

研究員報告書

第 5 号

(1984.3)

岡 山 理 科 大 学
情報処理センター資料室

目 次

1. モリブデン錯体のX線結晶解析

化 学 科 助 教 授 柴 原 隆 志…………… 1

2. 明朝体漢字をもとにした大型ゴシック体漢字の生成と表示

電 子 理 学 科 助 教 授 塩 野 充…………… 3

3. 分権システムの解法アルゴリズム

電 子 理 学 科 教 授 成 久 洋 之…………… 8

4. 試験の採点と集計を行うプログラム

教 養 部 教 授 浦 上 澤 之…………… 22

5. Software Reliability : S-shaped Software Reliability Growth Models and Their Comparisons

Graduate School of Systems Science Shigeru YAMADA …… 35

6. 最近傍則を用いた手書き漢字カテゴリー候補の生成

電 子 理 学 科 教 授 橋 本 礼 治…………… 45

モリブデン錯体のX線結晶解析

化学科助教授 柴原隆志

最近4軸型自動X線回折装置(AFC-6A型)を設置したが、この装置より得られるデータを用いて、構造解析を行うために、必要ないくつかのプログラムを作製した。データの流れおよびプログラム名(太字)を以下に記す。

関連機器

1. 反射強度データ採取

AFC-6A

2. **convert** : AFC-6Aで用いたデータディスク(両面単密)から
PC-9801E用にフォーマットしたディスク(両
面倍密)へデータに移すためのプログラム。

PC-9801E

addition : 反射データを追加するためのプログラム。

tool : データファイルの内容をディスプレイに表示するための
プログラム。

3. **express** : データをPCからMELCOMへ送るためのプログラム

PC-9801E

MELCOM

COSMO 800 III

4. **EDITOR** : **express** で送られた反射強度データを構造解析用各
種プログラムで使用するために編集するプログラム。

MELCOM

COSMO 800 III

convert, **addition**, **tool**, **EDITOR** は新たに開発し, **express** は既存のものを改造した。

現在、構造解析プログラムMULTAN 78の導入を計画、実行中である。

これらのプログラムは本学化学科4回生、桜庭直也君が卒業研究の一部として作製したものである。プログラム作製にあたり、多大の御指導をいただきました本学情報センター木村助教授に感謝いたします。

(様式 1)

– 2 –

明朝体漢字をもとにした 大型ゴシック体漢字の生成と表示

Generation and display of large size Gothic type KANJI
patterns using Ming type KANJI patterns

電子理学科助教授 塩 野 充
Mitsuru SHIONO

近年、電子計算機による日本語情報処理が一般化するにつれ、従来は人間が手書きで清書していたようなものでも、計算機の印刷出力のまま掲示板等に貼って使用することが多くなった。その際、見出しや標題として大きな文字を表示したい場合や、入試の合格発表のように、掲示位置と見る人の距離が離れているので大きな文字で表示しなければならない場合がある。そのような場合、計算機に内蔵されている漢字メモリの明朝体ドットパターンを次数変換等により拡大して表示する方法〔1〕〔2〕が使われることが多い。しかし、漢字パターンを表わす画面次数が通常、 24×24 (24 ドットとよぶ) や 32×32 (32 ドットとよぶ) 程度の小さいものなので、あまり大きく拡大するとアラが拡大されてどうしても粗雑で不自然なものとなってしまう。又、屋外の看板や標識等で分るように、大きな文字の場合は明朝体よりもゴシック体の方が視認性がよいことは明らかである。本研究では現在、漢字メモリとしてパソコンやワープロの間でも最も普及している 24 ドット明朝体パターンを用いて、任意の大きさの大型ゴシック体漢字を発生、表示する方法を提案し、実験を行う。本方式の特徴は、明朝体漢字パターンから心線図形を抽出し、その心線を直線と円弧だけで表現し、それに線幅を付加することにより、グラフィックプリンタによる容易な描画を可能にしている点である。即ち、ドット集合による漢字パターンのデジタル表現を、直線と円弧という幾何学的なアナログデータ集合に変換していることである。

図 1 に本方式のジェネラルフローを示す。入力は 24 ドットの明朝体の 2 値図形 (白=0, 黒=1) である。まず、ステップ (a) で 4 方向の方向コード化〔1〕を行う。これは次のステップ (b) の前処理として行うもので、図 2 の方向コード (1, 2, 3, 4) を用いる。(b) で図 3 に示す 7 種類の論理マスクを用いてウロコ等と呼ばれる明朝体飾りを除去する。これらの飾りは正確な心線図形を抽出する際の妨げとなるからである。同図で△印は don't care を、X は正整数を表わし、マスクの条件を満たす場合は○で囲んだ画素を 0 にする。使用する明朝体フォントのメーカーによって飾りの形状に若干の差異があるかも知れないがその際は論理マスクの僅かな変更で対処出来る。又、32 ドットの明朝体を使用する場合も同様で、これらの論理マスクに若干の変更を加えればよい。(c) では再び方向コード化を行う。これは飾りを除去した後の方が、以降の処理のために正確な方向コードが得られるからである。(b) では筆者らが以前発表した直角性保存細線化〔1〕を施して、漢字の心線図形 (字体) を T 型部分や L 型部分、+ 型部分等の直角性を保存した状態で得る。Hilditch〔5〕, Rosenfeld〔6〕, Deutsch〔7〕等の通常の細線化法 (8 連結) では直角性は保存されない。

後 2 者の細線化法では 4 連結も可能であり、4 連結で細線化を行うと直角性は保存されるが、斜め線分も 4 連結となり本方式の目的に適さない。本方式では垂直水平成分については 4 連結、非垂直水平成分については 8 連結となる細線化が必要である。(e) では直角性保存細線化の後処理として局所的な整形を行う。整形の論理マスクを図 4 に示す。(f) ではある閾値 r 以上の長さの垂直(方向コード 2) および水平(方向コード 4) の黒ラン(垂直水平成分と呼ぶ)を抽出する。(g) では方向コード 2, 4 を 1, 3 に併合して斜め線分(非垂直水平成分と呼ぶ)の抽出の前処理をする。(h) では非垂直水平成分の抽出を行う。(i) では抽出した非垂直水平成分に対して両端と中間点の 3 点による円弧の当てはめを行う。(j) では(f) で得られた垂直水平成分と(i) で得られた円弧を元にして、ゴシック体のアナログ表現を行う。(k) では倍率を掛けることによって所望の大きさにサイズ変換を行う。(l) ではそれを用いて BASIC 言語の描画命令(グラフィックコマンド)〔8〕により、CRT 画面上にゴシック体漢字を描画する。それをハードコピー命令により用紙上の必要な場所に印刷出力する。

以上、日本語情報処理による掲示用印刷出力のために、既存の 24 ドット明朝体漢字パターンを用いて、大型のゴシック体漢字を生成、表示する一手法の概略を紹介した。本手法を使えば、会議や式典などの各種行事の式字幕や案内標識の作成、および掲示による各種情報の伝達がパソコンプリンタ等で簡単、迅速に行うことが可能である。毛筆で手書きするときのように特別な書道熟練者を必要としない。本稿では漢字メモリとして最も普及していると思われる 24 ドットの漢字パターンを用いたが、勿論 32 ドットや 16 ドットの漢字パターンを用いても同じことが可能である。今後の課題としては、より洗練されたゴシック体、即ち少しでも本物のゴシック体に近いパターンが得られるように各部のアルゴリズムの改善、および今回は角ゴシックの表示だけであったが、丸ゴシック体の表示も出来るようにしたいと考えている。

文 献

- 〔1〕 塩野, 真田, 手塚: “漢字ドットパターンの次数変換と整形の一手法”, 信学論 J63-D, 7, pp. 557-561 (1980-07)
- 〔2〕 栃木敏子: “日本語文書拡大出力用 POSTER コマンドの使い方”, 東京大学大型計算機センターニュース, 16, 5, pp. 40-42 (1984-05).
- 〔3〕 塩野, 橋本: “明朝体漢字パターンを用いた大型ゴシック体漢字パターンの発生”, 昭 59 電気関係学会関西支部連合会, G8-13, (1984-11)
- 〔4〕 塩野, 橋本: “24 ドット明朝体漢字をもとにした大型ゴシック体漢字の成生と表示”, 第 15 回画像工学コンファレンス, 13-15, pp. 283-286, (1984-11)
- 〔5〕 塩野, 真田, 手塚: “明朝体漢字パターンからゴシック体漢字パターンへの字体変換の一手法”, 信学論 J65-D, 11, pp. 1366-1373 (1982-11).
- 〔6〕 C. J. Hilditch: “Linear Skeleton from Square Cupboards”, Machine Intelligence N, New York, pp. 403-420 (1969).

斗め
垂直
とし
(方
では
h)
3点
を元
てサ
に
場

を用
義や
プリ
きを
ノを
課
られ
ック

,7,
幾セ
9電
第
デー
てー

- [7] A. Rosenfeld and L. S. Davis : "A Note on Thinning", IEEE Trans. Syst. Man, cybern. , vol .SMC -6, pp . 226 -228 (1976)
- [8] E . S . Deutsch : "Thinning algorithms on rectangular, hexagonal, and triangular arrays ", Communications of the ACM, vol . 15, 9, pp . 827 -837 (1972)
- [9] 日本電気(株) : PC9801 E BASIC REFERENCE MANUAL (1984)

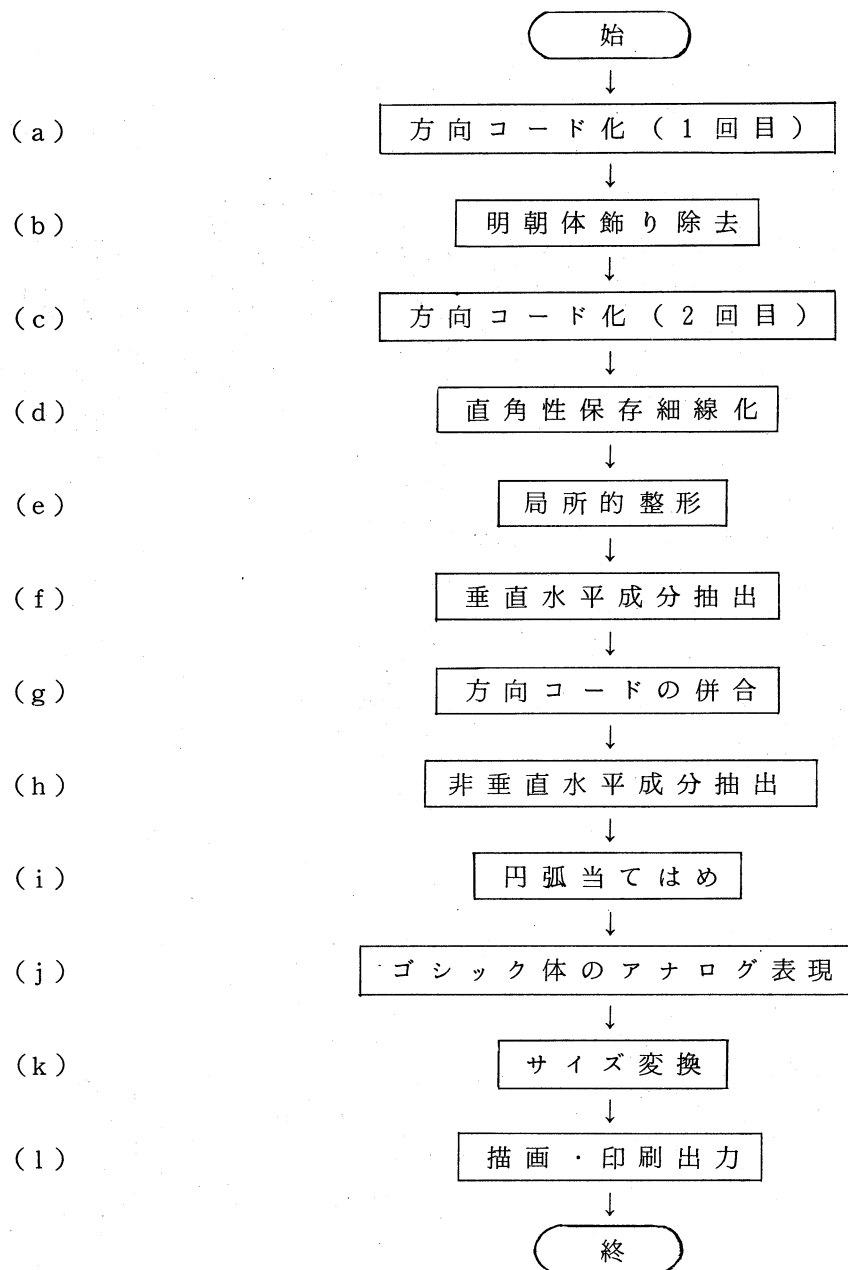


図1 本方式のジェネラルフロー

Fig . 1 - Genaral flowchart of the proposed system .

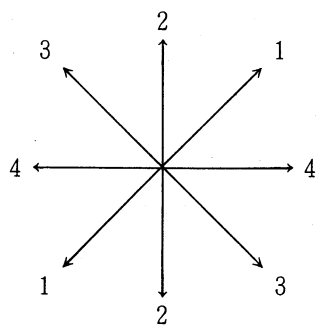


図2 方向コード

Fig. 2—Direction code.

