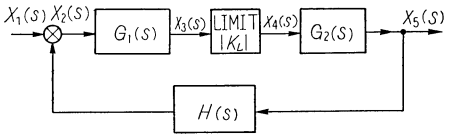
文　献　紹　介

A：数　値　解　析　　B：プログラミング　　C：計　算　機　方　式
D：回路および機器　　E：オートマトン　　F：応　用　そ　の　他

A-78.　非線型微分方程式の定差方程式による新解法

James M. Hurt: New Difference Equation Technique for Solving Nonlinear Differential Equations〔SJCC. 1964, pp. 169～179〕

第1図のような，非線型要素を含んだ自動制御系において，任意の入力 $X_1(s)$ に対する出力 $X_5(s)$ を求める問題を，Z変換を応用して，差分方程式を解くことに帰着させることができる．



第1図

1)　まず，リミタのない場合（線型の場合）を考える．サンプル周期Tを一つ固定して，幅Tのステップ入力に対する，この系のパルス伝達関数 $F(z)$ を次のように求める．すなわち，単位ステップ入力に対する出力を

$$X_5(s)=\frac{G_1(s)\cdot G_2(s)}{1+H(s)\cdot G(s)\cdot G_2(s)}\cdot\frac{1}{S}$$

に対応するz変換にサンプル周期 T を代入し，さらに，これをステップ入力に対するz変換，$z/(z-1)$で割って $F(z)$ を得る．これによってリミタのない場合の特性根，ゲインを得る．

2)　次にリミタのある場合（非線型の場合）を考える．

i)　この系の各線型要素の間には，サンプルがあるものと考える．

ii)　各線型要素のパルス伝達関数 $G_1(z)$，$G_2(z)$，$H(z)$ を計算する．ただし今考えている入力が，ステップ入力であるので，z変換の公式から得た式にサンプル周期Tをかけたものを用いる．特に $G_1(z)$ にはゲイン調整のために未知定数Kを入れておく．

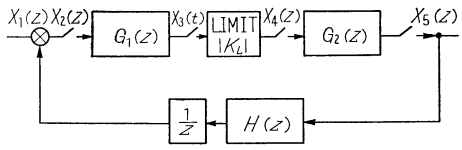
iii)　フィードバックループには，周期 T だけの遅れを入れるため，$H(z)$ にさらに $1/z$ を掛けたものを用いる．

iv)　このとき，この系全体のパルス伝達関数 $\tilde{F}(z)$ は

$$\tilde{F}(z)=\frac{G_1(z)G_2(z)}{1+H(z)G_1(z)G_2(z)\cdot\frac{1}{z}}$$

となる．ただし $G_1(z)$ は未知定数Kを含む．

v)　$\tilde{F}(z)$ の特性根，ゲインが $F(z)$ のそれと一致するように適当にKの値を定める．このとき，この系を表わすダイヤグラムは第2図のようになる．



第2図

vi)　これから

$$X_5(nT)\cdot\frac{1}{z}=X_5(nT-T)$$

などの関係を用いて，$X_1(nT)$，……，$X_5(nT)$ に関する差分方程式を得る．ただし $X_3(nT)$ と $X_4(nT)$ との関係は

$$X_4(nT)=\mathrm{LIMIT}_{|K_L|}X_3(nT)$$

となる．　　（米田英一）

A-79.　α を基数とする数を β を基数とする数に変換する算法

H.T. Gladwin: An Algorithm for Converting Integers from Base α to Base β.〔C.A.C.M., Vol. 7, No. 4 1964, pp. 241～242〕

二進法の計算機を使用するさい，8進数字と 10 進数字の間の変換を筆算で行なうことがある．この論文は非常に簡単な変換方法を紹介し，ついで，その方法を2個の任意の基数間の変換に一般化した次の定理を与えている．

定理　与えられた1より大きい自然数 α, β に対し

整数 $I=\sum_{i=0}^{n}a_i\alpha^i=\sum_{i=0}^{n}b_i\beta^i$

$I'=\sum_{i=0}^{n}a_i\beta^i$

を考える．ここで $0\leqq a_i<\alpha$, $0\leqq b_i<\beta$ ($i=0, 1, \cdots\cdots, n$) とし $\sigma=\alpha-\beta$ とおくと，$I'=I_0$ から $I=I_n$ を次の n 段階で求めることができる．

