent more energy than is demanded are a few of the assumptions made.

The process of developing this code was to take real data and real situations as well as contrived situations that are logical extensions of real problems and solving for each of the different situations (cases). A few of the different cases that were considered were: (1) must sell, where one has to produce a minimal amount of power yet system demand is below this level, (2) must buy, where even with all units on

at maximum capacity the system demand is not met, (3) CLD simple case, where dispatching was necessary but only between the three Jeffrey units, only one algorithmic cycle was necessary, (4) CLD between two Jeffrey units with the other unit on at maximum capacity as well as other must run units producing at must-run levels, and (5) CLD between Tecumseh and Lawrence units with other must run units producing at must-run levels and all Jeffrey units producing at a maximum capacity level.

The results obtained by this CLD process were very encouraging. As shown in the results analysis in Chapter 5, a financial savings of as much as 14 million dollars annually could be recognized when using the CLD process instead of the pick-and-choose method used by KPL presently or the EPRI program being tested by KPL for future use. This savings was recognized by simply dispatching over the least expensive units (in terms of fuel cost) as long as is possible and practical.

Using the IBM:PC is practical in the logical sense that the program can be run whenever desired at a cost much lower than programs run on a time-sharing process with Boeing as is done with the EPRI program. However, it is the recognition of the financial savings that indicate that using a CLD process over present techniques would be most beneficial.