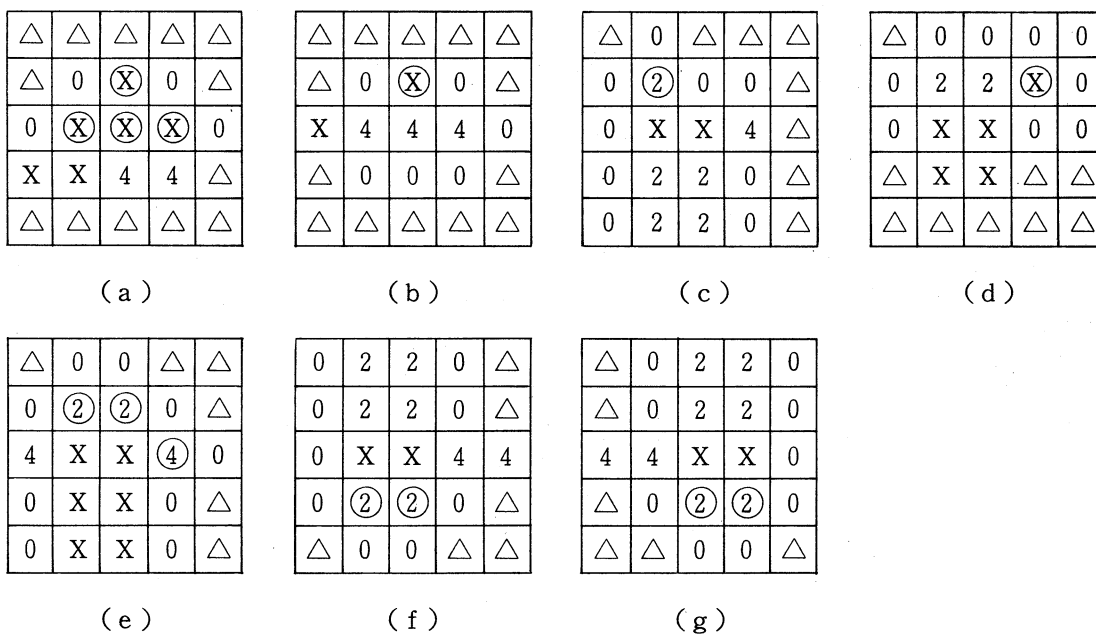


図3 明朝体飾り除去の論理マスク

Fig. 3—Logical masks for ornaments removal.

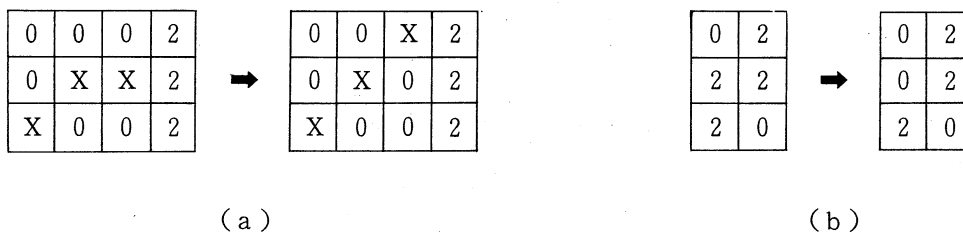


図4 局所的整形

Fig. 4—Local shape adjustment.

分権システムの解法アルゴリズム

電子理学科教授 成 久 洋 之

1. はじめに

大規模な線形計画問題を現実的に解こうとすると、殆んどの場合に構造化された問題として捉えられる。構造化されたしかも大規模な問題であれば出来るだけ分割した小問題あるいは部分問題として解かせる方が計算処理時間やその他の理由より効率的であるとされている。一方問題解法アルゴリズムとしての効率性と同時に組織形態や運用管理形態からみると、中央組織とその下部組織とに階層化して捉えることが出来る。その場合、中央が下部組織を管理・統制する方法が如何なる要領で実施されるべきかは中央集権的にするかあるいは分権的なものにするかの度合によりかなり違ったものとなる。

分権システムでの問題解決法は中央の意向を考慮しながら下部システム的意思を重視した決定法であり、中央と下部システムの情報のやりとりを可能な限り少なくするような方法を志向するものである。

問題解法アルゴリズムの観点からみると、Dantzig - Wolfeの分解原理に基づく解法はいはば中央集権的なアルゴリズムとみなされており、下部組織は中央組織のための情報収集機関とみなしたものである。それに対し、本論で提案しようとしているアルゴリズムは分権システムそのものの意思を尊重したself - automonousなシステムの解法アルゴリズムであり、全体組織としての最適解に至らずとも各下部組織体の決定を重視した決定法であり、中央の方針には従うが中央から介入の度合の少ないアルゴリズムを提案しようとするものであり、その意味でより現実的解法手法といえる。本研究結果では従来の方法で求めた処理時間の1/10程度の時間で最適値の5%程度の近似解が求められている。

2. 問題の定式化

構造化された線形計画問題をつぎのように定義する。

$$\begin{aligned} \max \quad & \sum_{j=1}^n C^j x^j \\ \text{(P)} \quad \text{s.t.} \quad & \begin{cases} \sum_{j=1}^n A^j x^j \leq b^0, \\ B^j x^j \leq b^j, \quad x^j \geq 0 \end{cases} \\ & (j=1, 2, \dots, n) \end{aligned}$$

ただし、 A^j , B^j は $m^0 \times n^j$ 行列、 $m^j \times n^j$ 行列、 b^0 , b^j , C^j , x^j はそれぞれ、 m^0 次元、 m^j 次元、 n^j 次元ベクトルとする。

ここで、 $z^j = C^j x^j$ とすると〔P〕をさらに、つぎの2つの問題として考える。

$$〔T〕 \quad \begin{cases} \max & \sum_{j=1}^n z^j \\ \text{s.t.} & \sum_{j=1}^n A^j x^j \leq b^0, \quad x^j \geq 0, \quad (j=1, 2, \dots, n) \end{cases}$$

$$〔D〕 \quad \begin{cases} \max & z^j \quad (= C^j x^j) \\ \text{s.t.} & B^j x^j \leq b^j, \quad x^j \geq 0, \quad (j=1, 2, \dots, n) \end{cases}$$

この2つの問題〔T〕と〔D〕はそれぞれ、中央問題と部分問題と呼ばれる。

問題〔P〕を〔T〕と〔D〕とに分けて考慮することにより、〔P〕を直接解くことの困難さを克服しようとするものであり、システム形態としては中央統制機能と地方分権機能との関連性を規定できるシステムの定式化となっている。つまり、〔D〕は j 番目の部分問題が〔P〕よりはかなりの小規模の線形計画問題となっていることを示しているわけである。一方、〔D〕における B^j や b^j に関する細部の技術的情報をすべて中央部で管理・統制することは大規模なシステムになればなる程困難となる可能性が極めて大であり、現実的でない場合が多い。そこで、中央部としてはこの種の問題解決あるいは部分問題に関する決定権は出来る限り、現場の組織あるいはシステム管理者に委譲し、中央部で直接関係のある共通資源の配分や、各サブ組織間の調整や統制に任ずることにより全体の最適化を図るべきであろう。具体的には、部分問題の利益 z^j の和 $\sum_{j=1}^n z^j$ を最大化すべく x^j ($j=1, 2, \dots, n$) を決定すればよい。

ここで、

$$d^j = A^j x^j \quad (j=1, 2, \dots, n) \quad (1)$$

で与えられる m^0 次元ベクトル d^j を定義する。すると〔T〕、〔D〕はさらにつぎのように表される。

$$〔T'〕 \quad \begin{cases} \max & \sum_{j=1}^n z^j \\ \text{s.t.} & \sum_{j=1}^n d^j \leq b^0 \end{cases}$$

$$〔D'〕 \quad \begin{cases} \max & z^j \quad (= C^j x^j) \\ \text{s.t.} & B^j x^j \leq b^j, \quad A^j x^j \leq d^j, \quad x^j \geq 0 \\ & (j=1, 2, \dots, n) \end{cases}$$

3. 問題解法の理論

〔P〕の問題の双対問題〔P Dual〕は双対変数をそれぞれ π^0 、 π^j とすることによりつぎのように与えられる。

$$\min \quad ({}^t\pi^0 b^0 + \sum_{j=1}^n ({}^t\pi^j b^j))$$

〔PDual〕

$$\text{s. t.} \quad \begin{cases} {}^t\pi^0 A^j + {}^t\pi^j B^j \geq C^j \\ \pi^0, \pi^j \geq 0 \\ (j=0, 1, 2, \dots, n) \end{cases}$$

したがって、〔P〕の問題に対応する〔PDual〕が実行可能性を満足すれば、

$${}^t\pi^0 A^j \geq C^j - {}^t\pi^j B^j \quad (2)$$

が成立するはずである。一方双対定理より、最適解においては

$${}^t\pi^0 b^0 + \sum_{j=1}^n ({}^t\pi^j b^j) = \sum C^j x^j$$

が成立する。

つまり、(2)の両辺に右より x^j を乗ずると

$${}^t\pi^0 A^j x^j \geq C^j x^j - {}^t\pi^j B^j x^j$$

となり、上式の両辺につき夫々 $j=1, \dots, n$ の和を考えると

$${}^t\pi^0 (\sum A^j x^j) \geq \sum C^j x^j - \sum ({}^t\pi^j B^j x^j)$$

となる。このことは

$${}^t\pi^0 b^0 + \sum ({}^t\pi^j b^j) \geq \sum C^j x^j \quad (3)$$

であるから、最適解では(2)の双対実行可能性が等号で成立し

$${}^t\pi^0 A^j = C^j - {}^t\pi^j B^j \quad (4)$$

となっていることを示している。

以上の事から、(2)の両辺に x^j を右乗することにより

$$\pi^j x^j \geq (C^j - \pi^0 A^j) x^j \quad (5)$$

となり、何らかの方法で最適解に対応する π^0 が求められたとすると部分問題としてDのかわりにD''を次のように考えれば

$$[D''] \quad \begin{cases} \max & (C^j - \pi^0 A^j) x^j \\ \text{s. t.} & B^j x^j \leq b^j, \quad x^j \geq 0 \end{cases}$$

双対定理より

$$(C^j - \pi^0 A^j) x^j = {}^t\pi^j b^j$$

が成立し、これは正しく全体問題〔P〕の最適条件となっている。

そこで、〔P〕を解くために、まず〔D〕を各部門毎に解く。それらの解が〔T〕の解となれば良い。

〔T〕の目的関数は変数分離型になっている事を考慮すると、〔D〕で求めた解が〔T〕の実行可能解であればそのときは〔P〕の最適解である。そうでない場合、つまり〔D〕の実行可能解であっても〔T〕の実行可能解でない時には〔D〕の解を修正しなければならない。

部分問題〔D〕の各端点を x^{jk} ($j=1, \dots, n, k=1, \dots, jl$) とすると,

$$\left\{ \begin{array}{l} x^j = \sum_{k=1}^{jl} \lambda^{jk} x^{jk} \\ \sum_{k=1}^{jl} \lambda^{jk} = 1, \quad 0 \leq \lambda^{jk} \leq 1 \end{array} \right. \quad (6)$$

として x^j の convex combination で表せる。このことは j 番目の部分問題である解が決定されたすると〔T〕問題では次のようになる。

$$\left\{ \begin{array}{ll} \max & \sum_j C_j \sum_k \lambda^{jk} x^{jk} \\ \text{s. t.} & \sum_j A^j \sum_k \lambda^{jk} x^{jk} \leq b^0, \quad x^{jk} \geq 0 \end{array} \right.$$

すなわち, 〔P〕を λ^{jk} の問題として整理すると

$$〔P'〕 \quad \left\{ \begin{array}{ll} \max & \sum_j \sum_k \lambda^{jk} C^j x^{jk} \\ \text{s. t.} & \sum_j \sum_k \lambda^{jk} A^j x^{jk} \leq b^0 \\ & \sum_k \lambda^{jk} = 1, \quad \lambda^{jk} \geq 0 \end{array} \right.$$

として書き表せることになる。この双対問題〔PDual〕は

$$\left\{ \begin{array}{ll} \min & (\eta^0 b^0 + \sum \eta^j) \\ \text{s. t.} & \left\{ \begin{array}{l} \eta^0 A^j x^{jk} + \eta^j \geq C^j x^{jk} \\ \eta^0 \geq 0 \end{array} \right. \end{array} \right.$$

である。そこで

$$z^{jk} = C^j x^{jk} \quad (7)$$

$$d^{jk} = A^j x^{jk} \quad (8)$$

とすると, 〔P'〕は

$$\begin{array}{ll} \max & \sum_j \sum_k \lambda^{jk} z^{jk} \\ \text{s. t.} & \sum_j \sum_k \lambda^{jk} d^{jk} \leq b^0 \\ & \sum_k \lambda^{jk} = 1, \quad \lambda^{jk} \geq 0 \end{array} \quad (9)$$