$I_k=I_{k-1}+\sigma\beta^{-1}\times$($I_{k-1}$ を β の多項式とみたときの β の指数が，$n-k$ より大きいすべての項の和)

$k=1, 2, 3, \cdots\cdots, n$, ただし，演算はすべて基数 β として行なう．

この定理の証明は I_k の β^m の係数が

$$\sum_{i=0}^{n-m}\binom{k-i}{k+m-n}a_{n-i}\sigma^{n-m-i},\quad k\geqq n-m$$

$$a_m,\quad k<n-m$$

で与えられることによっている．　（吉村一馬）

A-80.　多項式の計算法

J. Eve: The Evaluation of Polynomials 〔Numer. Math., Bd. 6, 1964, pp. 17～21〕

多項式 $P(x)=a_0x^n+a_1x^{n-1}+\cdots\cdots+a_n$ (a_i real) の値を計算するのに

$P_{i+1}=xP_i+a_{i+1}$, $i=0, 1, \cdots\cdots, n-1$

$(P_0=a_0)$

として計算する Newton-Horner の方法は，n 回の掛け算と n 回の加え算を必要とする．

1)　Knuth は次の方法で，〔$n+4/2$〕回の掛け算と n 回の加え算で計算できるようにした．

まず

$P(x)=Q_0(x)+Q_1(x)$

$Q_0(x)=a_0x^n+a_2x^{n-2}+a_4x^{n-4}+\cdots\cdots$

$Q_1(x)=a_1x^{n-1}+a_3x^{n-3}+a_5x^{n-5}+\cdots\cdots$

とおき

i)　$n=2m+2$ のときは

$S_0(x^2)=Q_0(x)$

$T(x^2)=Q_1(x)/x$

ii)　$n=2m+1$ のときは

$S_0(x^2)=Q_1(x)$

$T(x^2)=Q_0(x)/x$

で S_0, T を定める．

次に $T(x)$ の m 個の零点を $\alpha_1, \alpha_2, \cdots\cdots, \alpha_m$ とし

$S_{i-1}(x^2)=(x^2-\alpha_{m-i+1})S_i(x^2)+\beta_{m-i+1}$,

$i=1, 2, \cdots\cdots, m$ で定数 $\beta_1, \cdots\cdots, \beta_m$ を定める．S_m は

$S_m(x^2)=a_0x^2+\beta_0$　($n=2m+2$ のとき)

$=a_1$　($n=2m+1$ のとき)

となる．この α, β を用いると

$P_{i+1}=(x^2-\alpha_i)P_i+\beta_i$, $i=1, 2, \cdots\cdots, m$

$P_1=a_0x^2+a_1x+\beta_0$　($n=2m+2$)

$=a_0x+a_1$　($n=2m+1$)

$P=P_{m+1}$

となる．

2)　この Knuth の方法は，α_i がすべて実数でないときは意味がない．そこで Eve は本論文で，次の定理の成立つことを証明して，Knuth の方法に一般性をもたせた．

定理

多項式 $T(x)$ の m 個の零点は $P(x)$ の少なくとも $n-1$ 個の零点が半平面 Realpart$(x)\geqq 0$ (または Realpart$(x)\leqq 0$) にあると，すべて実数となる．

この定理により，$y=x+c$ (c は実数) なる変換を P にほどこせば，T のすべての根が実数となるのである．　（吉村一馬）

B-81.　Syntax に基づくコンパイリング

T.E. Cheatham, Jr. and K. Sattley: Syntax-Directed Compiling 〔SJCC. 1964, pp. 31～57〕

本論文は，まずある言語の syntax の適当な表現と，その言語の string を与えたとき，その string のその言語における syntactic analysis を自動的に行なうプログラムについて述べ，その後，そのプログラムの応用について述べている．

syntax を表現するには，Backus の Normal Form を用いる．この論文全体を通して，次のような簡単な syntax を持つ言語を例として使っている．

