電力系統における集中制御システムの最適信頼度設計手法

正員 松坂知行 (八戸工大)

非会員 石川清二郎 (東北電力)

非会員 皆沢和男 (東北電力)

1. ま え が き

電力系統において集中制御システムが広範囲に導入されてきている。一方、集中制御システムでは全系統の情報が集中制御所に集められているので、情報伝送システム、情報処理システムの信頼度設計は重要な問題である。

筆者は先に集中制御システムの信頼性の評価尺度として情報損失量の期待値⁽¹⁾、および電力系統の事故と制御システムとのそう遇する時間の期待値 (MTSF)⁽²⁾を提案した。本稿はこれらの評価尺度を用いて集中制御システムの最適信頼度設計を行なう手法について述べる。

与えられた制約条件のもとでシステムの信頼度指標を最大にするための問題は最適信頼度配分問題と呼ばれ、信頼性工学の一分野を成している。これらの方法には文献(1)などに述べられているように種々の方法が開発されているが、一般に信頼性工学では簡単な直並列システムを対象として、この信頼度関数、アベイラビリティ、MTBFなどを最大化する問題を扱っていることが多い⁽³⁾⁽⁴⁾。しかしこのような最適信頼度配分手法を現実のシステム設計に生かすためには、与えられたシステムに適した信頼度指標のもとに、それぞれのシステムに固有な条件を考慮した信頼度設計手法の開発が必要である。

本研究では、

(1) 費用を制約条件として情報損失量の期待値を最小にする冗長設計法

(2) MTSF を制約条件として費用を最小にする冗長設計法

を0-1計画問題⁽⁵⁾として定式化し、具体的な集中制御システムについて計算例を示した。

2. 信 頼 度 指 標

電力系統における集中制御システムで扱う情報には、系統の事故表示情報、遠隔操作情報、事故時の自動復旧操作情報などがある。

このうち事故表示情報は電力系統に事故が発生した場合に生ずる情報であり、電力系統の状態変化を伝えるので、系統運用上、上記三つの情報の中で最も重要と考えられる。ゆえに集中制御システムの、この情報に対する信頼度を高めることが信頼度設計上重要と思われる。そこで本稿では事故表示情報に対する信頼度設計法を述べることにする。

事故表示情報は設備事故、送電線事故などにより発生するが、このうち送電線事故のひん度が圧倒的に多いので送電線の事故表示情報と考えてよい。この事故は制御システムの平均修理時間よりもはるかに長い間隔で発生するため、制御システムの故障がそのまま情報損失に結びつかないので、後述するように MTSF を用いて情報損失量を求めなければならない。

以下、MTSF、情報損失量の期待値について述べる。

〈2・1〉 MTSF (Mean Time Between System Failures) 電力系統の事故と制御システムの故障との同時発生する時間間隔の期待値を MTSF という。いま情報発生点となるサブシステム i と情報受信点となるサブシステム j 間の MTSF を M_{ij} とすると

$$M_{ij} = \frac{1}{\alpha_{ij} \cdot \bar{A}_{ij}} \quad \dots \dots \dots (1)$$

で示される⁽²⁾。

ここに α_{ij} = サブシステム i, j 間に伝送される電力系統の事故発生間隔が指数分布に従

Optimum Redundancy Assignment of Supervisory Control Systems in Power System. By Tomoyuki Matsuzaka, Member (Hachinohe Institute of Technology) Seigiro Ishikawa, Non-member & Kazuo Minazawa, Non-member (Engineering Department of Tohoku Electric Power Company).

松坂知行: 正員, 八戸工業大学電気工学科

石川清二郎: 非会員, 東北電力(株)技術部

皆沢和男: 非会員, 東北電力(株)技術部

って発生する場合の指数分布のパラメータ,
 $\overline{A_{ij}}$ = サブシステム i, j 間の情報ルートの
 アンアベイラビリティ

〈2・2〉 情報損失量の期待値 情報発生点となる
 サブシステム i と情報受信点となるサブシステム j 間
 に伝送される情報量を $q_{ij}(\text{bit})$ とすると, i, j 間の
 情報損失量の期待値 $Q_{ij}(\text{bit/s})$ は

$$Q_{ij} = q_{ij} / m_{ij} = q_{ij} \alpha_{ij} \overline{A_{ij}}$$

で示すことができる。

よってシステム全体の情報損失量の期待値 $\overline{Q}$ は

$$\overline{Q} = \sum_{(i,j)} q_{ij} \cdot \alpha_{ij} \overline{A_{ij}} \dots\dots\dots (2)$$

となる。ここに

$\sum_{(i,j)}$ = 情報発生点 i と情報受信点 j により構
 成される情報ルートの集合についての和

3. 最適信頼度設計問題の定式化

本章では第2章で述べた事故表示情報に対する情報
 損失量の期待値および MTSF を信頼度指標として用
 い, 費用を制約条件として情報損失量の期待値を最小
 にするシステムの信頼度設計法, 各情報ルートの MT-
 SF を制約条件としシステムの構成費用を最小にする
 ための信頼度設計法を0-1計画法を用いて定式化す
 る手法について述べる。