となる。つまり部分問題の端点を求めることにより, 〔P'Dual〕の実行可能解となる λ^{jk} が求まればよい。このことは次の δ を定義すると,

$$\delta = C^j x^{jk} - \eta^0 A^j x^{jk} - \eta^j$$

この δ が $\delta \leq 0$ であれば最適解が求められたことを意味している。この場合, η^0, η^j は〔P'〕のシンプレックス乗数を示すものとする。

部分問題の端点集合から中央の問題を解きシンプレックス乗数, つまり価格 η^0 あるいは π^0 を決定する方法は分権システムの解法理論とはなり得ないといえる。すなわち, Dantzig-Wolfe の分割手法はあくまでも中央中心の集権システムの思想といえる。分権システムとしては aggregate された

データに基づく決定過程を追求すべきであり、全体としての最適化のための調整情報の交換を最小限にすべきである。しかも、よしんば、Dantzig - Wolfe 流の解法理論（分割原理）によっても、各部分問題の端点集合を網羅することは極めて非現実的なアプローチといえる。これは z^{jk} と d^{jk} とが決定されたとしたときの λ^{jk} をパラメータとして z^{jk} と d^{jk} との関連性にも言及できることを示唆しているといえよう。

〔P'〕における条件式で $\sum_j \lambda^{jk} = 1$ の代りに、 $\sum_j \lambda^{jk} \leq 1$ としてみると、〔P'Dual〕での条件式は、

$${}^t\eta^0 A^j x^{jk} + \eta^j \geq C^j x^{jk}$$

$$\eta^0 \geq 0, \eta^j \geq 0$$

となるはずである。これはすなわち、

$${}^t\eta^0 A^j x^{jk} \geq C^j x^{jk}$$

を当然満足しているはずである。そこで、

$${}^t\eta^0 A^j x^{jk} = C^j x^{jk}$$

が成立したものと仮定すると

$${}^t\eta^0 d^{jk} = z^{jk}$$

となる。つまり

$$\begin{aligned} \sum_j z^{jk} &= \sum_j {}^t\eta^0 d^{jk} = {}^t\eta^0 \sum_j d^{jk} \\ &= {}^t\eta^0 b^0 \end{aligned} \quad (10)$$

として表せるはずである。ここで、(10) 式の $\sum_j {}^t\eta^0 d^{jk}$ の代りに、 $\sum_j \eta^0 d^{jk}$ を考えてみよう。すなわち、ベクトル η^0 、 d^{jk} の代りにその一要素である実数 η^0 、 d^{jk} に着目することにする。すると

$$\sum_j z^{jk} = \sum_j \eta^0 d^{jk} = \eta^0 \sum_j d^{jk}$$

が成立する。

すなわち、 $\eta^0 = \frac{\sum_j z^{jk}}{\sum_j d^{jk}}$ となる。

これは最適解において $\sum_j z^{jk}$ と $\sum_j d^{jk}$ との比が一定であることを示している。

本来、部分問題毎に求めた η^{oj} に対して、

$$\sum_j z^{jk} = \sum_j \eta^{oj} d^{jk} = \eta^0 \sum_j d^{jk}$$

となれば最適解の条件を満たすものと云える。

いま,

$$\eta^{oj} = \frac{z^{jk}}{d^{jk}}$$

と仮定すると, $\eta^{oj} = \eta^o$ であればよいので

$$\frac{\sum z^{jk}}{\sum d^{jk}} = \frac{z^{jk}}{d^{jk}} = \eta^o$$

が成立する。つまり, d^{jk} と z^{jk} とに関して,

$$\frac{d^{jk}}{\sum d^{jk}} = \frac{z^{jk}}{\sum z^{jk}}$$

であればよい。このことから, 部分問題に対して, 夫々の z^{jk} が求められれば, 中央では

$$\sum_j d^{jk} \leq b$$

であるかを判定し, $\sum d^{jk} > b$ であれば

$$\sum_j \lambda^{jk} d^{jk} = b$$

となるように, λ^{jk} を決定すればよい。ただし, λ^{jk} は

$$\lambda^{jk} = \frac{b z^{jk}}{d^{jk} \sum z^{jk}}$$

とする。

4. 解法アルゴリズムのまとめ

step 1 (j 部分問題)

$$\begin{cases} \max & z^j = C^j x^j \\ \text{s.t.} & B^j x^j \leq b^j, \quad x^j \geq 0 \end{cases}$$

を解く。

step 2 $d_i^j = [A^j]_i x^j$ を計算する。

ただし, $d^j = \{d_i^j\}$, $[A^j]_i$ は A^j の第 i 番目の条件式を表すベクトルとする。

z^j , d_i^j を中央へ送る。

step 3 (中央問題)

$$\sum_j d_i^j \leq b_i$$

であるかどうかを調べる。

(i) $\sum_j d_i^j \leq b_i$ ならば

$$\lambda_i^j = \frac{b_i}{\sum_{j \in I} d_i^j} \quad (j \in I)$$

$$\lambda_i^j = 1 \quad (j \in J)$$

(ii) $\sum_j d_i^j > b_i$ ならば

$$\lambda_i = \frac{b_i z^j}{d_i^j \sum_{j \in I} z^j} \quad (j \in I)$$

$$\lambda_i = 1 \quad (j \in J)$$

ただし,

$$I = \{j \mid d_i^j > 0\}, \quad J = \{j \mid d_i^j \leq 0\}$$

step 4 ($\bar{d}_i$ の決定)

$$\bar{d}_i = \lambda_i d_i - \frac{\sum_{j \in I} d_i^j z^j}{\sum_{j \in I} z^j} \quad (j \in I)$$

$$\bar{d}_i = \lambda_i d_i \quad (j \in J)$$

$\bar{d}_i$ を j 部分へ送る。

step 5 (j 部分問題)

$$\left\{ \begin{array}{ll} \max & z^j = C^j x^j \\ \text{s.t.} & \begin{pmatrix} B^j \\ [A^j]_1 \\ [A^j]_2 \\ \vdots \\ [A^j]_{i-1} \\ [A^j]_i \end{pmatrix} x^j \leq \begin{pmatrix} b^j \\ \bar{d}_1^j \\ \bar{d}_2^j \\ \vdots \\ \bar{d}_{i-1}^j \\ \bar{d}_i^j \end{pmatrix}, \quad x^j \geq 0 \end{array} \right.$$

step 6 $i = m^*$ であるかどうか調べ, $i < m^*$ であれば $i = i + 1$ として step 2, $i = m^*$ であれば $\sum z^j$ を求めて停止する。

5. 計算例

次の線形計画問題を考える。

B^1						$\leq$	b^1
	B^2					$\leq$	b^2
		B^3				$\leq$	b^3
			B^4			$\leq$	b^4
				B^5		$\leq$	b^5
					B^6	$\leq$	b^6

A^1	A^2	A^3	A^4	A^5	A^6	$\leq$	b^0
-------	-------	-------	-------	-------	-------	--------	-------

C^1	C^2	C^3	C^4	C^5	C^6	$\rightarrow$	max
-------	-------	-------	-------	-------	-------	---------------	-----

$$B^1 = \begin{bmatrix} x_1^1 + 9x_2^1 & & + 9x_4^1 + 9x_5^1 + 8x_6^1 + 7x_7^1 \\ 5x_1^1 + 2x_2^1 + 7x_3^1 + 9x_4^1 + 6x_5^1 + 5x_6^1 + & x_7^1 \\ 6x_1^1 + 7x_2^1 + 3x_3^1 & & + 3x_5^1 + 3x_6^1 + 6x_7^1 \\ 9x_1^1 + 5x_2^1 + 4x_3^1 + 9x_4^1 + 3x_5^1 + 2x_6^1 + 5x_7^1 \end{bmatrix}$$

$$B^2 = \begin{bmatrix} 4x_1^2 + 9x_2^2 + 3x_3^2 + 2x_4^2 + 5x_5^2 + 7x_6^2 + 2x_7^2 \\ 9x_1^2 + & x_2^2 & & + & x_4^2 + & x_5^2 + 2x_6^2 + 3x_7^2 \\ 5x_1^2 + 8x_2^2 + 3x_3^2 + & x_4^2 + 4x_5^2 + 5x_6^2 + 9x_7^2 \\ 4x_1^2 + 3x_2^2 + 7x_3^2 & & + 7x_5^2 + 7x_6^2 + 4x_7^2 \end{bmatrix}$$

$$B^3 = \begin{bmatrix} 8x_1^3 + 5x_2^3 + 3x_3^3 + 8x_4^3 + & x_5^3 + 9x_6^3 \\ 9x_1^3 + 9x_2^3 + 8x_3^3 + 7x_4^3 + 5x_5^3 + 2x_6^3 + 7x_7^3 \\ 9x_1^3 + 6x_2^3 + 5x_3^3 + & x_4^3 + 6x_5^3 + 7x_6^3 + 3x_7^3 \\ & 3x_2^3 + 3x_3^3 + 6x_4^3 + 9x_5^3 + 5x_6^3 + 4x_7^3 \end{bmatrix}$$

$$B^4 = \begin{bmatrix} 6x_1^4 + 4x_2^4 & & + 4x_4^4 + 4x_5^4 + 8x_6^4 + 2x_7^4 \\ & 2x_2^4 + 2x_3^4 + 4x_4^4 + 6x_5^4 & & + 6x_7^4 \\ 6x_1^4 + 2x_2^4 + 8x_3^4 + & & + 8x_5^4 + 8x_6^4 + 6x_7^4 \\ 4x_1^4 & & + 4x_3^4 + 4x_4^4 + 8x_5^4 + 2x_6^4 \end{bmatrix}$$

$$B^5 = \begin{bmatrix} 5x_1^5 + 3x_2^5 + 8x_3^5 + & x_4^5 + 9x_5^5 & & + 9x_7^5 \\ 9x_1^5 + 8x_2^5 + 7x_3^5 + 5x_4^5 + 2x_5^5 + 7x_6^5 + 9x_7^5 \\ 6x_1^5 + 5x_2^5 + & x_3^5 + 6x_4^5 + 7x_5^5 + 3x_6^5 \\ 3x_1^5 + 3x_2^5 + 6x_3^5 + 9x_4^5 + 5x_5^5 + 4x_6^5 + 9x_7^5 \end{bmatrix}$$

$$B^6 = \begin{bmatrix} 6x_1^6 + 3x_2^6 + 9x_3^6 + 2x_4^6 + & x_5^6 + 3x_6^6 + 4x_7^6 \\ 7x_1^6 + & x_2^6 + 8x_3^6 + 9x_4^6 + 7x_5^6 + 6x_6^6 + 3x_7^6 \\ 9x_1^6 + 2x_2^6 + & x_3^6 + 3x_4^6 + 4x_5^6 + 7x_6^6 + & x_7^6 \\ 8x_1^6 + 9x_2^6 + 7x_3^6 + 6x_4^6 + 3x_5^6 + 9x_6^6 + 2x_7^6 \end{bmatrix}$$

$$A^1 = \begin{pmatrix} 4x_1^1 + x_2^1 + 5x_3^1 + 6x_4^1 + x_5^1 + 7x_6^1 + 8x_7^1 & & & & & & \\ & x_2^1 + x_3^1 + 2x_4^1 + 3x_5^1 + 5x_6^1 + 8x_7^1 & & & & & \\ 6x_1^1 + 7x_2^1 + 3x_3^1 & & + 3x_5^1 + 3x_6^1 + 6x_7^1 & & & & \\ 8x_1^1 + 5x_2^1 + 3x_3^1 + 8x_4^1 + x_5^1 + 9x_6^1 & & & & & & \end{pmatrix}$$

$$A^2 = \begin{pmatrix} 5x_1^2 + 3x_2^2 + 8x_3^2 + x_4^2 + 9x_5^2 & & + 9x_7^2 & & & & \\ 3x_1^2 + x_2^2 + 4x_3^2 + 5x_4^2 + 9x_5^2 + 4x_6^2 + 3x_7^2 & & & & & & \\ 9x_1^2 + 5x_2^2 + 4x_3^2 + 9x_4^2 + 3x_5^2 + 2x_6^2 + 5x_7^2 & & & & & & \\ 9x_1^2 + 9x_2^2 + 8x_3^2 + 7x_4^2 + 5x_5^2 + 2x_6^2 + 7x_7^2 & & & & & & \end{pmatrix}$$

$$A^3 = \begin{pmatrix} 9x_1^3 + 8x_2^3 + 7x_3^3 + 5x_4^3 + 2x_5^3 + 7x_6^3 + 9x_7^3 & & & & & & \\ 7x_1^3 & & + 7x_3^3 + 7x_4^3 + 4x_5^3 + x_6^3 + 5x_7^3 & & & & \\ 7x_1^3 + 2x_2^3 + 9x_3^3 + x_4^3 & & & + x_6^3 + x_7^3 & & & \\ 9x_1^3 + 6x_2^3 + 5x_3^3 + x_4^3 + 6x_5^3 + 7x_6^3 + 3x_7^3 & & & & & & \end{pmatrix}$$

$$A^4 = \begin{pmatrix} 6x_1^4 + 5x_2^4 + x_3^4 + 6x_4^4 + 7x_5^4 + 3x_6^4 & & & & & & \\ 6x_1^4 + x_2^4 + 7x_3^4 + 8x_4^4 + 5x_5^4 + 3x_6^4 + 8x_7^4 & & & & & & \\ 2x_1^4 + 3x_2^4 + 5x_3^4 + 8x_4^4 + 3x_5^4 + x_6^4 + 4x_7^4 & & & & & & \\ & 3x_2^4 + 3x_3^4 + 6x_4^4 + 9x_5^4 + 5x_6^4 + 4x_7^4 & & & & & \end{pmatrix}$$

$$A^5 = \begin{pmatrix} 3x_1^5 + 3x_2^5 + 6x_3^5 + 9x_4^5 + 5x_5^5 + 4x_6^5 + 9x_7^5 & & & & & & \\ x_1^5 + 9x_2^5 & & + 9x_4^5 + 9x_5^5 + 8x_6^5 + 7x_7^5 & & & & \\ 5x_1^5 + 9x_2^5 + 4x_3^5 + 3x_4^5 + 7x_5^5 & & & + 7x_7^5 & & & \\ 9x_1^5 + 3x_2^5 + 2x_3^5 + 5x_4^5 + 7x_5^5 + 2x_6^5 + 9x_7^5 & & & & & & \end{pmatrix}$$

$$A^6 = \begin{pmatrix} 3x_1^6 + 2x_2^6 + 5x_3^6 + 7x_4^6 + 2x_5^6 + 9x_6^6 + x_7^6 & & & & & & \\ 5x_1^6 + 2x_2^6 + 7x_3^6 + 9x_4^6 + 6x_5^6 + 5x_6^6 + x_7^6 & & & & & & \\ 7x_1^6 + 4x_2^6 + x_3^6 + 5x_4^6 + 6x_5^6 + x_6^6 + 7x_7^6 & & & & & & \\ x_1^6 & & + x_3^6 + x_4^6 + 2x_5^6 + 3x_6^6 + 5x_7^6 & & & & \end{pmatrix}$$