<program>::=<assignment>|<assignment>;<program>

<assignment>::=<variable>=<arith expr>

<arith expr>::=<ferm>|<arith expr>+<term>

<term>::=<tactor>|<term>*<factor>

<factor>::=<variable>|<integer>|(<arith expr>)

<variable>::=$\mathcal{V}$

<integer>::=$\mathcal{I}$

syntax の表現と具体的な string が与えられたときに，その string の syntactic analysis を行なうには，その string を左から scan してゆき，string 上の任意の位置で，それまでに scan した部分が syntax のどういう type の列になっているかを，与えられた

syntax の表現を参照して決定する．このようなことを string の最後まで行なうことにより，その string の syntactic な構造が決定される．

入力として syntax の表現だけでなく，それに対する semantics を文章で表現したもの（interpretive semantics）を入れておけば，与えられた string のその semantics による意味を決めることができる．また semantics を machine code を使って表現したもの（machine code semantics）を入れておけば，与えられた string を machine code に変換したものを得ることができる．

その他，1）誤りの検出，訂正，2）scan する順序と coding する順序の関係，3）declaration の処理，4）能率のよい coding というような問題につき簡単に触れた後，結論として，この方法は能率の点で operator-precedence technique に比しやや劣るが，object program の効率を第一の目標とする compiler においては，syntactic analysis を行なう時間の割合はきわめて小であるから，この欠点は，それほど重要でないこと，language の specification を変更したときにも algorithm そのものは変える必要はなく，入力である syntax の表現を変えるだけでよいという大きな長所があること，このような手法は，むしろ将来有用になるであろうということをあげている．（小林孝次郎）

B-82. 事項検索と推論して質問に答える情報検索システム

W.S. Cooper: Fact Retrieval and Doductive Question-Answering Information Retrieval System〔JACM. 11 (2) April, 1964, pp. 117～137〕

情報検索は，文書索引と事項検索の2つのタイプに分類することができる．事項検索では，ある事項に関する質問に答えること，つまり英語などの自然語の文章を論理的に推論し，その真偽を判定することが必要となる．

事項検索システムを構成するさいの主要な問題点は，いかに自然文の不明確さを取り除き，その論理的関係を正しく把握するかにある．

ここに報告する情報検索システムは，質問文つまり入力言語を英語のある型の平叙文に限定する．

入力言語は L-言語と呼ばれ，次の文法に従うものとする．

（1） 語彙は名詞 N，形容詞 A，形容名詞 C，動詞 V に分類される．

（2） $B=(A^1\cdot A\cdot\{\#AND\}^1)^1\cdot A$

（3） $M=\{\#A, \#AN\}^1\cdot B^1\cdot C^1\cdot N$

（4） $R=\{\#WHICH, \#THAT\}\cdot(\{\#IS, \#ARE\}\cdot\{\#NOT\}^1\cdot(M+B)+(\{\#DOES, \#DO\}\cdot\{\#NOT\})^1\cdot V)$

（5） $S=M\cdot R^1$

（6） $P=\{\#IS, \#ARE\}\cdot\{\#NOT\}^1(S+B)+(\{\#DO, \#DOES\}\cdot\{\#NOT\})^1\cdot V$

（7） $L=\{\#ALL, \#ANY, \#EACH, \#EVERY, \#NOT\#ALL, \#NOT\#EVERY, \#SOME, \#NO\}^1\cdot S\cdot P$

このうち語彙は，自由にとれるが，例では化学のテキストからとったものを示している．

このシステムでは，L-言語による質問文を，論理的推論に適した内部言語 L*-言語に変換して記憶し，推論を行なう．L*-言語は，$Axy(\equiv x\geqq y)$，$Exy(\equiv x\cap y=0)$，$Ixy(\equiv x\cap y\neq 0)$，$Oxy(\equiv x\not\subset y)$ の4つを形成を持つ2項論理函数である．

L-言語の質問に対し，L-L* 変換を行なって，記憶してある L*-語の文章を探索し，質問文を論理的な帰結とする文章がみつかれば TRUE，矛盾する文章がみつかれば FALSE と印字して，みつかった文章の対を印刷する．