〈3・1〉 費用制限のもとにシステムの情報損失量を最
 小にする冗長設計法 集中制御システムでは集中制
 御所と各子局との間で情報を授受するので, 情報を授
 受するサブシステムの添字 i, j を一つの添字を用
 いて情報ルートを表わすことができる。いま集中制
 御所と各子局 j との間の情報損失量の期待値を Q_j ,
 情報量を q_j , アンアベイラビリティを $\overline{A_j}$, 情報の平
 均発生ひん度を α_j とすると

$$Q_j = q_j \alpha_j \overline{A_j} = q_j \alpha_j (1 - A_j) \dots\dots\dots (3)$$

となる。ここに A_j はアベイラビリティである。

よって(3)式の Q_j を小さくするためには $q_j \alpha_j A_j$
 を大きくすればよい。いま

$$P_j = q_j \alpha_j A_j = q_j \alpha_j \left(\prod_{i \in S_j} A_i \right) \dots\dots\dots (4)$$

$i \in S_j$ = 情報ルート j を構成するサブシステ
 ム i の集合

とすると P_j は集中制御所と子局 j 間との情報到達
 量を示す⁽⁶⁾。 P_j を大きくするためには A_i を大きく
 すればよい。 A_i は情報ルートの等価故障率を λ_i , 等
 価修理率を u_i とすると

$$A_i = \frac{u_i}{u_i + \lambda_i} \dots\dots\dots (5)$$

で示されるから A_i を大きくするには λ_i を小さく,
 u_i を大きくすればよいが, 一般に λ_i はハードウェア
 の構成を変えないと根本的に改善することはできな
 いし, また, ある特定の値になるようにハードウェアの設
 計を行なうことは困難である。 u_i を大きくするには
 保守態勢がからみ, 簡単にはある特定の値に維持する
 ことは困難である。従って情報ルートのアベイラビ
 ティを改善するには, 情報ルートを構成する個々のサ
 ブシステムを冗長化するのが实际的である。そこで以
 下冗長化により情報到達量の期待値を最大化する定式
 化について述べる。

いま, (4)式のアベイラビリティ A_i を自然対数で
 表わしたときの関数を $L_n P_j$ と定義する。

$$\left. \begin{aligned} L_n P_j &= q_j \alpha_j L_n A_j \\ &= q_j \alpha_j \sum_{i \in S_j} L_n A_i \end{aligned} \right\} \dots\dots\dots (6)$$

$\sum_{i \in S_j}$ = 情報ルート j を構成するサブシステム

i の集合についての和

とすると(4)式の P_j と(6)式の $L_n P_j$ は A_i
 に関して単調増加関数であり1対1に対応している。
 よって P_j を大きくするかわりに $L_n P_j$ を大きくす
 れば情報損失量の期待値を小さくできる。システム全
 体としての $L_n P_j$ の合計を求めると

$$\left. \begin{aligned} \sum_j L_n P_j &= \sum_j \left(\sum_{i \in S_j} L_n A_i \right) q_j \alpha_j \\ &= \sum_{i=1}^N \left(\sum_{j \in S_i} q_j \alpha_j \right) L_n A_i \end{aligned} \right\} \dots\dots\dots (7)$$

ここで

$\sum_{j \in S_i}$ = サブシステム i を通る情報ルート j の

集合についての和

である。 N はサブシステムの数である。

いま変数 x_{ik} を

$$x_{ik} = \begin{cases} 1: \text{サブシステム } i \text{ の並列数を } k \text{ とする} \\ \text{とき} \\ 0: \text{そうでないとき} \end{cases}$$

とすると情報到達量を最大にするための目的関数
 $f(\mathbf{x})$ は

$$f(\mathbf{x}) = \sum_{i=1}^N \sum_{k=1}^{L_i} a_{ik} \cdot x_{ik} \dots\dots\dots (8)$$

但し

$$\left. \begin{aligned} \mathbf{x} &= (x_{1k1}, x_{2k2}, x_{3k3}, \dots, x_{NkN}) \\ a_{ik} &= \left(\sum_{j \in S_i} q_j \alpha_j \right) L_n [1 - (1 - A_i)^k] \\ L_i &= \text{サブシステム } i \text{ の } k \text{ の上限} \end{aligned} \right\} \dots\dots (9)$$

次に制約条件を考える。一般には費用, 重量, 体積
 などの制約条件が考えられるが電力システムの集中制御シ
 ステムでは費用条件が最も重要であるので制約条件と

しては費用のみを考える。

いま,

R_i = サブシステム i の修理費用

t_i = サブシステムの使用期間

C_{i0} = サブシステム i の価格

C_S = システムの費用制限

とすると制約条件は

$$\sum_{i=1}^N \sum_{k=1}^{L_i} c_{ik} \cdot x_{ik} \leq C_S \quad \dots\dots\dots (10)$$