C

C

C

C

C

C

b

b

第

第

$$C^1 = [6x_1^1 + 5x_2^1 + x_3^1 + 6x_4^1 + 7x_5^1 + 3x_6^1]$$

$$C^2 = [3x_1^2 + 3x_2^2 + 6x_3^2 + 9x_4^2 + 5x_5^2 + 4x_6^2 + 9x_7^2]$$

$$C^3 = [3x_1^3 + 2x_2^3 + 5x_3^3 + 7x_4^3 + 2x_5^3 + 9x_6^3 + x_7^3]$$

$$C^4 = [x_2^4 + x_3^4 + 2x_4^4 + 3x_5^4 + 5x_6^4 + 8x_7^4]$$

$$C^5 = [3x_1^5 + x_2^5 + 4x_3^5 + 5x_4^5 + 9x_5^5 + 4x_6^5 + 3x_7^5]$$

$$C^6 = [7x_1^6 + 7x_3^6 + 7x_4^6 + 4x_5^6 + x_6^6 + 5x_7^6]$$

$$b^1 = \begin{pmatrix} 14 \\ 23 \\ 37 \\ 60 \end{pmatrix} \quad b^2 = \begin{pmatrix} 90 \\ 13 \\ 22 \\ 35 \end{pmatrix} \quad b^3 = \begin{pmatrix} 15 \\ 22 \\ 37 \\ 59 \end{pmatrix} \quad b^4 = \begin{pmatrix} 90 \\ 13 \\ 22 \\ 35 \end{pmatrix}$$

$$b^5 = \begin{pmatrix} 60 \\ 70 \\ 13 \\ 20 \end{pmatrix} \quad b^6 = \begin{pmatrix} 15 \\ 22 \\ 37 \\ 59 \end{pmatrix} \quad b^0 = \begin{pmatrix} 160 \\ 170 \\ 240 \\ 280 \end{pmatrix}$$

第 1 部分問題

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	b
x_2	0	1	-0.16	0.84	0.91	0.81	0.79	0.12	-0.02	0	0	1.09
x_1	1	0	1.47	1.47	0.84	0.67	-0.12	-0.05	0.21	0	0	4.16
x_{10}	0	0	-4.65	-14.65	-8.37	-6.74	1.16	-0.54	-1.09	1	0	4.37
x_{11}	0	0	-8.37	-8.37	-9.07	-8.14	2.09	-0.16	-1.77	0	1	17.07
	0	0	7.00	7.00	2.56	5.12	3.26	0.30	1.14	0	0	30.44

第 2 部分問題

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	b
x_8	-10	0	0	0	0	0	-10	1	-1	-1	0	55
x_4	9	1	0	1	1	2	3	0	1	0	0	13
x_3	-1.33	2.33	1	0	1	1	2	0	-0.33	0.33	0	3
x_{11}	13.33	-13.33	0	0	0	0	-10	0	2.33	-2.33	1	14
	70	20	0	0	10	20	30	0	7	2	0	135

第 3 部分問題

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	b
x_6	0.56	0.2	0	0.65	-0.11	1	-0.32	0.12	-0.05	0	0	0.82
x_3	0.99	1.08	1	0.71	0.65	0	0.96	-0.33	0.14	0	0	2.55
x_{10}	0.15	-0.76	0	-7.12	3.49	0	0.46	-0.7	-0.36	1	0	18.55
x_{11}	-5.76	-1.21	0	0.61	7.58	0	2.73	-0.52	-0.18	0	1	47.27
	6.97	5.15	0	2.42	0.30	0	0.91	0.94	0.27	0	0	20.09

第 4 部分問題

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	b
x_8	0	3.33	-6.67	6.67	0	0	0	1	0.67	-1	0	76.67
x_7	0	0.33	0.33	0.67	1	0	1	0	0.17	0	0	2.17
x_6	0.75	0	0.75	-0.5	0.25	1	0	0	-0.13	-0.13	0	1.13
x_{11}	2.5	0	2.5	5	7.5	0	0	0	0.25	-0.25	1	32.75
	3.75	1.67	5.42	0.83	6.25	0	0	0	0.71	0.63	0	22.96

第 5 部分問題

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	b
x_8	-1.08	-2.7	0	-12.7	0	-6.22	-2.43	1	0	-0.38	-1.27	29.68
x_9	8.92	7.3	0	-2.7	0	3.78	-2.47	0	1	0.62	-1.27	52.68
x_5	0.89	0.73	0	0.73	1	0.38	-0.24	0	0	0.16	-0.03	1.57
x_3	-0.24	-0.11	1	0.89	0	0.35	1.7	0	0	-0.14	0.19	2.03
	4.05	5.14	0	5.14	0	0.81	1.62	0	0	0.92	0.51	22.22

第 6 部分問題

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	b
x_7	1.33	0.83	2.17	0	-0.17	0.5	1	0.3	-0.07	0	0	3.03
x_4	0.33	-0.17	0.17	1	0.83	0.5	0	-0.1	0.13	0	0	1.43
x_{10}	6.67	1.67	-1.67	0	1.67	5	0	0	-0.33	1	0	29.67
x_{11}	3.33	8.33	1.67	0	-1.67	5	0	0	-0.67	0	1	44.33
	2	3	5	0	1	5	0	0.8	0.6	0	0	25.2

これらの各部分問題の解に基づいて共通資源の再配分を行った結果、

$$z^1 = 29.20$$

$$z^2 = 132.82$$

$$z^3 = 19.83$$

$$z^4 = 22.96$$

$$z^5 = 22.22$$

$$z^6 = 23.61$$

となって、 $\sum_{j=1}^6 z^j = 250.63$ となる。ところが、これを線形計画問題として分割せずに解いた場合、最適解 $z = 255.77$ となる。

6. 結果の考察

この種の構造化された線形計画問題に対し提案されている代表的解法アルゴリズムとしてDantzig-Wolfe の分割法があるが、この方法と本アルゴリズムとの概略の比較をしてみる。まず、問題の大きさについては

$$\text{DW法} \quad \sum_{j=1}^n (m^j + n^j) + (m^0 + n) \times (m^0 + n + kn)$$

$$\text{本提案法} \quad \sum_{j=1}^n (m^0 + m^j) \times n^j$$

となる。次に情報交換量については

$$\text{DW法} \quad n \times (m^0 + 1) + m^0 + n$$

$$\text{本提案法} \quad 3n$$

となって、いずれも本提案法の方がかなり小さくなっていることがわかる。さらに各種タイプの問題を解かせた結果、

グループ数 (n)	B^j の大きさ	$\sum_{j=1}^n A^j$ の大きさ	最適値	本提案近似解法
2	2×4	2×8	17.21	17.13
3	3×5	3×15	56.65	54.05
5	3×6	3×30	163.12	148.49
6	4×7	4×42	255.77	250.63
10	5×10	5×100	611.33	569.28
10	10×20	10×200	7232.97	6476.73

となって、約5%程度の誤差を持った近似解が求められている。これは中央での資源配分を1回だけ行った結果であることを考慮するとある面では現実的解法アルゴリズムといえる。勿論、くり返

し回数を増やせば精度の向上を計りうるわけであり、計算時間も増加することになるが前述の5%程度の近似解を得るための処理時間は最適値を線形計画法で求める場合の所要時間の1/10程度となっている。

7. 文 献

- 〔1〕 Dantzig, G.B. and Wolfe, P. "The decomposition algorithm for linear programs," *Econometrica*, vol. 29, No. 4, pp. 767-778 (1961)
- 〔2〕 成久洋之, "d-分割分権システムの解析"
岡山理科大学紀要, vol. 16, pp. 15-27 (1981)
- 〔3〕 成久, 岡, "Direct Decompositionアルゴリズムについて," 日本オペレーションズ・リサーチ学会秋季大会アブストラクト集, pp. 51-52 (1983)

プログラム登録表

(様式1)

ライブラリ名	(センタ記入)	作成日付	
プログラム名	ALPHA		
作成者氏名	成久洋之		
形式	(a) コンプリート・プログラム c 関数副プログラム b サブルーチン・副プログラム d その他		
使用機器	a 磁気テープ 入力 () 本, 出力 () 本 b CRTの種類 () (c) 磁気ディスク・ファイル () 個 d その他 ()		
プログラムサイズ	a ソース・プログラム (394) 行 b オブジェクト・プログラム (69.9) KW		
使用言語	a FLAG d COBOL g LISP (b) FORTRAN IV e METASYMBOL h PL/I c FORTRAN 77 f PASCAL i その他 ()		
処理形態	a バッチ処理専用 (c) バッチ・TSS兼用 b TSS専用		
使用条件等	データ入力形式 ① N, M (2 I 10) (N:変数の数, M:条件式の数) ② (0, 0) (2 I 10) → simplex method) (1, 0) (2 I 10) → revised method) ③ データとして $\begin{cases} a_{11}, a_{12}, \dots, a_{1n}, b_1 & (8 F 10) \\ \vdots \\ a_{m1}, a_{m2}, \dots, a_{mn}, b_m \\ c_1, c_2, \dots, c_n \end{cases}$		
(枠内に入りきらない場合は, A4サイズのもので説明を添付してください)			

試験の採点と集計を行うプログラム

教養部教授 浦上 澤之

試験の問題を機械的に採点できる問題と、教師が手作業で採点する問題に分け、マーク・カードを使って採点及び集計を行うプログラムを作成した。機械的に採点する問題には複数個の解を設定でき、各解に異なる得点を付与することができる。また、学生有能力別の評価ができるように、各問題ごとに要求される能力を列挙し、能力別の集計もすることができる。学期、また年度を通して多数回の試験を行うときには、その期間内の集計をすることもできる。

1. 序 論

学生の学習傾向や教師の学生指導方針によっては、定期試験だけでは不十分なことがある。また、1クラスの学生数が多いときには、学生管理は適切に行い難い。学生管理、学習指導に使えると同時に、学生を stimulate する方法として、気軽にできる小テストを取り上げ、テストの結果を処理するプログラムを作成した。

試験のコンピュータによる処理という点、共通1次試験の欠陥を想起しやすい。現在身近にあるコンピュータとその周辺装置を使い、比較的短時間で処理できるためには、出題形式が限定される。そこで、形式を2種類に分ける。1つは機械的に採点できる選択肢方式であるが、問題設定を多様化できるように選択肢の指定を2桁の数によって行う。他は通常の筆記形式で、この解答の採点は教師が自分で行う。物理の試験について言えば、この形式を使って学生の物理的な image, skill, sense の評価も取り入れられる。試験の実施にはマーク・カードを使うので、第1の形式の出題では学生が解答をマーク・カードに記入し、第2の形式では教師が採点結果の評価を記入する。

問題を解くためには、理解力・応用力・表現力・計算力等いろいろな能力を必要とする。問題の種類や出題の仕方によって、これらの能力を調べやすくなることができる。また、解答の分析によって、各能力のレベルを定めることもできるであろう。そこで、本プログラムでは、解答作業に駆使される能力をも評価できるようにした。各問題に複数個の解を用意し、どの解を得るにはどのような能力がどの程度使われるかを定めておく。能力別評価は学生の個人別指導の資料となる。