この論文では，L-L* 変換を行なうための変換函数をいくつか与え，これに基づいて IBM 7090 でプログラムした変換のアルゴリズムについて述べている．（新井克彦）

B-83. 資源が限定されている場合の日程計画上の方法

Jerome D. Wiest: Some properties of schedules for large projects with limited resources〔JORSA, Vol. 12, No. 3, 1964, pp. 395～418〕

現在 CPM，PERT，その他アロー・ダイヤグラムを用いたプロジェクトの日程計画が開発されているが，いずれも資源（労働力，機械台数など）が制限されていない．この論文では，これらの技法を，実際に資源が限定されている場合に，いかに拡張していくかを論じている．大略は

1．スケジュール・チャートの作成
2．スラックの計算
3．クリティカル・シーケンスの決定

の3つに分け，定められた作業順序，使用可能な資源

のもとで，日程をいかに最小にしていくかを展開している．

第1図

スケジュール・チャートは，いわゆる，ガント・チャートにアロー・ダイヤグラムの機能すなわち，作業の前後関係をもたせ，さらに資源の制限をつけ加えている．資源が制限されている場合，CPM，PERT のスラックでは，資源の不足により，日程上遅れを起こす．ここでは，‘Job Shifting’という概念を導入して，新スラックの計算を行ない，このスラックにより，リクリティカル・シーケンスを決定している．Job Shifting 操作は，資源の制限内で行ない，右移動の場合，完成日に影響を及ぼすことに注意がいる．この操作によるスケジューリングを行ない，スラックを次のように決定する．

$$TS_j(g,\ h)=LS_j(h)-ES_j(g)$$

$g\epsilon J_L$，$h\epsilon J_R$，J_L は左正規移動された activity の集合，J_R は右正規移動された activity の集合．

クリティカル・シーケンスは，この意味におけるスラックが0の集合である．実際のスケージューリングでは，左（右）正規移動ではなく，部分的な左または右移動を行なうことにより，日程の長さを最小化する方法が詳しくのべられている． （深野拓一）

B-84. OGO 衛星のための実時間高速処理プログラムシステム

R.J. Coyle and J.K. Stewart: Real Time quick-look Analysis for the OGO Satellites〔SJCC. 1964, pp. 125～138〕

OGO 衛星は 50 種類の測定器を積んで，地球物理関係の観測をすることを目的としている．それが発するデータは，2箇所の地上局を通ってすべて制御センタの計算機へ送られ，必要な処理をされる．このプログラミングシステムの目的は，衛星と測定器の状態をすみやかに解析し，それらを制御することである．

採用した SDS 920 は小形計算機で，24 ビット語8千語，サイクル時間 8 μs，入出力用に 32 の割込みチャネルをもっている．これには特別のコンソールがついていて，計算機—衛星間の情報交換は，すべてこれを通して行なう．データの伝送速度は 64 kc で，18 ms ごとに1フレームが送られてくる．

プログラムは実時間モニタ制御と測定データ処理の2つの部分からなる．そして前者が完全に後者を制御している．また，実時間モニタ制御は，モニタ計画プログラム，モニタ処理，割込み処理の3つに分けることができる．

（1） モニタ計画プログラム　これは優先表を用いてシステム内のすべてのルーチンを制御し，監視し，計画するものである．割込みが起こると，計算機はそのときの状態を保存し，割込みから戻ったときに以前の状態を回復する．優先表は割込みの優先順位が異なる各ルーチンを代表するモジュールのリストである．モジュールはリンク，指示ビット（ルーチンの状態を表わす），ルーチン自身の頭の番地をそれぞれ情報として含んでいる．リンクを用いて優先順位の最も高いルーチンを選ぶのに，間接アドレス命令を巧みに用いている．

（2） モニタ処理　これは入出力関係の処理を扱う．SDS 920 は，バッファのある入出力用の2チャネルと割込み用の 32 チャネルをもっている．入出力用のうちの1チャネルは，測定データのためのものである．これも前記のモニタ計画プログラムの制御を受ける．