ここに,

$$c_{ik} = k(c_{i0} + R_i \lambda_i t_i) \quad \dots\dots\dots (11)$$

またサブシステム i は少なくとも 1 個は必ず使用されるので

$$\sum_{k=1}^{L_i} x_{ik} = 1 \quad (i=1, 2, \dots, N) \quad \dots\dots\dots (12)$$

の制約条件が必要である。従って (10) 式, (12) 式の $N+1$ 個の制約条件のもとに (8) 式を目的関数として最大化する 0-1 計画問題となる。(8) 式, (10) 式 (12) 式は一般的な多重化を行なう場合の 0-1 計画問題への定式化であるが, サブシステム数 N が 100 の場合には仮に並列数の上限 L_i を 2 としても, 制約条件数は 101, 変数の数は 200 になり 0-1 計画問題としては問題のサイズが大きすぎ解法が非常に困難となる。並列数の上限は電力システムの制御システムでは一般に 2 であるので, これを考慮した場合には以下のように問題のサイズを小さくできる。

(8) 式を変形し

$$f(\mathbf{x}) = \sum_{i=1}^N (a_{i1} x_{i1} + a_{i2} x_{i2})$$

$x_{i1} + x_{i2} = 1$ の関係を用いて

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^N [a_{i1}(1 - x_{i2}) + a_{i2} x_{i2}] \\ &= \sum_{i=1}^N a_i x_i + \sum_{i=1}^N a_{i1} \end{aligned}$$

ここで

$$a_i = a_{i2} - a_{i1} = \left(\sum_{j \in S_i} q_j \alpha_j \right) l_n \frac{1 - (1 - A_i)^2}{A_i} \quad \dots\dots\dots (13)$$

$$x_i = x_{i2}$$

とする。

$\sum_{i=1}^N a_{i1}$ は一定であるので新しい目的関数 $f_1(\mathbf{x})$ は

$$f_1(\mathbf{x}) = \sum_{i=1}^N a_i \cdot x_i \quad \dots\dots\dots (14)$$

制約条件 (10) 式についても同様にして

$$\sum_{i=1}^N (c_{i1} x_{i1} + c_{i2} x_{i2})$$

$$= \sum_{i=1}^N c_i \cdot x_i + \sum_{i=1}^N c_{i1} \leq C_S$$

ここで

$$\left. \begin{aligned} c_i &= c_{i2} - c_{i1} \\ &= c_{i0} + R_i \lambda_i t_i \end{aligned} \right\} \quad \dots\dots\dots (15)$$

である。また,

$$c_S' = \sum_{i=1}^N c_{i1}, \quad c_S'' = c_S - c_S' \quad \dots\dots\dots (16)$$

とおくと新しい制約条件は下式のようになる。

$$\sum_{i=1}^N c_i \cdot x_i \leq c_S'' \quad \dots\dots\dots (17)$$

(14) 式を目的関数, (17) 式を制約条件とする新しい 0-1 計画問題は制約条件式が 1, 変数の数が N となり問題のサイズは非常に小さくなる。

〈3.2〉 制御所と各子局 j との MTSF に制約条件を与えてシステムの費用を最小にする方法
いま情報ルートごとの MTSF の制約条件を T_j とすると情報ルート j の MTSF は $1/\bar{A}_j \cdot \alpha_j$ で示されるから

$$\frac{1}{\bar{A}_j \cdot \alpha_j} \geq T_j \quad (j=1, 2, \dots, M) \quad \dots\dots\dots (18)$$

ここに M は情報ルートを示す。また T_j は下記の

$$\frac{1}{\alpha_j \{1 - \prod_{i \in S_j} [1 - (1 - A_i) L_i]\}} \geq T_j \quad (j=1, 2, \dots, M)$$

制限条件を満たすものとする。

(18) 式より

$$\sum_{i \in S_j} l_n A_i \geq l_n \left(1 - \frac{1}{\alpha_j T_j} \right) \quad (j=1, 2, \dots, M) \quad (19)$$

よって冗長化によってこの関係を得るには

$$\sum_{i \in S_j} \sum_{k=1}^{L_i} b_{ik} \cdot x_{ik} \geq l_n \left(1 - \frac{1}{\alpha_j T_j} \right) \quad \dots\dots\dots (20)$$

($j=1, 2, \dots, M$)

ここに

$$b_{ik} = l_n [1 - (1 - A_i)^k]$$

$L_i=2$ の場合には, (13) 式を導いたのと同様にし

$$\begin{aligned} &\sum_{i \in S_j} (b_{i1} \cdot x_{i1} + b_{i2} \cdot x_{i2}) \\ &= \sum_{i \in S_j} b_i \cdot x_i + \sum_{i \in S_j} b_{i1} \geq l_n \left(1 - \frac{1}{\alpha_j T_j} \right) \\ \therefore \quad &\sum_{i \in S_j} b_i \cdot x_i \geq B_j \quad (j=1, 2, \dots, M) \quad \dots\dots\dots (21) \end{aligned}$$