上述のような趣旨で、フォートランとキー付編成ファイルを使って試験処理のためのプログラムを作成した。以下の節で、そのプログラムについて述べる。

2. マーク・カード

データの入力には80カラムの光学式読取り用のマーク・カードを使う。

2-a. 解答カード

試験実施時に学生が、自分の学生番号から解答までを記入するカードである。教師が手

作業で行う採点の点数を記入する欄も含まれる。

col. 1～6: 6桁の学生番号を記入する。採点原簿等はこの学生番号を行番号とするキー付編成ファイルに作成される。初期にはキー付編成ファイルのフォーマットを規定した原簿オリジナルDVOSR%のみが存在する。第1回の試験を処理すると、原簿オリジナルDVOSR%から作業用採点簿DVAR%と作業用能力簿DVBR%が作成される。解答カードを読み込みながら、学生番号を行番号とする行が挿入されていく。第2回以降の試験で未登録の学生番号があれば、新しく行が挿入される。

col. 7～10: 4桁の暗誦番号を記入する。暗誦番号は初期に学生が自分で定め、以後変更はできない。解答カードを読み込みながら、登録済の学生については学生番号と暗誦番号の照合が行われる。照合の結果一致しないときには、リジェクトファイルDVER%に記録される。

col. 11～12: 2桁に記号化した学科コードを記入する。

col. 13～14: 2桁に記号化したクラスコードを記入する。

col. 15～18: 試験実施の日付を4桁で記入する。2月18日実施ならば0218とする。この欄は解答カードと他のカードとの区別(カードの種類の識別)に使われるので1231より大きい数を記入してはならない。

col. 19～21: 試験の問題用紙に記されている3桁の調査番号を記入する。調査番号によって正解カード、配点カード、能力区分カード、能力レベルカードのデータが呼び出され、処理が行われる。3桁の調査番号は後述する6桁の明細調査番号の1部である。

col. 22～24: 1つのカラムを使って1回分の出欠を記入する。3カラムあるので3回分を記録できる。出欠を次のようなランクに分ける。

出席, 5分以内遅刻, 5分以上遅刻, スポーツ大会等に参加, 病気, 欠席,
試験のみを受講

このようなランキングに適切な付点をし、出席状況の集計を行う。出席状況は、attitudeの評価と合わせてstudy habitの評価に使われる。この欄は教師が記入してもよい。

col. 25～27: 未使用

col. 28～67: この区間の40カラムを2カラムずつに分割し、2カラムを使って1つの問の解答を記入する。問題は20問の出題ができ、選択肢の記号番号は01～99である。未記入欄は00とみなされるので、00は選択肢として使用できない。

col. 68～73: この区間の6カラムを2カラムずつの3欄に分割する。この3欄には、教師が筆記形式の問題の解答を評価して、教師が記入する。使い方は次に述べる正解カード等の指定の仕方によって異なる。

(a). 正解カードの対応する欄が空白のとき、その欄(正解カードのcol. 68～69が空

白のとき、解答カードの col. 68～69 の欄をいう) には、3つの major ability の 1つの評価をパーセント単位で記入する。3つの major ability とは、通常よく話題になる独創力、応用力、分析力等から3つの能力を選びだしたものである。物理学では image, skill, sense を採用した。この評価は採点の合計には入れない。

(b). 正解カードの対応する欄が 01 または 1 のとき、その欄 (正解カードの col. 68～69 が 01 または、col. 68 が空白で col. 69 が 1 のとき、解答カードの col. 68～69 をいう) は、major ability の 1つの評価として使われると同時に、筆記形式の問題の得点として採点の合計に加えられる。採点の満点は通常 100 点が都合がよいので該当する筆記形式の問題はパーセント単位では評価しない。100 点満点内の該当する問題への配点を配点カードの対応する欄に指定する。この配点は major ability としての評価をパーセント単位に換算するために使われる。(配点カードに 20 が指定され、得点が 15 点ならば、major ability としての評価は 75 パーセントである。)

(c). 正解カードの対応する欄が 02 または 2 のとき、その欄には筆記形式の問題の得点を記入する。この得点は採点の合計に加えられる。該筆記形式問題への配点は配点カードの対応する欄に指定できる。

(d). 正解カードの対応する欄が 03 または 3 のとき、その欄には 1つの能力の評価を記入する。1つの能力とは 3つの major ability 以外の能力をいう。能力は能力区分カードの対応する欄に指定する。したがって、筆記形式の問題の解答はその能力の評価のために評価される。評価は、能力レベルカードの対応する欄に指定があるときにはその指定値を満点として、指定がないときには 100 点満点でパーセントを単位として行う。1欄が 2 カラムであるから、1～99 の点数が記入できる。

col. 74～75: 2 カラムを使って assignment 等の得点を教師が記入する。

col. 76～77: 2 カラムを使って attitude 等の評価をパーセント単位で教師が記入する。この評価は出席状況と合わせて study habit の評価に使われる。60 以下はマイナスに評価され、空白は標準とみなされる。

2 - b. 正解カード

問題作成時に、調査番号等と正解を教師が記入する。使用回の欄は年度ごとに修正しなければならない。正解カードと以下に述べるカードは解答カードと一緒に光学式カードリーダーから読み込み、ファイルする。読み込みファイルの名称は後述する使用回が 101 のとき DV 1010 % のように決める。コンピュータで試験の処理を行うと、DV 1010 % 等の読み込みファイル内の正解カードと以下に述べるカードのデータは、調査データファイル DVSD % (キー付編成ファイル) に格納される。これらのファイルにはパスワードが付けられる。

col. 1～6: 6桁の明細調査番号を記入する。調査データファイル DVSD % には、明細調査

番号を行番号とする行が挿入され、正解カードのデータが格納される。明細調査番号のつけ方は次のようである。

col. 1：講義番号を記入する。1種類の講義だけならば空白でよい。

col. 2：講義用テキストの章などの区別に使用し、1～9を記入する。

col. 3～4：問題を分類する番号を記入する。01～99まで問題を作成できる。col. 2～4の3桁が解答カードの調査番号欄に記入される調査番号として使われる。

col. 5：解を分類する番号を記入する。1～9の9種類の解を用意することができる。

0は記入ミスを防止するためには使用しない。多数の問の中の1部の問にだけ、複数解を用意することができる。このときには正解カード以下のカードを解の数に対応するだけ作成するが、それらのカードのcol. 28～67は複数解の欄だけ記入し、他は空白に残す。採点結果の表示ではこの解の分類により、何番の解で得点がいづらかが解るように印刷される。明細調査番号のcol. 1～5は正解カード以下のカードについて共通である。col. 6はカードの分類に使われる。

col. 6：正解カードでは2を記入する。

col. 7～10：前期と後期に分けて第何回目の試験で使ったかという使用回を4桁で記入する。前期は0101～0116を、後期は0201～0216を使う。印刷フォーマットによって16回分に制限されている。この使用回の指定によって採点結果は集計、表示のとき適切な個所にはめられる。したがって、補講で受けた調査番号101の解答カードが、調査番号102のカードデッキ内にあって、読込みファイルDV1020%中に読込まれていても、適切な使用回で処理できる。後述する合格点に達しなくて追試験を受ける学生についても、同一使用回で2回までの追試験が記録され、正しく集計される。使用回は問題の年度初使用時に修正しなければならない。複数解のときには最初のカードの使用回が有効である。

col. 11～14：空白または0000を記入する。

col. 15～18：2000を記入する。2はcol. 6の2と同一であり、正解カードであることをチェックするために使う。2000以上は正解カード等の調査データ用カードに使われ、解答カードの日付(1231より小さい数)と区別される。

col. 19～21：合格点を2桁で記入する。採点結果が合格点以上ならば、結果の表示にOKが書かれる。合格点に達しないとき、学生はさらに2回までトライでき、その採点結果が記録される。追試験を受講した回数が記録され、総合評価に使われる。複数解のときには最初のカードの合格点が有効である。

col. 22～27：空白にする。

col. 28～67：2カラムずつ使って、問の数だけ正解の選択肢の記号番号01～99を記入する。問のない欄は空白に残す。ある問に複数の解があるときには正解カードのcol. 5の番号の異なるカードを作成し、複数の解がある欄のみに記号番号を記入する。

複数解の各正解カードに対応して、配点カード、能力区分カード、能力レベルカードが作成される。処理では解答カードと正解カードの1枚ずつが比較される。合致する欄には対応する配点カードからは得点が、能力レベルカードからは能力区分カードに従って分類された能力に評価が与えられる。比較はcol. 28~29, 30~31, …の順で行われ、最初の正解カードに空白の欄が現われるとそれ以降の欄は比較されない。したがって1枚目の正解カードにはcol. 28~29の欄から正解を記入しはじめ、問のある範囲は空白にしてはならない。通常はcol. 28~29を問1に、col. 30~31を問2に、…というように使用する。処理結果の印刷ではこの問の番号の下に解答、その下に正解ごとに得点(marks)、能力レベルが表示される。

col. 68~73: 2カラムずつの3欄に分割し、解答カードの対応する欄の使用法を指定する。

2カラムの記入の仕方によって、次の4つのように解答カードの欄は扱われる。

- (a). 空白のとき、解答カードの対応する欄の記入は筆記形式問題がmajor abilityの1つの評価に使われたものとして扱われる(パーセント単位)。
- (b). 01または1のとき、major abilityの1つの評価として使われると同時に、問の得点として採点の合計に加えられる。この問への100点内の配分点は配分カードの対応する欄に指定する。この配点はmajor abilityとしての評価をパーセント単位に換算するために使われる。
- (c). 02または2のとき、筆記形式問題の得点として採点の合計に加えられる。この問題への配点は配点カードの対応する欄に指定できる。
- (d). 03または3のとき、major ability以外の能力の評価として扱われる。能力の種類は能力区分カードの対応する欄に指定する。評価の満点は、能力レベルカードの対応する欄に指定があるときにはその指定値を、指定がないときには100点を満点とする。複数解のときには最初のカードで指定する。

2-c. 配点カード

問題作成時に、調査番号等と配点を教師が記入する。

col. 1~6: 6桁の明細調査番号を記入する。読み込みファイルが読まれると、調査データファイルDVSD%には明細調査番号を行番号とする行が挿入され、配点カードのデータが格納される。明細調査番号のつけ方は次のようにする。

col. 1~5: 正解カードのcol. 1~5と同じである。

col. 6: 配点カードでは3を記入する。

col. 7~10: 正解カードでは使用回を記入する欄であるが、その回の試験の加重を記入する。たとえば、第10回目の試験を定期試験のために行い10倍の加重をつけるときには、0010あるいはcol. 7~8は空白、col. 9~10に10と記入する。受験回数の集計には加重が加えられ、平均点、総合評価にも加重が考慮される。追試験は受験回数に数えられない。複数解のときには最初のカードの加重が有効である。

col. 11～14：空白または0000を記入する。

col. 15～18：3000を記入する。3はcol. 6の3と同一であり、カードのチェックに使われる。

col. 19～21：採点するときの満点を記入する。

col. 22～27：空白にする。

col. 28～67：2カラムずつを使って各問に対する配点を記入する。問のない欄は空白に残す。

複数解があるときには正解カードに対応させて、複数の解がある欄のみに採点を記入する。1枚目の配点カードは各問に対する配点であるが、2枚目以降のカードは代替解あるいは副次解に対する採点という性質をもつ。したがって、配点より大きい点数は用いない。

col. 68～73：2カラムずつに分割し、正解カードの対応する欄に01または1が記入されているとき使われる。このとき、解答カードの対応する欄はmajor abilityの評価として使われると同時に、問の得点として採点の合計に加えられる。この問への100点内の配点を配点カードのこの欄に記入する。この配点はmajor abilityとしての評価をパーセント単位に換算するために使われる。

2-d. 能力区分カード

各問を解くためにはいずれかの能力が使われる。このカードにはその使われる能力を正解カード等の欄と対応させて、教師が問題作成時に記入する。使われる能力は問と同時に解によっても異なり、また1つの能力を指定するだけでは不十分なことがある。そこで、各正解カードに3通りの能力カード（区分カードとレベルカード）がつけられる。

col. 1～6：6桁の明細調査番号を記入する。能力区分カードのデータは調査データファイルDVSD%の明細調査番号を行番号とする行に格納される。

col. 1～5：正解カードのcol. 1～5と同じである。

col. 6：最初の能力区分カードでは4，2枚目では6，3枚目では8を記入する。

3枚のカードは3種類の能力に対応し、それぞれに付随する能力レベルカードをつける。解の分類はcol. 5でなされている。

col. 7～14：空白にする。

col. 15～18：col. 6の4，6，8に対応して、最初の能力区分カードでは4000，2枚目では6000，3枚目では8000を記入し、能力区分カードであることのチェックに使われる。

col. 19～27：空白にする。

col. 28～67：2カラムずつを使って各問を解くために使われる能力の指定を行う。能力の指定は1～99で行う。現在は、印刷フォーマットの都合で30種類の能力に分類している。問のない欄は空白に残す。複数解があるときには正解カードに対応させて、複数の解がある欄のみに能力を指定する。問によっては能力は指定しなくてもよい。

col. 68～73：2カラムずつに分割し、正解カードの対応する欄が03または3のときに使われる。このとき筆記形式問題はmajor ability以外の能力、すなわち上述のcol. 28～67