（3） 割込み処理　32 の割込みチャネルは，それぞれがさらに優先順位の高いチャネルによって割込みされることがある．割込みが何重にも起こったときは，一つの処理が終っても，もとの割込みルーチンへ戻るので，このときに限り，モニタ計画プログラムの制御を受けないことになる．すべての割込みルーチンを処理すると，計画プログラムへ戻って，再び優先順位の高いルーチンをさがすことになる．

（4） 測定データ処理　これは多くの独立なルーチンから成っており，その機能は，測定器および衛星自身の状態に関する入力データの処理である．磁心記憶装置に収めきれないほどこのルーチンはぼう大になるので，必要に応じて磁気テープから出し入れする．このさいに実時間処理を妨げないように配慮している．

（金山 裕）

B-85. 汎用索表形コンパイラ

S. Warshall and R.M. Shapiro: A General Purpose Table Driven Compiler〔SJCC. 1964,

p. 59～65]

コンパイラに能率のよいオブジェクト・プログラムをつくりださせるための最適化手続にはいろいろなやり方がある．入力ストリングそのものを対象とする方式や，入力ストリングの syntax を樹構造に直したものを対象とするもの，あるいはその樹構造からつくったマクロ命令群を最適化するものなどをあげることができる．

ここで述べているコンパイラは，最後のタイプに属する一種の syntax directed compiler である．

このコンパイラは，次のような5個の論理的段階に分かれている．

Phase I: Analyzer

入力ストリングを分析して，その文法規則を樹構造に直す部分で，ソース言語の文法規則を Backus normal form で書いたものと，本質的に同等であるようないくつかの表を使って，文法分析を行なう syntax-directed analyzer である．

Phase II: Generator

ここでは Phase I で作った樹構造から，n-アドレスのマクロ命令を発生する．そのためには，発生規則に関する情報を含んだ1組の表（generation strategy と呼んでいる）を利用する．

Phase III. In-Sequence Optimizer

Phase II の出力であるマクロ命令群を最適化するのがこの段階の仕事になる．それらのうちには，コンパイル時に値を決めることのできるようなものをみつけだして処理することとか，一連のマクロ命令群にわたるある程度 global な最適化，すなわちその部分全体を通じて値が変わらないような計算を見つけたり，インデクス・レジスタなどの使い方を調べて，それら特殊レジスタに関する不必要な呼び出しを省いたりするようなことが含まれる．

Phase IV: code selector

マクロ命令群を一連の記号機械命令に変える部分で，ここでは code selector strategy と呼ぶ表が主要な役割を演じる．機械固有の各レジスタに何がはいっているかを常に把握して，できるだけむだのないような命令群をつくることもこの部分の仕事になる．

Phase V: Assembler

Phase IV の出力を処理して適当な形式でオブジェクト・プログラムをつくりだす．つまり指定によって記号プログラム，絶対番地あるいは相対番地方式のオブジェクト・プログラムなどを出力することができ，記号表などをいっしょに出すことも可能である．

さて，このコンパイラを特定のソース言語と特定の計算機の組合せに対して具体化する場合，コンパイラ全部をいちいち手で書いていてはたいへんなので，適当な方法で機械化しなければならない．ここでは syntax-directed compiler の特性を利用した bootstrap 方式について述べている．すなわち初めに最小限の部分を書き，それらをもとにして何回も機械処理を繰り返し，漸次 full scale のシステムに近づけるやり方である．

このコンパイラでは，文法規則の記述や，マクロ命令およびそれからの機械命令の発生などの規則を記述するためにいくつかの言語体系をもっており，その実例も紹介されている．

最後に CDC-1604 による CDC-1604 のためのコンパイラを，IBM-7090 によって作りあげた過程の説明があり，きわめてむだの少ない，オブジェクト・プログラムを出すことができたと報告されている．

（吉村鉄太郎）

C-86. 剰余類の理論による計算機の性能向上

R.D. Merrill: Improving Digital Computer Performance Using Residue Number Theory [IEEE. Trans. EC-13, No. 2, April, 1964, pp. 93～101]