ここに

$$\left. \begin{aligned} b_i &= l_n \frac{1 - (1 - A_i)^2}{A_i} \\ B_j &= l_n \left(1 - \frac{1}{\alpha_j T_j} \right) - \sum_{i \in S_j} b_{i1} \end{aligned} \right\} \quad \dots\dots\dots (22)$$

また、目的関数は (17) 式を参照して

$$f_2(\mathbf{x}) = \sum_{i=1}^N c_i \cdot x_i \dots\dots\dots (23)$$

よって M 個の制約条件 (21) 式のもとに目的関数 (23) 式を最小にする 0-1 計画問題として定式化することができた。

4. 解法アルゴリズム

本章では 0-1 計画問題の解法について説明する。0-1 計画問題を標準形で書くと連立不等式条件

$$\sum_{j=1}^N \bar{a}_{ij} x_j \leq \bar{b}_i \quad (i=1, 2, \dots, M) \dots\dots\dots (24)$$

と 0-1 条件

$$x_j = 0 \text{ また } 1 \quad (j=1, 2, \dots, N) \dots\dots\dots (25)$$

のもとに下記の目的関数

$$f = \sum_{j=1}^N \bar{c}_j x_j \dots\dots\dots (26)$$

を最大にする $x_j (j=1, 2, \dots, N)$ を求めよという問題となる。

f を最小化する場合には新しい変数 $x'_j = 1 - x_j$ ($j=1, 2, \dots, N$) を導入すれば容易に問題を書き改めることができる。条件 (25) 式を満たす $(x_1, x_2, \dots, x_N)$ の数は全部で 2^N 個ある。条件 (24) 式を満たし、 f を最大にする最適解はこの 2^N 個の中に含まれている。ゆえに最適解を求めるには 2^N 個のすべての組合せについて、条件 (24) 式を満たすか否かチェックし、条件 (24) 式を満たすものについては (26) 式を計算すればよい。しかしこの方法では N が大きくなるにつれ、計算時間が指数的に増大してしまう。そこで計算時間を可能なかぎり少なくして、最適解を求めるための方法が従来からいろいろ研究されている。以下本稿で使用した二つのアルゴリズムについて説明する。

〈4・1〉 アルゴリズム 1⁽⁸⁾ この計算原理は分岐限定法の計算原理に基づくものでその基本的思想は、条件 (25) 式が満たされているか否かのチェックを、 2^N 個の組合せすべてについて行なうのではなく、一部の組合せについてのみチェックを行ない、しかも条件 (25) 式を満たすもの数え落しがないようにすることによって計算の効率化を図ろうとするとところにある。本手法は Balas によって提案された。

まず以下の説明に用いる用語について述べる。

部分解 S : 各変数 x_j が 0 または 1 に固定された $\mathbf{x}' = (x'_1, x'_2, \dots, x'_N)$ の部分集合のことである。

自由変数: 部分解 S の中に入らない変数を自由変数という。

完備化: 自由変数に 0 または 1 の値が与えられるこ

とを部分解 S が完備化されるという。

次に Balas の計算法を述べる。

(1) すべての x'_j ($j=1, 2, \dots, N$) が 0 である解が実行可能ならば、これが求める解である。さもなければすべての x'_j を 1 として評価された f の値を f とする。 S を空集合とする。(2) へいく。

(2) 自由変数を 0 とおいて部分解 S が完備化される時、満たされない集合 V を求める。 V が空集合なら (7) へ、そうでなければ (3) へいく。

(3) 自由変数を 0 とおいて部分解 S が完備化される時の f の値を f_P とする。(4) へいく。

(4) 次の条件 (a), (b) を満たし、かつ S の中に入らないすべての変数を集合 T に記憶する。(a) 目的関数の中で $f - f_P$ よりも小さい係数をもつ。(b) V の中のいずれかの制約において正の係数をもつ。 T が空集合なら (9) へ、そうでなければ (5) へいく。

(5) V の中の各制約が T の中にある変数のみを S に追加して満たされるなら (6) へ、そうでなければ、(9) へいく。

(6) T の中において、 $\bar{a}_{ij} > 0$ なる係数和 $\sum_{i=1}^M \bar{a}_{ij}$ の最大値を与える j を求め、この変数を S に加えて (2) へいく。

(7) 自由変数を 0 とおいて S を完備化する。この完備化された解 (完備解) が現在の最良解 $\bar{\mathbf{x}}'$ である。 $\bar{\mathbf{x}}'$ に対する目的関数の値を計算し $\bar{f}$ とする。(8) へいく。

(8) S の中の最も右側にある正の要素を求め、それを補元 (負の要素) で置きかえて、それより右側にある全要素を削除する。(2) へいく。

(9) S の中の全要素が負なら (10) へ、そうでなければ (8) へいく。

(10) それまでに最良解を得ていればそれが求める最適解である。計算は終了する。

Balas のアルゴリズムは後に Glover の多段階双対法⁽⁹⁾ Geoffrion の陰伏の計算法⁽¹⁰⁾により多くの工夫がなされ、解の収束性が高められた。本稿で用いた手法も Balas のアルゴリズムと Glover, Geoffrion の手法を組合せたもので文献 (7) の計算機プログラムによった。