の欄で指定される 30 の能力のうちの 1 つのために評価される。その能力の指定をこの欄で行う。評価は解答カードに教師が記入するが、処理で集計するときには、機械的に採点される問に使われる能力と同じように扱われる。

2-e. 能力レベルカード

問を解いて、ある解答を導くためには、使われる能力があるレベルが必要である。その解を導くために必要な能力のレベルを教師が評定して、原則として 1～10 の範囲で記入する。能力は能力区分カードで指定する。したがって、1 つの正解カードに 3 通りの能力区分カードがあれば、その各々に対応して能力レベルカードも 3 通り作成する。

col. 1～6: 6 桁の明細調査番号を記入する。能力レベルカードのデータは調査データファイル DVSD% の明細調査番号を行番号とする行に格納される。

col. 1～5: 正解カードの col. 1～5 と同じである。

col. 6: 最初の能力区分カードに対応する能力レベルカードでは 5, 2 枚目に対応するカードでは 7, 3 枚目に対応するカードでは 9 を記入する。能力区分カードと能力レベルカードは、1 枚ずつで 1 つの組になり、col. 6 の番号でいえば、4 と 5, 6 と 7, 8 と 9 がそれぞれ組をなす。

col. 7～10: 能力の加重を記入する。最初の配点カードの対応する欄は学期内における試験の加重を指定するが、このカードでは該能力区分カードと能力レベルカードに記されている能力をどの程度重視するかを指定する。したがって、能力の重要度によって能力を分類し、異なる能力カードの組に記すこともできる。

col. 11～14: 空白にする。

col. 15～18: col. 6 の 5, 7, 9 に対応して、最初の能力レベルカードでは 5000, 2 枚目では 7000, 3 枚目では 9000 を記入し、カードのチェックに使われる。

col. 19～27: 空白にする。

col. 28～67: 2 カラムずつを使って、各問を解いて正解カードの対応する欄が指定する解を得るために、能力区分カードの対応する欄が指定する能力をどの程度のレベルで必要とするかを指定する。レベルの指定は 1～10 の範囲であるが、最高 99 まで指定できる。能力区分カードで能力の指定を行っていない欄は空白にする。処理結果の印刷では、第 1 行に問番号、第 2 行に解答カードから解答が表示され、その下に各正解カードから正解とそれによる得点 (marks), さらに各能力カードによる能力評価が ability score として表示される。能力レベルの指定は原則として 1～10 の範囲で行うが、表示ではこれを 10 倍して パーセント 単位にする。能力集計表では、3 つの major ability と 30 の能力のレベルの加重平均 (パーセント単位) と加重度 (各能力が何回問われたかを加重を考慮して示す数) が表示される。合格点に達しないときには補講等で追試験を 2 回まで受けられ、合格するまでは作業用採点簿 DVAR% で集計されないが、作業用能力簿 DVBR% では集計される。

すなわち、合否にかかわらず試験時における能力として集計される。

col. 68～73：2 カラムずつに分割し、正解カードの対応する欄が 03 または 3 のとき使われる。

このとき筆記形式問題は能力区分カードの対応する欄で指定する能力のために評価される。この欄は、その評価の満点を指定するために使う。指定がないときは 100 点を満点とする。指定値が 20 で解答カードの対応する欄に記入された評価が 12 ならば、使われた能力レベルは 6 である。表示するときには col. 28～67 の能力に加重平均され、パーセント単位で処理される。

3. 試験の実施と処理の仕方

問題作成から成績発表までを本プログラムを使って行うときの手順とプログラムの流れについて簡単に述べる。

3-a. 問題作成

選択形式の問題は 20 題まで、筆記形式の問題は 3 題まで出題できる。これらの問題を作成するとき、正解カード、配点カード、能力区分カード、能力レベルカードも同時に作成する。

3-b. 試験の実施

解答カードに必要な暗誦番号は前もって、各自の暗誦番号を決めておくよう指示しておく。学科コード、クラスコードの記入方法を通知して、問題用紙と 80 桁のマーク・カードを各 1 枚ずつ学生に配布し、学生番号と氏名をそれぞれ数字と漢字で問題用紙とマーク・カードに記入させる。試験を開始し、学生はマーク・カードにマークを記していくが、混乱はほとんどないようである。記入は濃いめの鉛筆が望ましい。解答を終了した学生には、解答用紙と解答カードを対にして提出させる。

3-c. 筆記形式問題の評価

筆記形式の問題が高々 3 問あるので、正解カード、配点カード、能力区分カード、能力レベルカードの col. 68～73 の指定に従って評価し、解答カードに記入する。このとき、出席状況、assignment、attitude についても必要に応じて記入する。

3-d. カードの読み込み

解答カードを他のカードと共に光学式カードリーダーから読み込ませる。正解カード、配点カード、能力区分カード、能力レベルカードは一度処理にえば、調査データファイル DVSD%に入っているので以降は読み込ませる必要はない。読み込みファイルの名称は使用回が 101 のとき DV1010%，使用回 101 のための第 1 回目の補講のとき DV1011%のように決める。DV 1020%に使用回 101 や 103 の解答カードや他のカードが含まれていてもかまわない。このファイルは順編成ファイルであり、エディタを使って修正できる。

3-e. コマンド・プロシジュー

コマンド・プロシジュー XVS VY%を位置パラメータなしで起動させると、処理用フォー

トラン・プログラムSVSV%からプログラムテスト用プログラムSVSVYT%がコピーされ、これがコンパイル・リンクスされる。そして、プログラムテスト用読み込みファイルDVTST%を使って処理が行われる。

物理の使用回 101 の試験を処理するときは

!XVSVY%. PHY, 101, S

とキーインする。この処理が使用年度の初回であるならば、原簿オリジナルDVOSR%から作業用採点簿DVAR%と作業用能力簿DVBR%がコピーで作られる。初回でないなら採点簿オリジナルDVAO%, 能力簿オリジナルDVBO%から作業用採点簿DVAR%, 作業用能力簿DVBR%がそれぞれコピーで作られる。次に処理用ロード・モジュールLVSVM%が起動される。

3-f. 処理用プログラム

rejected subject (学生番号と暗誦番号が正しく記されていない学生等) を記録するリジェクトファイルDVER%と、正解カード等の調査用データを格納する調査データファイルDVSD%をオープンして、読み込みファイルから読み込みを開始する。解答カードは、各解答カードに記されている3桁の調査番号によって分けられ、解答ファイルDVES1%, DVES2%, …… に格納される。

読み込みファイルからの読み込み、分けが終わると、各解答ファイルから1解答分のデータが配列変数IEV (33) に読み込まれる。IEV 中の3桁の調査番号から最初の正解カード等の明細調査番号が作られ、調査データファイルDVSD%から最初の正解カード、配点カード、能力区分カード、能力レベルカードがそれぞれ配列変数ISOLUT (33), IPOINT (33), IABIL (33), IABPT (33) に読み込まれる。採点結果は配列変数IMPNT (33), 能力評価の結果はIMABR (33, 2, 3) に入る。IMABR が (33, 2, 3) の配列になっているのは能力区分と能力レベルで2が必要であり、さらに1解答に3種類の能力を指定するには3が必要なためである。したがって、最初の正解カードについて、第1の能力カードが処理されると、第2, 第3, の能力カードが処理される。次に2枚目の正解カードへと進められる。

1枚の解答カードの採点、評価が行われると、IMABR (33, 2, 3) は能力別の配列変数IABR (31, 3) に集計される。31は30の能力区分と、全能力平均のために、3は各能力についての加重平均、加重和と加重度のために必要である。

採点、評価の印刷はクラス、調査番号によって分けながら、各学生について問番号、解答(IEV), 正解(ISOLUT), 点数(IMPNT), 能力評価(IMABR)が表示される。この一覧表は学生の識別に学生番号と暗誦番号を用いたファイルDVK%, 学生番号のみを用いたDVKN%, 暗誦番号のみを用いたDVKP%で印刷できる。クラスの区別はDVK1%, DVK2%のようにする。いずれも順編成ファイルである。

採点、評価の期および年度を通しての集計は配列変数IMPNT (33), IABR (31, 3) からIARS (16, 4, 2) になされる。16の配列は期中の16回の試験のために、4は追試験

2 回を含めて 3 回の受験機会のためと major ability 等の情報を格納するために、2 は前期と後期のために必要である。IARS によって各回の試験で取れている得点や能力も考慮した総合評価など、期および年度を通しての採点を中心にした集計結果が表示できる。この一覧表も学生の識別に学生番号と暗誦番号を用いたファイル DVAL%, 学生番号のみを用いた DVALN%, 暗誦番号のみを用いた DVALP% で印刷できる。

前述した能力別の配列変数 IABR を用いて、能力別の集計結果もファイル DVBL%, DVBLN%, DVBLP% によって印刷できる。

なお、IARS (16, 4, 2), IABR (31, 3) は作業用採点簿 DVAR%, 作業用能力簿 DVBR% から学生番号によって呼びだされ、また集計後は学生番号によってそれらのキー付編成ファイルに格納される。したがって、1 解答の処理が終わったときには作業簿のその学生の行は更新されている。

3-g. 処理結果の表示

処理用プログラムによって 1 つの読み込みファイルが処理されると、リジェクトファイル DVER%, 調査データファイル DVSD%, 採点評価ファイル DVK1%, DVK2%, 採点集計ファイル DVAL%, 能力集計ファイル DVBL% がディスプレイに表示される。処理が望ましいと思われるとき、OK をキーインすると上述の処理結果のファイルの印刷が開始される。

コマンド・プロシージャ XVPRESULT% を使っても処理結果のファイルの印刷はできる。

3-h. 処理の仕上げ

処理用プログラムによって作業用採点簿 DVAR%, 作業用能力簿 DVBR% は更新されたが、採点簿オリジナル DVAO%, 能力簿オリジナル DVBO% は更新されていない。前段階で作成された処理結果の一覧表も点検され、承認されたならば、仕上げとして簿オリジナルの更新を行う。そのためにはコマンド・プロシージャ XVFNSVY% を用いる。これを位置パラメータなしで起動すると、プログラムテスト用の作業用簿 DVART%, DVBR% がプログラムテスト用の簿オリジナル DVAOT%, DVBOT% にそれぞれコピーされる。

物理の試験の仕上げをするときには

!XVFNSVY%. PHY

とキーインする。これによって DVAR%, DVBR% がそれぞれ DVAO%, DVBO% にコピーされる。次の処理を行うときには、この更新された簿オリジナルから作業用簿がコピーで作られる。

3-i. 簿オリジナルの削除

次年度の処理を行うためには、現在の作業用簿と簿オリジナルを削除する。これらのファイルが存在しないときには、コマンド・プロシージャ XVSVY% を起動すると、原簿オリジナル DVOSR% から作業用簿 DVAR%, DVBR% が作成される。

作業用簿と簿オリジナルを削除するためには、コマンド・プロシージャ XVDLSVY% を用いる。これを位置パラメータなしで起動するとプログラムテスト用の DVART%, DVBR%,

DVAOT%, DVBOT%が削除される。物理用のファイルを削除するためには

!XVDFSVY%. PHY

とキーインする。これを行うと、次年度の処理を開始できる。

4. ディスカッション

共通1次試験の形式の問題を採点、集計できる機械が市販されている。1時期に1種類の問題だけを採点するためには、十分に使える機械のようである。教室でカードリーダーやプリンタを装えた端末が使える、いく種類かの問題の解答を即座に採点、分析できれば、学生指導に役立てられるであろう。本稿に述べたプログラムは分析まではできないが、このような試みのために作られたプログラムのうちの1つである。

このプログラムでは期中に16回まで試験を実施できる。これは試験範囲が広いと放棄しがちな最近の学生をstimulateする方法として、小範囲の試験を多数回課することができるようにし、また学力を調べるためよりも学習を促すために、5分程度の復習テストをたびたび実施できるようにするためである。出席状況を記入できる欄もあるので、多人数のクラスでは出欠のチェックに使えば、学生管理の一助にもなる。また、レポート等assignmentの整理にも使える。

能力別の評価を取入れたのは、学生の多様化に対応するためである。本プログラムは物理の講義で使用したが、学力、態度が学生ごとに大きく異なるため、物理の内容だけでなく、表現法、計算法から作図法までに気を配らなければならない。そこで、学生側についても、彼らの表現の仕方、数式の計算方法から図の描き方等まで、広い範囲にわたっての能力が絡んでくる。このような広い範囲にわたる能力の統計をとる方法として、能力別の評価を考えたが、能力の評価法、またそのための問題作成には、試行錯誤を繰返さねばならない。同一問題で能力の評価法を変えて処理を繰返せること、またいくつかの類似問題を使って数年間にわたって調査できることが、コンピュータ処理の強味である。

A Program for Computerized Marking and Tabulation of the
Course Work Tests and Term Examinations

Takuyuki S. URAGAMI

Faculty of Science, Okayama University of Science

Ridai - cho, Okayama 700

(Received)

A program was coded to mark and tabulate the result of the course work tests and term examinations by using the mark cards which are filled in by a student and read by a computer. The questions for a test and examination paper are divided into two categories; one consists of the questions, the solutions for which can be sorted and marked by a computer, and another the questions which ought to be checked and marked by a teacher. Plural solutions can be specified as answers for a question which is to be marked by a computer. The student's various abilities involved in working out a question can be evaluated by listing the abilities necessary for a question.