この論文では，整数を表現するのに通常行なわれている2進の表現と，剰余類の組合せによる表現を併用して，計算機の性能を向上させる方法について詳述している．整数を剰余類の組合せによって表現する方法は，整数どうしの加算，減算および乗算を行なうとき，それぞれの剰余類の成分ごとに独立にその演算を行なえばよく，したがって carry の処理が不要であるという大きな利点を持っている．その反面，整数の正負を決定したり overflow を検出したりするのに，従来の表現の場合に比し，複雑な操作を必要とするという欠点を持っている．したがって整数を剰余類の組合せによって表現する方法は，加減算，乗算が多く，scaling を行なう必要が少ないような計算，たとえば連立一次方程式の解や，逆行列の値を exact に求めたり，相関関数を求めるような計算に対して有利である．

それぞれの剰余類の底を選ぶには

（i）2^n または 2^n-1 という型の数であること（こ

のように選ぶことにより従来の処理方式とあまり違わない方式で，算術演算を処理することができる)，

(ii) 互いに素であること（表示できる数の範囲が底の積になる)，

(iii) その底の組合せによって表示できる数の範囲が，2のある累乗になるべく近いこと．

という3つの点に注意する必要がある．

算術演算は，それぞれの剰余類に対応した Adder と shift register を使って，各成分ごとに次のようにして行なわれる．

(i) 加算は，2^n が底の場合には，単なる2進数としての加算，2^n-1 が底の場合には，end-around carry の加算によって行なう．

(ii) 減算は，2^n 底の場合には，2の補数表示になおしたものの加算，2^n-1 が底の場合には，1の補数表示になおしたものの加算によって行なう．

(iii) 乗算は，shift しながら加算することによって行なう．ただし底が 2^n-1 のときには，shift は end-around で行なう．

剰余類の底として，128, 127, 63, 31 の4つを使った場合，通常の表現の場合に比し，加減算時間は 1/3 に乗算時間は 1/12 になる．また連立一次方程式の解を求めるに要する時間は，次数が十分大きい場合には通常の表現の場合に比し 1/6 になる．

(小林孝次郎)

E-87 単純な線の図形のエントロピー

W.H. Foy, Jr.: Entropy of Simple Line Drawinge 〔IEEE, Trans. IT-10 No. 2, April, 1964, pp. 165～167〕