〈4・2〉 アルゴリズム 2 Balas の手法は変数の数が増えると収束に時間がかかり実用的でなくなる。また現実の問題を解く場合、目的関数や制約条件式の係数が推定値や近似値である場合が多く、このような場合には厳密な最適解ではなくても最適解に近い解で

満足できることが多い。計算時間が短かく準最適解が得られる手法として、文献 (11) のアルゴリズムが有効と思われる。このアルゴリズムを適用するには (24) 式、(26) 式の係数には次の制約が必要である。

$$\bar{a}_{ij} \geq 0, \bar{b}_i > 0, c_j > 0 \quad \dots\dots\dots (27)$$

また (24) 式を以後の説明のため次のように変形しておく。

$$\sum_{j=1}^N \bar{a}_{ij}' x_{ij} \leq 1 \quad \dots\dots\dots (28)$$

ここに

$$\bar{a}_{ij}' = \bar{a}_{ij} / \bar{b}_i \quad \dots\dots\dots (29)$$

このアルゴリズムを述べるにあたり、記号を説明をする。

T : 変数の添字集合、すなわち $\{1, 2, \dots, N\}$,
 T_u : x_j の値が 1 に固定された変数の添字集合,
 T_d : T_u に属さない変数の添字集合、すなわち $T - T_u$, P_j : $x_j = 1$ をとったときに必要な制約係数ベクトルで $(\bar{a}_{1j}', \bar{a}_{2j}', \dots, \bar{a}_{Mj}')$ を意味する。 D_i : T_u に含まれる添字を持つ j に関する制約係数の合計、すなわち $\sum_{j \in T_u} \bar{a}_{ij}'$, P_i : D_i を要素とするベクトル、すなわち $(D_1, D_2, \dots, D_M)$, T_c : 解候補となる変数の添字集合、すなわち、

$T_c = \{j | j \in T_d, P_j \leq B - P_u\}$, ここに B は (28) 式の右辺に対応する規格化された M 次元の行ベクトル $(1, 1, 1, \dots, 1)$ である。

以下アルゴリズムを説明する。

(1) T_u を空集合とし、 P_u を零ベクトルとする。また f の値を零、 $x_j = 0$ ($j = 1, 2, \dots, N$) とする。

(2) $\{j | j \in T_d, P_j \leq B - P_u\}$ なる j をすべて T_c に入れる。

(3) T_c が空集合なら計算は終了し、 T_u に含まれる添字を持つ変数が 1 で他は 0 であり、これが解である。もし空集合でなければ (4) へいく。

(4) T_c に属する解候補に対する有効こう配 G_j (effective gradient) を以下のように計算する。

(a) P_u が零ベクトルなら

$$G_j = c_j M^{1/2} / \sum_{i=1}^M \bar{a}_{ij}' (j \in T_c) \quad \dots\dots\dots (30)$$

この状態は繰返し計算の最初のと時のみ起こる。

(b) P_u が零ベクトルでなければ以下を計算する。

$$|P_u| = \left(\sum_{i=1}^M D_i^2 \right)^{1/2} \quad \dots\dots\dots (31)$$

$$E = P_u / |P_u| \quad \dots\dots\dots (32)$$

$$V_j = P_j \cdot E = \left(\sum_{i=1}^M \bar{a}_{ij}' D_i \right) / \left(\sum_{i=1}^M D_i^2 \right)^{1/2} \quad (33)$$

$$G_j = \bar{c}_j / V_j$$

$$= \bar{c}_j \left(\sum_{i=1}^M D_i^2 \right)^{1/2} / \left(\sum_{i=1}^M \bar{a}_{ij}' D_i \right) (j \in T_c) \quad (34)$$

(5) G_j の最大となる j をみつけ、これを k とする。

(6) T_u に k を加える。 P_u に P_k を入れる。 f に $\bar{c}_k$ を加える。 T_d より k を除く、 $x_k = 1$ としてステップ 2 へいく。

このアルゴリズムは制約条件式の数が一つの場合について考えると判りやすい。すなわち (30) 式 (34) 式で $M=1$ とおき、 $\bar{c}_j / \bar{a}_{ij}'$ の最も大きい j をみつけ、この値を k とし、 k に相当する制約係数 $\bar{a}_{ik}'$ を (28) 式の右辺より差引き、 $1 - \bar{a}_{ik}'$ を新たな制約として上述の手順を繰返し、(28) 式の関係が満たされなくなるまでこの手続を続ける。この手法を制約条件式の数が複数個の場合に拡張したのが本節で述べたアルゴリズムである。