(様式 1)

(枠内に入りきらない場合は、A4サイズのもので説明を添付してください)

Software Reliability: S-shaped Software Reliability Growth Models and Their Comparisons

Shigeru YAMADA*

*Graduate School of Systems Science, Okayama University
of Science, Ridai-cho 1-1, Okayama 700, Japan

Abstract The S-shaped growth curves of detected software errors can be observed during software testing phase in the software development. The existing software reliability growth models so far proposed for such an error detection process are discussed. The models are applied to actual software error data and the results are compared.

Key Words Software reliability growth model, S-shaped growth, logistic curve, Gompertz curve, nonhomogeneous Poisson process.

1. INTRODUCTION

Software reliability is one of the important issues of today's software development. Several software reliability models have been proposed for the last decade. Among these models, software reliability growth models have been emphasized. The software reliability growth models can describe a software error detection process and estimate software reliability in terms of the number of errors remaining in the software system. In this area, the following models are interesting: Littlewood [5] (Bayesian model), Musa [6] (execution time model), and Goel and Okumoto [1] (nonhomogeneous Poisson process (NHPP) model).

In Japan, several software reliability growth models have been developed for analyzing the software error detection process in which S-shaped growth

curves of detected errors are observed in the testing phase. Deterministic models, i.e., the logistic and the Gompertz growth curves, have been widely used to estimate the error content of software systems [4, 10, 11]. Some computer manufacturers and software houses actually apply the logistic and the Gompertz growth curve models. Besides deterministic models, stochastic software reliability models have been developed by Yamada and Osaki [9, 12-14], and Ohba and Kajiya [3, 8]. Yamada-Osaki's model (the S-shaped software reliability growth model) is based on an NHPP with the S-shaped mean value function as a stochastic process for the software error occurrence phenomenon. Ohba-Kajiya's model (the inflection S-shaped software reliability growth model) has been developed to incorporate the mutual dependency of errors by modifying the logistic growth curve model.

Other related studies include software quality assessment based on test data. The underlying technique is the error seeding method which can evaluate software development tools and techniques. Ido et al. [2] have proposed the capture-recapture method for estimating the residual error content and obtained efficient test strategies by applying the mutation analysis. Ohba [7] has introduced a useful software quality index measured by the product of the test coverage ratio and the test capture ratio. The method integrates Yamada-Osaki's model and the error seeding method.

This paper discusses the software reliability growth models which have been developed and used for the S-shaped growth curves of detected software errors in Japan. The application and comparison of the models are shown for actual software error data.

2. EXISTING MODELS

Based on our experience of test data analysis of real software projects, an observed curve of the cumulative number of detected errors often follows an S-shaped growth curve. Following software reliability growth models have been proposed for analyzing the S-shaped reliability growth. An error is defined as a cause of a failure which is an unacceptable departure of program operation.

The logistic and the Gompertz growth curves have been applied to estimate the error content of implemented software systems for last decade. The growth

curves were originally developed to predict demand trend, economic growth, or future population. The total number of errors detected up to time t in the testing phase is given for the logistic growth curve model as:

$$n_L(t) = \frac{k}{1 + m \cdot \exp(-pt)}, \quad m, p, k > 0, \quad (1)$$

and for the Gompertz growth curve model as:

$$n_G(t) = k \cdot a^{(b^t)}, \quad 0 < a, b < 1, k > 0, \quad (2)$$

where k, p, m, a , and b are constant parameters to be estimated by regression analysis. The parameter k is the initial error content of a software system.

Stochastic software reliability growth models consider the time dependent behavior of software errors. The models are based on the assumption that the total number of errors detected up to time t follows an NHPP, denoted by $\{N(t), t \geq 0\}$. The S-shaped software reliability growth model has been developed by modifying Goel-Okumoto's model [1]. The model is formulated for the error detection process including failure detection (failure occurrence) and error isolation (failure cause analysis) as:

$$\Pr\{N(t) = n\} = \frac{\{M(t)\}^n}{n!} \exp(-M(t)) \quad (n = 0, 1, 2, \dots), \quad (3)$$

$$M(t) = a(1 - (1 + bt) \cdot \exp(-bt)), \quad a, b > 0, \quad (4)$$

where a is the expected number of errors to be eventually detected and b is the error detection rate per error in the steady-state. The inflection S-shaped software reliability growth model is an NHPP model which describes a failure detection phenomenon by modifying the logistic growth curve model. The mean value function is given by:

$$I(t) = \frac{N(1 - \exp(-\phi t))}{(1 + \psi \cdot \exp(-\phi t))}, \quad N, \phi, \psi > 0, \quad (5)$$

where N is the initial error content, ϕ is the error discovery ratio, and ψ is the inflection factor.

3. PARAMETER ESTIMATION

The model parameters in (1), (2), (4), and (5) can be estimated as follows. Suppose that y_k is the observed cumulative number of detected errors up to the k -th observation time t_k ($k=1, 2, \dots, n; 0 < t_1 < t_2 < \dots < t_n$).

For the logistic growth curve model, the following differential equation can be obtained from (1);

$$\left\{ \frac{dn_L(t)}{dt} / n_L(t) \right\} = \frac{p}{k} (k - n_L(t)). \quad (6)$$

Based on linear regression analysis, the estimated relationship, i.e., the regression line, will be of the form

$$Y = c + d \cdot X, \quad (7)$$

where

$$Y = \left(\frac{dn_L(t)}{dt} \right) / n_L(t), \quad X = n_L(t), \quad c = p, \quad d = -p / k. \quad (8)$$

Then, using the estimated regression coefficients $\hat{c}$ and $\hat{d}$ in (7), the estimated model parameters in (1) can be calculated as:

$$\begin{aligned} \hat{p} &= \hat{c}, \\ \hat{k} &= -\hat{c} / \hat{d}, \end{aligned} \quad (9)$$

$$\hat{m} = (n \cdot \hat{k} - \sum_{i=1}^n y_i) / \left(\sum_{i=1}^n y_i \cdot \exp(-\hat{p}t_i) \right),$$

where the differential quotient in (8) needs to be smoothed. In a similar way of method above, the parameters in (2) of the Gompertz growth curve model can be estimated.

The model parameters in the stochastic software reliability growth models can be estimated by the method of maximum likelihood. Then, using the NHPP's properties, the joint probability mass function of $\{N(t_1) = y_1, N(t_2) = y_2, \dots, N(t_n) = y_n\}$, i.e., the likelihood function, for the S-shaped reliability growth model is given by:

$$\begin{aligned}
L &\equiv \Pr \{ N(t_1) = y_1, N(t_2) = y_2, \dots, N(t_n) = y_n \} \\
&= \exp(-M(t_n)) \prod_{k=1}^n \frac{\{M(t_k) - M(t_{k-1})\}^{y_k - y_{k-1}}}{(y_k - y_{k-1})!},
\end{aligned} \tag{10}$$

where $t_0 = 0$ and $y_0 = 0$. Taking the natural logarithm of both sides of (10) and differentiating it with respect to the unknown parameters a and b in (4) yields the simultaneous likelihood equations as:

$$y_n = a(1 - (1 + bt_n) \cdot \exp(-bt_n)), \tag{11}$$

$$a \cdot t_n^2 \cdot \exp(-bt_n) = \sum_{k=1}^n \frac{(y_k - y_{k-1})(t_k^2 \exp(-bt_k) - t_{k-1}^2 \exp(-bt_{k-1}))}{[(1 + bt_{k-1}) \exp(-bt_{k-1}) - (1 + bt_k) \exp(-bt_k)]}, \tag{12}$$

which can be solved numerically. The similar method can give the maximum likelihood estimates of the unknown parameters N , ϕ , and ψ in (5) of the inflection S-shaped software reliability growth model by substituting $I(t)$ for $M(t)$ in (10).

4. APPLICATION AND COMPARISON

Actual software error data will be used to compare the existing software reliability growth models. The data analyzed here were taken in the function test and system test of an input-output execution control program development. The program consists of approximately 4,000 lines of code written in basic assembler language. Twenty three errors were detected in a 13 day period. The observation times of detected errors were measured in net test days. An S-shaped growth curve can be seen in Figure 1.

The estimation results for the existing models, i. e., $\hat{n}_L(t)$, $\hat{n}_g(t)$, $\hat{M}(t)$, and $\hat{I}(t)$, are also shown in Figure 1. We chose two criteria for comparison; the sum of squares of the differences between the actual number of errors y_k and the estimated number of errors $\hat{y}_k$ detected up to the k -th observation time t_k ($k=1, 2, \dots, 13$), i. e.,

$$F = \sum_{k=1}^n (y_k - \hat{y}_k)^2 \quad (\hat{y}_k = \hat{n}_L(t_k), \hat{y}_k = \hat{n}_G(t_k), \hat{y}_k = \hat{M}(t_k), \text{ or } \hat{y}_k = \hat{I}(t_k)), \quad (13)$$

and the accuracy of estimation defined by:

$$E = \left| \frac{A - \alpha}{A} \right|, \quad (14)$$

where A is the true value of the total number of errors in the software system, and α is the estimated expected number of errors to be eventually detected based on each model. Table 1 summarized the results where $A = 26$ in the criterion of E since 3 additional errors were detected after the testing.

Consequently, compared with the existing software reliability growth models based on the criteria, the S-shaped software reliability growth model fits the observed data best.

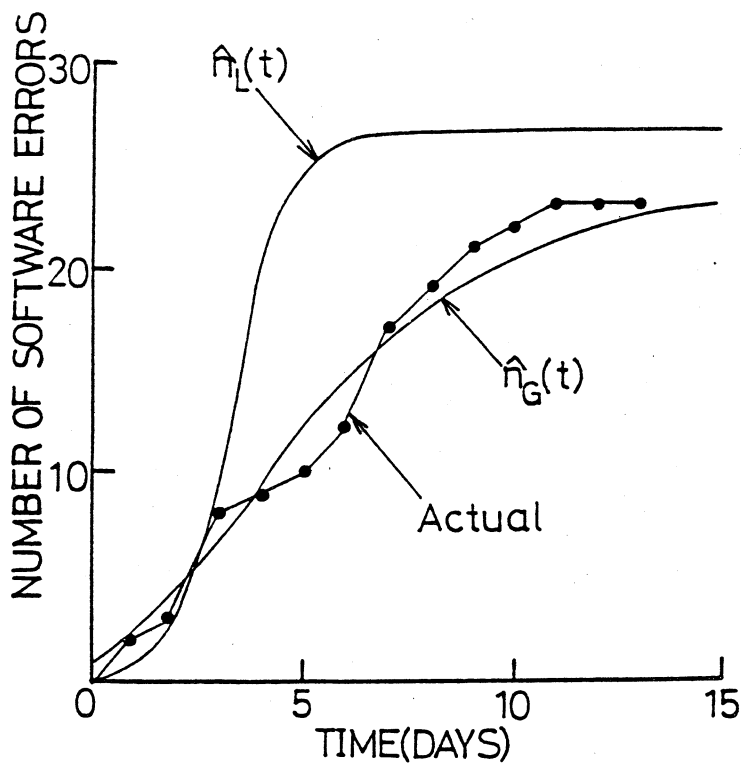
5. CONCLUDING REMARKS

For the S-shaped growth of detected errors in the testing phase, the models for the exponential growth curve of detected errors such as Goel - Okumoto's model do not fit the observed data. In fact the calculated values of criteria for comparison are $F=38.65$ and $E=22.3\%$ for the actual data in the preceding section. The results imply poor estimation. Based on our experiences (e.g., [9]) for many software error data sets, we have proved that it is not reasonable to apply exponential growth models to an observed S-shaped growth.

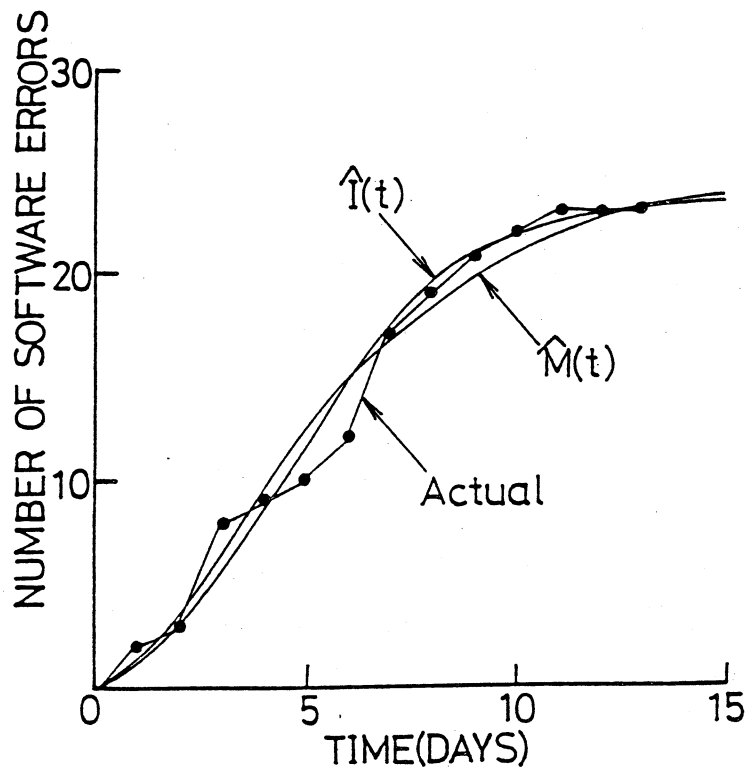
REFERENCES

- [1] A. L. Goel, K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Trans Reliab.*, vol R-28, 1979 Aug, pp 206.-211.
- [2] S. Ido, N. Murata, T. Nakagawa, "On the estimation of hiding bugs by the capture & recapture method and its application," (in Japanese) *IPS Japan Proc. WGSE Meeting*, vol 19, 1981.