パタン認識においては，対象とするパタンのもつ情報量を知ることが重要になってくる．本論文は連続な線からなる図形のエントロピーを計算したものである．

正方形の図形をN個の正方形のセルのマトリクスに分割し，そのセルの黒白によって図形を表わす．

黒いセルと白いセルが等確率でランダムに（必ずしも連続でない）きまるとすると，この図形のもつエントロピーは

$H_{TV}=N$ ビット

となる．

しかし，一般には全マトリクスのうちの黒い部分は，ごくわずかを占めるにすぎない．平均して N_P 個のセルが黒いものとすれば，その図形のもつエントロピーは

$$H_{den}=N\left\{p\log\frac{1}{p}+(1-p)\log\frac{1}{1-p}\right\}\ \text{ビット}$$

である．

しかし，以上の計算では図形は必ずしも連続ではないから，次の連続図形のエントロピーより大きくなっている．以下に述べる条件が図形に関して成立するものとする．

1. 図形は何本かの連続な線からなっている．その本数は

$P(\alpha)=$prob（線の本数$=\alpha$), $\alpha=1, 2, \cdots\cdots$

なる確率分布に従う．

2. おのおのの線の長さ（セルの個数）は

$P(\beta)=$prob（1本の線の長さ$=\beta$), $\beta=1, 2, \cdots\cdots$

なる確率分布に従う．この確率はαとは独立に与えられる．

3. 「連続」とは，セルの一辺を共有することを意味する．

4. 線の両端は他の線上にない．

5. 2本の線は1個のセルしか共有しない．

6. どの線も5セル以内の長さのループをもたない．

以上の仮定から計算したエントロピーは

$$\begin{aligned}H_{cont}=&-\sum_{\alpha}P(\alpha)\log P(\alpha)-\sum_{\alpha}P(\alpha)\log(\alpha!)\\&+\sum_{\alpha}\alpha P(\alpha)\Big\{-\sum_{\beta}P(\beta)\log P(\beta)\\&+\Big(\sum_{\beta}\beta P(\beta)-2\Big)\log\Big(\frac{25}{9}\Big)\\&+\log 2N\Big\}-\frac{9}{25C}\Big\{\sum_{\beta}\beta P(\beta)-1\Big\}\\&\Big\{\sum_{\beta}\beta P(\beta)\sum_{\alpha}(\alpha-\sum\alpha P(\alpha))^2P(\alpha)\\&+\sum_{\beta}\beta P(\beta)\sum_{\alpha}\alpha P(\alpha)\Big\{\sum_{\alpha}\alpha P(\alpha)-1\Big\}\\&+4\sum_{\alpha}\alpha P(\alpha)\sqrt{N}\Big\}\\&-\frac{625}{648}\sum_{\alpha}P(\alpha)\sum_{\beta}P(\beta)(\beta-1)\Big(\frac{9\sqrt{3}}{25}\Big)^{\beta}\end{aligned}$$

ここで

$PN=\sum_{\alpha}P(\alpha)\sum_{\beta}P(\beta)$

とすると，$P<\frac{1}{2}$ のとき $H_{cont}<H_{den}$ となる．具体的な数値例をあてはめてみる．マトリクスが $2^{12}\times2^{12}$($N=2^{24}$) であって，線の本数は1本から50本まで一様分布，線の長さは平均 6700 セルのポアソン分布であるとする（$P=0.0102$)．このとき

$H_{TV}=167.5\times10^5$ ビット

$H_{den}=13.76$　〃

$H_{cont}=2.52\times10^5$ ビット

連続の仮定をとり入れても，図形のエントロピーはかなり大きい．しかし，この計算の途中で明らかになることは，エントロピーを減らすためには線の結合状態を考慮する必要があるということである．特定の応用ないし符号化の方法には触れていない．（金山　裕）

F-88. 計算機による設備レイアウト

Elwood S. Buffa, Gorden C. Armour and Thomas E. Vollmann: Allocating Facilities with CRAFT〔Harvard Business Revieu, Vol. 42 No. 2, 4, 5, 1964, pp. 136～158〕

最も有効な設備レイアウトを数学的に求める方法とそのための計算機プログラム“CRAFT”の紹介と説明である．ここで“最も有効な”とは設備間の物の運般費用総計が最小であることを意味する．

いま，　n＝設備総数

v_{ij}＝設備 i, j 間の運般量

u_{ij}＝設備 i, j 間で1単位の物を1単位距離運般するに必要な費用

d_{ij}＝設備 i, j の各中心間の距離とする．

i, j 間の単位距離運般費用は $y_{ij}=v_{ij}u_{ij}$ となる．そこで総費用 TC_0 は

$$TC_0=\frac{1}{2}TC=\sum_{i=1}^{n}\sum_{j=1}^{n}y_{ij}d_{ij}$$

で表わされる．これを最小とする各中心の位置を決めればよい．

ここで適当な制限式を仮設することができれば，この問題はいわゆる Quadratic integer programing problem に帰着するが，この適当な解法はみつからない．また，$E_k=\sum_{i=1}^{n}\sum_{j=1}^{n}y_{ij}d_{ij}(k)$ とし，$d(k)$ を (k) なるレイアウト下の距離とすれば，(k) のすべての場合の E_k を計算し，最小となる (k) をとればよい．この場合 (k) の数は $n!/M$ となる．ここで M は場所の対象性によって決まる小さな値である．この場合 (k) は非常に大きな数となり，実行は不可能である．そこで全設備中2つの設備の組について，それを入れ替えた場合に減少する総費用 ΔTC_{ef} を計算し，その中から ΔTC が最大となる組の入れ替えを行なう．このための計算は ${}_nC_2$ 回行なえばよい．入れ替えをした時点で再び次の step を行ない，ΔTC がすべて負または0となるまで行なうことによって，最も有効なレイアウトを得ることができる．ここで設備の入れ替えが不可能な場合が起こる．この場合は最大 ΔTC ではないが，可能であるものの中から組を選ぶ必要がある．

以上の Process を IBM 7090 用 program としたものが CRAFT である．この program によって設備数が 20 の場合，1分以内の計算で減少費用 30% 近いレイアウトが得られた．（高橋　宏）