上記二つのアルゴリズムの計算時間の比較は第 5 章の計算例の中で述べる。

5. 計算例

次に第 4 章で述べた最適化手法を第 1 図のシステムに適用してみる。

本システムを構成するサブシステムの MTBF, MT-TR, C_{i0} , R_i に関するデータを第 1 表に示す。

また、情報ルート、事故率 α_j に関するデータを第 2 表に示す。第 1 表、第 2 表のデータをもとにして〈3・1〉、〈3・2〉節で定式化された 0-1 計画問題を解いてみる。

〈5・1〉 計算例 1 本節では〈3・1〉節で定式化された費用制限のもとに情報損失量の期待値を最小化する冗長設計法の計算例を示す。

まず、冗長化するサブシステムのケーブルについて述べる。ケーブルの費用はケーブルの価格と工事費より成るが、このうち後者の値は工事方法により異なり、第 1 表の値は架空ケーブルの場合である。またすでに二重化されている部分のサブシステムは対象から除いて考える。次に事故表示情報量 q_j について考える。第 1 図のシステムは 66 kV 送電線の監視制御を対象としており、送電線に一回事故が発生するとしゃ断器の開閉表示が 1 ビット、地絡継電器の表示が 1 ビット、線路名表示 1 ビット、再閉路の実施表示 1 ビット、再閉路不成功表示 1 ビット、合計 5 ビットが伝送される。最後の再閉路不成功表示は再閉路成功時に

第 1 図 集中制御システム

Fig. 1. Centralized supervisory control system.

は伝送されず、このときは合計表示ビットは 4 ビットになるが最大ビット数を取り 5 ビットとした。

次に (13) 式, (15) 式を用い, 第 1 図, 第 1 表, 第 2 表のデータをもとに a_i , c_i を計算した。(15) 式の $t_i=43,800(\text{h})$ とし, (17) 式の $cs''=1,500$ として計算した結果, 二重化すべきサブシステムはアルゴリズム 1, 2 いずれの場合も $i=1$ (電源装置), $i=2$ (計算機本体), $i=7$ (親局装置), $i=10$ (親局装置), $i=18$ (親局装置), $i=22$ (通信線搬送装置), $i=23$ (通信線搬送装置), $i=29$ (通信線搬送装置), $i=30$ (通信線搬送装置), $i=31$ (通信線搬送装置) となった。これらのサブシステムはいずれも情報が集中しており解の結果は納得できる。

次に計算時間について比較すると, 使用した計算機 IBM 370/158 の場合, アルゴリズム 1 では翻訳で 8.38 秒, 実行で 3.17 秒, アルゴリズム 2 ではそれぞれ 2.55

秒, 0.25 秒で, アルゴリズム 2 の方がずっと早い。特にアルゴリズム 1 では (17) 式の制約 cs'' の数値い

第 2 表 情報ルートと送電線の事故率

Table 2. Information routes and occurrence rates of fault signals.

ルート番号	情報ルート	α (事故発生回数/年)
1	58-39-4-3-2-1	2.40
2	59-40-5-3-2-1	2.30
3	60-45- $\begin{bmatrix} -25-42-24-41-20 \\ -28-44-27-26-43-21 \end{bmatrix}$ -6-3-2-1	1.40
4	61-47-30-46-29- $\begin{bmatrix} -25-42-24-41-20 \\ -28-44-27-26-43-21 \end{bmatrix}$ -7-3-2-1	5.80
5	62-30-46-29- $\begin{bmatrix} -25-42-24-41-20 \\ -28-44-27-26-43-21 \end{bmatrix}$ -8-3-2-1	3.10
6	63- $\begin{bmatrix} -25-42-24-41-20 \\ -28-44-27-26-43-21 \end{bmatrix}$ -22-9-3-2-1	3.70
7	64-31- $\begin{bmatrix} -48 \\ -51 \end{bmatrix}$ -22-10-3-2-1	3.80
8	66-33-49-32-31- $\begin{bmatrix} -48 \\ -51 \end{bmatrix}$ -22-11-3-2-1	2.60
9	65-35-50-34-33-49-32-31- $\begin{bmatrix} -48 \\ -51 \end{bmatrix}$ -22-12-3-2-1	2.60
10	67-36- $\begin{bmatrix} -48 \\ -51 \end{bmatrix}$ -23-13-3-2-1	1.60
11	68-38-52-37-36- $\begin{bmatrix} -48 \\ -51 \end{bmatrix}$ -23-14-3-2-1	2.70
12	69-53-36- $\begin{bmatrix} -48 \\ -51 \end{bmatrix}$ -23-15-3-2-1	0.80
13	70-54-16-3-2-1	2.40
14	71-55-17-3-2-1	0.30
15	72-56-18-3-2-1	5.60
16	73-57-19-3-2-1	0.30

第 1 表 サブシステムの信頼度指標と費用

Table 1. Reliability indices and costs of subsystems.