- [3] M. Kajiyama, M. Ohba, "An S-shaped software reliability growth model with learning factor of testing," (in Japanese) **IPS Japan Proc. Spring Conf.**, 7C-8, 1982, pp 401-402.
- [4] A. Kanno, **Software Engineering**, (in Japanese) Tokyo:JUSE, 1979.
- [5] B. Littlewood, "Theories of software reliability: How good are they and how can they be improved?" **IEEE Trans. Software Eng.**, vol SE-6, 1980 Sept, pp 489-500.
- [6] J.D. Musa, "The measurement and management of software reliability," **Proc. IEEE**, vol 68, 1980 Sept, pp 1131-1143.
- [7] M. Ohba, "Software quality = test accuracy $\times$ test coverage," **Proc. 6th Int. Conf. Software Eng.**, Tokyo, 1982 Sept, pp 287-293.
- [8] M. Ohba, M. Kajiyama, "Inflection S-shaped software reliability growth model," (in Japanese) **IPS Japan Proc. WGSE Meeting**, vol 28, 1983.
- [9] M. Ohba, S. Yamada, K. Takeda, S. Osaki, "S-shaped software reliability growth curve: How good is it?" **Proc. COMPSAC 1982**, Chicago, 1982 Nov, pp 38-44.
- [10] T. Sakamaki, "Software reliability: Software reliability forecast for quality management," (in Japanese) **IECE Japan Technical Reports on Reliability**, vol R81-8, 1981.
- [11] K. Sakata, "Formulation for predictive methods in software production control: Static prediction and failure rate transition model," (in Japanese) **Trans. IECE Japan**, vol 57-D, 1974 May, pp 277-283.
- [12] S. Yamada, S. Osaki, "Software reliability growth models and their comparisons," (in Japanese) **Trans. IECE Japan**, vol J65-D, 1982 July, pp 906-912.
- [13] S. Yamada, M. Ohba, S. Osaki, "S-shaped reliability growth modeling for software error detection," **IEEE Trans. Reliab.**, vol R-32, 1983 Dec, pp 475-478, 484.
- [14] S. Yamada, S. Osaki, "Reliability growth models for hardware and software systems based on nonhomogeneous Poisson processes: A survey," **Microelectron. Reliab.**, vol 23, 1983, pp 91-112.



(a) Deterministic models



(b) Stochastic models

Figure 1. The estimated number of detected errors for the compared models.

Table 1. Model comparisons .

Model	Criterion for comparison		The estimated expected number of errors to be eventually detected, $\hat{a}$, $\hat{k}$, or $\hat{N}$
	$F = \sum_{k=1}^{13} (y_k - \hat{y}_k)^2$	$E = \left \frac{A - \alpha}{A} \right $	
Logistic growth curve model $n_L(t)$	29.6	13.7 %	22.5
Gompertz growth curve model $n_g(t)$	23.5	8.7 %	23.7
Yamada - Osaki's model $M(t)$	21.5	5.3 %	24.6
Ohba - Kajiyama's model $I(t)$	20.3	10.4 %	23.3

(A = 26)

(様式 1)

- 44 -

最近傍則を用いた手書き漢字

カテゴリー候補の生成

Generation of Candidates of Category of Hand Printed
Chinese Characters by Nearest Neighbor Rule

電子理学科教授 橋 本 礼 治
Reiji HASHIMOTO

1. まえがき

人間のパターン認識の特徴は、現象的にこれまで多数の人々によって言及されているが、要約すると次の二点に絞られる。その一つは、第六感的もしくは直観的、連想的、感覚的などの一連の用語で表わされる、非論理的で現在の計算機で実現が困難とされている機能を用いていることであり、もう一つは、仮説の生成とその検証、すなわち、まず似たものを見出すという第1種の識別と、その差異を見出すという第2種の識別という二段階の処理形態である。この形態は、心理学の研究で、“人間はある段階である種の予測を生成し、続いてその予測の検証が行われる”という記述で早くから指摘されており⁽¹⁾、また、知識工学におけるエキスパートシステムや音声認識におけるヒアセイシステムにおいて工学的に実現されている。

適確な予測が生成されたとき、それをトップダウン的に検証するという方式が非常に有効であることはすでに実証されている。また、検証という性格上、この処理が論理的色彩の強いものであることも肯定されよう。そして、これに関する技法については、これまでに十分な研究成果が得られている。従って問題は、有効な予測方式の開発にかかっているということになる。しかもこの部分は、予測の検証の部分と対比的に、もう一つの特徴として上述した連想的機能が強く介入していると考えられる。

以上のことは、パターン認識の一つである文字認識についてもいえることである。文字認識において連想的と考えられる手法としては、種々の類似度による手法⁽²⁾、ランプレートマッチング法⁽³⁾、⁽⁴⁾相関行列による連想記憶に基づく手法などが挙げられるが、これらはいずれも記憶容量、処理時間という現実的制限を重視しているために、その能力に大きな制限が生じている。一方、人間の脳は過去の経験をすべて記憶しているという説があり、膨大な量の情報を並列的に処理することにより、神秘的とも考えられるような能力を具現している可能性がある。従って、人工系の方も、現在の技術の許す範囲でこの方向に沿って考え直す必要がある。このような考え方で過去に提案されている技法は、おそらく最近傍則(NNR)が唯一のものであろう⁽⁵⁾。そこで、NNRを利用して、簡単な手書き漢字に関する予測生成の実験を行ってみた。この場合、予測は文字のカテゴリーそのものである。なお、現在、手書きの認識において一般的に用いられている手法は、大分類による文字カテゴリー候補の生成と、候補内での詳細認識という二段階方式である。いま考えている方式も形態的には、この中に含まれることになる。しかし乍ら、大分類は、対象カテゴリー数が

多いためにまず手頃な数の候補に絞るとというのが基本思想であり、工学的実現のための便宜的手法である。これに対し、いま考えているNNRによる方法は、人間のように、予測の生成段階で殆んど適確な最終的認識結果を得ようとするもので、予測の検証は、文脈、意味などの点でかなり疑わしいときにのみ行われるようなものを目指しており、予測生成の性格が基本的に異なるものである。

2. シミュレーション

NNRの問題点は、記憶容量と処理時間の大きさにある。しかし、現在、記憶素子の小型、低廉化が急ピッチで進んでいることから考えて、より本質的な問題は処理時間の大きさであると考えられる。処理時間は並列処理などを含むハードウェア構成と、探索領域の分割を含むソフトウェアの両面から短縮が見込める。しかし、ここでは処理時間の短縮化には触れずに、基本的なシミュレーション結果のみを記すことにする。

使用文字データは、電子技術総合研究所作成の手書き漢字データベースETL-8 (B-TYPE) から、表1に示す50カテゴリーの文字を抽出したものである。

表1 使用文字データの категория

右	円	王	下	火	気	金	月	見	口
左	山	子	糸	耳	車	手	出	女	小
上	森	人	水	正	生	石	先	早	足
大	男	中	虫	町	天	田	土	日	入
年	白	文	本	名	木	目	夕	力	林

現データは64×63画素の二値画像であるが、これを二次元FFTにより81次元に圧縮した。表1のカテゴリーは小学校一年次に習得される漢字から選んだもので、もっと複雑な文字に対しては、次元数を増す必要がある。

表2は、一カテゴリー当りの記憶データ数N（従って全記憶データ数は50N）のときに、k個（kは5, 10, 15, および20）の最近傍点の中に正しいカテゴリーが存在する率（正解存在率）Rおよびk個中における異なるカテゴリーの数の平均値 $\bar{k}_e$ を示したものである。なお試行数は1カテゴリー当たり15で、総計750である。

表2 正確存在率のシミュレーション結果 (%)

$k \backslash N$	5		10		15		20	
	R	$\bar{k}_e$	R	$\bar{k}_e$	R	$\bar{k}_e$	R	$\bar{k}_e$
20	96.0	1.7	97.2	2.6	98.1	3.6	98.4	4.7
50	98.5	1.5	99.5	2.0	99.7	2.6	99.7	3.2
70	98.8	1.4	99.6	1.8	99.7	2.2	99.9	2.7
100	99.3	1.3	99.7	1.7	99.9	2.0	99.9	2.4

表3は、 k 個の最近傍点のうちで、最も多いカテゴリーの点数が k_s ($k_s \leq k$) 以上である場合の出現頻度とそのときの正解率を示したものである。この結果から、例えば N が100、 k が20のとき、 k_s が18以上では、正解率は100%となるのが分る。このような場合の出現率は、例えば k_s が18のときは53.7%であり、これは試行回数の約半数の場合に、直観的に正しく文字を認識できることを示している。

表3 近傍点 k 個内の最多カテゴリーの数 k_s ,
出現率 F (%) およびそのときの正解率 R (%)

$N = 20, k = 20$

k_s	11	12	13	14	15	16	17	18	19	20
F	50.4	43.2	35.1	29.2	22.0	14.5	9.9	5.6	2.1	0.4
R	93.4	96.3	98.8	99.5	99.4	99.1	100	100	100	100

$N = 50, k = 20$

k_s	11	12	13	14	15	16	17	18	19	20
F	74.9	70.1	65.9	60.8	54.4	49.6	43.3	36.4	27.9	18.5
R	96.1	97.7	98.8	99.6	99.5	99.7	100	100	100	100

$N = 70, k = 20$

k_s	11	12	13	14	15	16	17	18	19	20
F	81.5	78.0	74.1	69.6	64.0	57.6	52.5	47.3	40.7	30.1
R	96.6	98.6	99.3	99.6	99.6	99.8	100	100	100	100

N = 100, k = 20

ks	11	12	13	14	15	16	17	18	19	20
F	85.6	82.5	78.8	74.7	70.9	66.7	60.8	53.7	46.9	36.7
R	98.8	98.9	99.2	99.6	99.6	99.6	99.8	100	100	100

3. むすび

対象カテゴリー数が少ないために、以上の結果からただちに実用性を述べることはできないが、一応の見通しは得られたといえる。しかしながら、本来の目的である、予測生成段階での最終認識結果の決定は、正解率を大きくしたとき、全認識試行の約半数の割合でしか行われれないという結果は不満足である。この改善は、NNRの方式そのものよりも、それに適用する特徴ベクトルの選択により可能であると思われる。人間における直観的なカテゴリー候補の生成は、文字の論理的構造に陽に依存しない、非論理的なものであると仮定したが、これは構造解析的手法のような極めて論理的な手法を用いないという意味であって、特徴ベクトルに文字の構造を積極的に取り入れるということとは矛盾しない。従って、このような特徴ベクトルの発見が望まれる。

参 考 文 献

- (1) D.E.ルーメルハート(御領訳)：“人間の情報処理”，サイエンス社，PP. 43-101(1979)
- (2) 中田和男：“パターン認識とその応用”，コロナ社(1978)。
- (3) 塩野充，馬場口登，真田英彦，手塚慶一：“方向性マッチングによる常用手書き文字の認識”，信学論(D)，63-D，NO. 5，P. 402(昭55-5)。
- (4) T.KOHONEY：“Correlation Matrix Memories”，IEEE Trans, C-21, P. 353(1972)。
- (5) T.M.Cover and P.E.Hart：“Nearest Neighbor Pattern Classification”，IEEE Trans. IT-13, P. 21(1967)。

謝辞 手書き漢字データベースの使用を許可された電子総研パターン情報部およびこのデータベースの管理やプライベートディスクパックに関し種々御配慮頂いた本学情報処理センターの各位に感謝する。

プログラム登録表

(様式 1)

ライブラリ名	(センタ記入)	作成日付	
プログラム名	NNRDECB		
作成者氏名	橋本礼治		
形式	㊶ コンプリート・プログラム c 関数副プログラム b サブルーチン・副プログラム d その他		
使用機器	a 磁気テープ 入力()本, 出力()本 b CRTの種類() ㊷ 磁気ディスク・ファイル(2)個 d その他()		
プログラムサイズ	㊸ ソース・プログラム(250)行 b オブジェクト・プログラム() KW		
使用言語	a FLAG d COBOL g LISP ㊹ FORTRAN IV e METASYMBOL h PL/I c FORTRAN 77 f PASCAL i その他()		
処理形態	㊺ バッチ処理専用 c バッチ・TSS兼用 b TSS専用		
使用条件等			

(枠内に入りきらない場合は、A4サイズのもので説明を添付してください)

岡山理科大学情報処理センター

岡山市理大町1-1

TEL (0862) 52-3012