F-89. セメント窯におけるブレンディングの計算制御

IAN HAY: Computer Control of Cement Kiln Feed Blending〔Instrument & Control System, Vol. 37, No. 7, 1964, pp. 134～138〕

セメントクリンカは，いくつかの酸化物を主成分とする原料岩石が粉粋，混合され，さらに，これが焼成されて作られる．“要求される組成分をもつクリンカを作るのに，各種の原料の供給比をいくらにすべきか”が問題であり，これの計算制御を考える．

いま原料岩石は4種（j で示す）であるとする．その主成分は，CaO, SiO_2, AL_2O_3, Fe_2O_3, MgO の5種（i で示す）の酸化物であり，一方，クリンカの混合組成物は C_3S, C_3A, C_2S の3種（k にて示す）が，そのほとんどである．

f_j＝原料入れ j からの原料供給比

q_{ij}＝供給量 j における酸化物 i の重量分率(既知)

S_i＝原料岩石 j における全不揮発性物質

γ_i＝混合原料における酸化物 i の重量分率

γ'_i＝クリンカにおける酸化物 i の重量分率

とすれば

$$\gamma_i=\sum_{j=1}^{4}q_{ij}f_j,$$

$$\gamma'_i=\sum_{j=1}^{4}f_jq_{ij}/S_j=\sum_{j=1}^{4}q_{ij}F_j,\quad(F_j=f_j/S_j)\qquad(1)$$

となる．一方，c_k をクリンカにおける混合物 k の重量分率とすれば，c_k は γ'_i の函数として

$$c_k=\sum_{i=1}^{5}\gamma'_i\Phi_{Ki}\quad(\Phi_{Ki}\text{ は係数値として与えられる})$$

として与えられる．(1) の式から，

$$c_k=\sum_{i=1}^{5}\Phi_{Ki}\sum_{j=1}^{4}q_{ij}F_j=\sum_{i=1}^{5}\sum_{j=1}^{4}\Phi_{Ki}q_{ij}F_j\qquad(2)$$

さらに定義より

$$\sum_{i=1}^{5}\sum_{j=1}^{4}q_{ij}F_j=1\qquad(3)$$

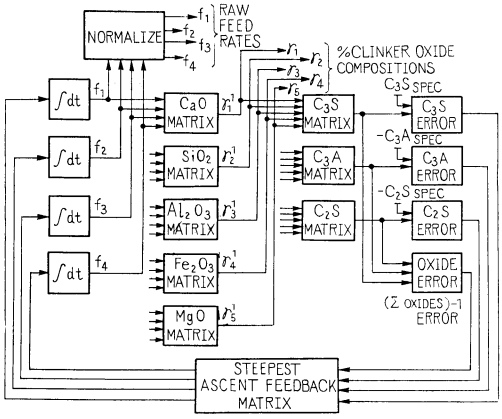
なることから，この式と(2)の式の4式から F_j が求められ，さらに既知の S_j の値から f_j の値が得られる．

この計算制御システムにおいては，アナログ計算機を用い，（2），（3）式から定義される誤差の自乗和を0にするよう“山登り法”を用いて制御している．すなわち

$$E_k=\left[\sum_{i=1}^{5}\sum_{j=1}^{4}\Phi_{Ki}q_{ij}F_j\right]-c_k,\quad 2E_k/2F_j=\sum_{i=1}^{5}\Phi_{Ki}q_{ij}$$

$$E_4=\sum_{i=1}^{5}\sum_{j=1}^{4}q_{ij}F_j\Big]-1,\quad 2E_4/2F_j=\sum_{i=1}^{5}q_{ij}$$

として $S=g_2E_1^2+g_3E_2^2+g_3E_3^2+g_4E_4^2$（$g_j$ は weighting factor）をできるだけ早く最小にするよう，つねに ds/dt（t は dummy variable）が負となるようなアナコム・システムを作り，f_j の値を得るようにしている．そのブロック図は，つぎのようなものである．



第1図

なお文献では，この“山登り法”の具体的なテクニック，そのための基本回路，さらには，Dead-Bandのシミュレーションなどに関しても説明が加えられている．

（北畠　宏）