サブシステム	MTBF(h)	MTTR(h)	R_i	c_{io}
親局装置	40,000	4	2	20
子局装置	40,000	4 (6)	3	100
通信線搬送装置	40,000	4 (6)	2	50
マイクロ端局装置	25,000	4 (6)	2	50
計算機本体	800	4	5	1,000
プロセス入出力装置	16,000	4	2	500
電源装置	5,000	4	5	100
通信ケーブル	40,000/10km	8	2	450/10km

かんによっては計算時間が5分を過ぎても収束しないことがあり、この例のように変数の数（この場合 60）が多くなると実用性に乏しいことを示している。しかし〈5・2〉節で述べるように制約条件式の数が増えたと、実行時間についてはアルゴリズム 1の方が早くなってくることが判った。この理由は、アルゴリズム 1で本計算機プログラムには、与えられた制約条件式の線形結合によって更に強い制約条件（いわゆる代理制約条件⁽⁹⁾）を作り出し、探索範囲を狭める手法が組み込まれており、複数制約条件式の場合、これが有効に働くためであると考えられる。

また、このときの所要費用は 1,476.51 となった。

第 3 表に本計算例の冗長設計を行なった後のシステムの信頼度尺度を示す。一方、信頼度を改善する前の信頼度尺度を第 4 表に示す。第 3 表と第 4 表を各情報ルートごとと比較すると MTSF, MTBF とともにほぼ 1 けた大きくなり、アベイラビリティも約 0.5% 改善されているのがわかる。システム全体では MTSF が 1 けた大きくなり、MTBF は約 2 倍になり、アベイラビリティが 0.7% ほど改善されている。

次に費用と信頼度の改善度について述べる。第 4 表、第 3 表のシステム全体の MTSF, MTBF, MTTR, アベイラビリティをそれぞれのシステムを構成する費用で割り、単位価格あたりの信頼度尺度の値を求め、改良後と改良前の値を比較してみると、MTSF, MTBF, MTTR, アベイラビリティでそれぞれ約 6.45 倍、2.25 倍、1.14 倍、0.93 倍であり、MTSF, MTBF については単位価格あたりの信頼度が向上している。

〈5・2〉計算例 2 本節では〈3・2〉節で定式化した各情報ルートに制約条件を与えてシステムの構成費用を最小化する冗長設計法の計算例を示す。

第 4 表の冗長化する前の各情報ルートのアベイラビリティと MTSF の値をもとにして各情報ルートの MTSF を改善する。

いま $T_j=10^6$ としよう。すると (22) 式により B_j の第 1 項が計算される。また第 2 項は冗長化する前の各情報ルートのアベイラビリティを A_j' とすれば

$$\sum_{i \in S_j} b_{in} = l_n A_j'$$

により計算できる。

また (23) 式の c_i は〈5・1〉節と同様に、すでに二重化されているブロックのサブシステムを除いたものについて計算すればよい。

上記の 0-1 計画問題をアルゴリズム 1, 2 によって求めた結果、二つのアルゴリズムとも解は等しくなり、

第 3 表 費用制限のもとで情報損失量の期待値を最小にしたときのシステムの信頼度指標

Table 3. Reliability indices in the case of minimizing the expected information loss subject to cost constraints.

子局番号	MTSF(h)	MTBF(h)	MTTR(h)	アベイラビリティ
1	0.4147×10 ⁷	5,038	4.5	0.9991
2	0.4644×10 ⁷	5,236	4.2	0.9992
3	0.7815×10 ⁷	5,298	5.3	0.9992
4	0.1527×10 ⁷	4,945	4.9	0.9990
5	0.2820×10 ⁷	4,675	4.7	0.9990
6	0.3122×10 ⁷	5,450	4.4	0.9992
7	0.3396×10 ⁷	6,133	4.3	0.9993
8	0.2679×10 ⁷	3,872	5.0	0.9987
9	0.1888×10 ⁷	2,967	5.4	0.9982
10	0.6035×10 ⁷	4,805	4.3	0.9991
11	0.2125×10 ⁷	3,352	5.0	0.9985
12	0.1072×10 ⁸	4,496	4.5	0.9990
13	0.3841×10 ⁷	4,824	4.8	0.9990
14	0.2674×10 ⁸	4,443	4.9	0.9989
15	0.1671×10 ⁸	5,116	4.6	0.9991
16	0.3435×10 ⁸	5,135	4.6	0.9991

システムの MTSF 0.20981×10⁸(h)
 システムの MTBF 502.77 (h)
 システムの MTTR 5.9 (h)
 システムのアベイラビリティ 0.99275

第 4 表 第 1 図のシステムの信頼度指標

Table 4. Reliability indices of system in Fig. 1.

子局番号	MTSF(h)	MTBF(h)	MTTR(h)	アベイラビリティ
1	0.5514×10 ⁶	607.4	4.0	0.9934
2	0.5806×10 ⁶	610.2	4.1	0.9934
3	0.9568×10 ⁶	611.1	4.0	0.9935
4	0.2126×10 ⁶	580.9	4.2	0.9929
5	0.4016×10 ⁶	583.8	4.1	0.9930
6	0.3644×10 ⁶	613.1	4.0	0.9935
7	0.3274×10 ⁶	569.0	4.0	0.9930
8	0.4469×10 ⁶	545.3	4.1	0.9925
9	0.4179×10 ⁶	522.5	4.3	0.9919
10	0.7775×10 ⁶	569.0	4.2	0.9930
11	0.4237×10 ⁶	540.8	4.2	0.9923
12	0.1530×10 ⁷	564.3	4.1	0.9928
13	0.5457×10 ⁶	604.1	4.1	0.9933
14	0.4276×10 ⁷	597.6	4.1	0.9932
15	0.2316×10 ⁸	601.1	4.1	0.9932
16	0.4432×10 ⁷	608.8	4.0	0.9934

システムの MTSF 0.3020×10⁸(h)
 システムの MTBF 329.9 (h)
 システムの MTTR 4.8 (h)
 システムのアベイラビリティ 0.9856

二重化すべきサブシステムは $i=1$ (電源装置), $i=2$ (計算機本体) となった。

計算時間はアルゴリズム 1 では翻訳で 8.58 秒、実行で 1.47 秒、アルゴリズム 2 では翻訳で 2.64 秒、実行で 2.61 秒要した。実行時間についてみれば〈5・1〉節で述べた理由によりアルゴリズム 1の方が早い。

またこのときの所要費用は 1,171.8 である。

第 5 表はサブシステム 1, 2 を二重化したときのシ

システムの信頼度尺度を示す。各情報ルート の MTSF はいずれも $10^6(h)$ 以上になっていることが認められよう。

次に〈5・1〉節と同様の考えでシステム全体の信頼度尺度と費用の関係を述べる。改良後と改良前の単価あたりの信頼度尺度の値を比較すると、MTSF, MTBF, MTTR, アベイラビリティで 5.20 倍, 1.81 倍, 1.10 倍, 0.95 法である。

6. むすび

以上情報損失量の期待値を導入し、この値を費用制限のもとで最小化するようなシステムの冗長設計法、および各情報ルート の MTSF を所要の値にしてシステムの構成費用を最小にする冗長設計法を 0-1 計画問題として定式化する手法について述べた。また多重化の限度を二重化とすることにより制約条件式、変数の数を非常に少なくできる定式化について述べ実例を示した。この実例を二つのアルゴリズム 1, 2 で解いた結果、両者とも一致した。アルゴリズム 2 は準最適解を求める手法であるが、厳密な最適解を求めるアルゴリズム 1 より計算時間がずっと少なく、実用的で大規模システム向けであるといえる。今後新システムの信頼度設計、既設システムの設備計画などに本手法を活用していきたいと考えている。

最後に御指導頂いた東北大学工学部の穴山教授、木村正行教授に感謝する。

(昭和 51 年 5 月 31 日受付)

第 5 表 MTSF に制限を与えて費用を最小にしたときのシステムの信頼度指標

Table 5. Reliability indices in the case of minimizing cost subject to constraints of mean time between system failures.

子局番号	MTSF(h)	MTBF(h)	MTTR(h)	アベイラビリティ
1	0.4147×10^7	5,038	4.5	0.9991
2	0.4644×10^7	5,236	4.5	0.9992
3	0.7815×10^7	5,298	4.2	0.9992
4	0.1105×10^7	3,672	5.1	0.9986
5	0.2173×10^7	3,788	4.9	0.9987
6	0.3122×10^7	5,450	4.4	0.9992
7	0.1766×10^7	3,244	4.2	0.9987
8	0.1867×10^7	2,607	4.7	0.9982
9	0.1445×10^7	2,163	5.0	0.9977
10	0.4193×10^7	3,244	4.2	0.9987
11	0.1686×10^7	2,509	4.8	0.9981
12	0.7711×10^7	3,100	4.3	0.9986
13	0.3841×10^7	4,824	4.8	0.9990
14	0.2674×10^8	4,430	4.9	0.9989
15	0.1540×10^7	4,640	4.6	0.9990
16	0.3435×10^8	5,135	4.6	0.9991

システムの MTSF	$0.16661 \times 10^8(h)$
システムの MTBF	632.64 (h)
システムの MTTR	5.6 (h)
システムのアベイラビリティ	0.99127

文 献

- (1) 松坂, 他: 電学論 96-B, 15 (昭 51-1)
- (2) 松坂, 他: 電力技術研究会資料 PE-76-8 (昭 51-3)
- (3) 佐々木: 基礎システム工学 163 (昭 47) 共立出版
- (4) 玄, 他: 信学論 58-D, 1, 9 (昭 50-1)
- (5) 玄: 信学論 58-D, 7, 373 (昭 50-7)
- (6) 都築: 電中研報, 74089 (昭 50-6)
- (7) J. Kuester & J. Mize: Optimization Techniques with Fortran, McGraw-Hill, 91, (1973)
- (8) 黒田, 他訳: 整数計画法入門 52 (昭 51) 倍風館
- (9) F. Glover: Oper. Res. 13, 6, 879 (1965)
- (10) A.M. Geoffrion: SIAM Rev., 9, 2, 173 (1967)
- (11) Y. Toyoda: Management Science 21, 12, 1417 (1975)