

数理計画問題記述簡易言語 (PDL/MP) とその処理システム†

藤 井 実^{††} 斎 藤 博 一^{†††} 横 川 三 津 夫^{††}
佐 藤 治^{††} 安 川 茂^{††}

線形計画問題、混合整数計画問題等の線形の数理計画問題は、計算機の高速化とアルゴリズムの改良により、一部の大規模問題を除いて、ほとんどの問題がわずかな計算時間で解けるようになっている。しかし、多くのユーザは、数理計画問題を簡単に記述できる言語がないため、計算機への入力データ作成に多大な時間を費やしている。このため、筆者らは、科学計算型の極めて簡単な数理計画問題記述言語 PDL/MP を開発した。この言語は、数理計画問題を数式に近い形で記述できるため、言語の習得、問題の記述、修正が極めて短時間でできる。PDL/MP の処理システムは、PDL/MP で記述された問題を解釈し、世界中で幅広く使われている MPS 系ソフトウェアの入力データを自動作成する。本論文では、PDL/MP の概要、処理システム、適用例について記述する。

1. はじめに

線形計画 (LP: Linear Programming) 法¹⁾は、1947 年 Dantzig によってシンプレックス法が開発されて以来、数理計画 (MP: Mathematical Programming) 法¹⁾の中心手法として、多くの産業で利用され、発展してきた。最近では、計算機の高速大容量化、計算手法の改良などによって、従来解くことの難しかった大規模 LP 問題や混合整数計画 (MIP: Mixed Integer Programming) 問題¹⁾なども容易に解くことができるようになっている。これらの数理計画問題は、その大半が MPS (Mathematical Programming System) 系の汎用数理計画法システム²⁾⁻⁴⁾ (以下、MPS 系ソフトと呼ぶ) で解かれている。

しかし、この MPS 系ソフトを利用する場合、ユーザは、数理計画問題における非零要素を一つずつ MPS 形式の入力データ²⁾で記述しなければならない。この入力データ作成には多大な時間を要する。数理計画問題の規模が大きく、技術革新の激しさなどからモデルを頻繁に変える必要の多い今日では、モデルを解くための時間のうち、入力データ作成時間の占める割合が極めて大きくなり、数理計画モデルを処理する上で大きな問題点となっている。

LP モデルでこの問題点を簡単に説明する。一般に

LP 問題は、以下のように定式化される¹⁾。

[LP 問題]

$$\begin{cases} \text{制約条件} & : Ax \geq b & (1.1) \\ & x \geq 0 & (1.2) \\ \text{評価関数(目的関数)} & : c'x \Rightarrow \min & (1.3) \end{cases}$$

この LP モデルの典型的な処理フローを図 1 に示す。実際に利用されている LP モデルは、 A, c などの係数が複雑な式で表されることが多く^{5), 6)}、 $A = \{a_{ij} : i = 1 \sim m, j = 1 \sim n\}$ とすると、各係数 a_{ij} はモデル・パラメータ (ケース・スタディで変化させる可能性のあるデータ) p と基礎データ e の関数として表される。

$$a_{ij} = h_{ij}(p, e) \quad (1.4)$$

このため、LP モデルを使ってケース・スタディを行う場合、パラメータ p の一部を変えると、変えられたパラメータ p^* の関連する (1.4) 式の a_{ij} をすべて再計算し、その a_{ij} の入力データをすべて更新させる作業が必要となる。この作業は、少し問題が大きくなると手作業では不可能に近くなる。

そこで、図 1 の係数計算部分を既存の高級言語でプログラム化したり、 A, b, c データから MPS 形式データを自動作成するプログラムの開発が、多くのユーザで個別に行われてきている。しかし、この方法も汎用化されないと、問題ごとに新しい係数計算プログラムを作成しなければならないという不便さがあった。

これらの不便さから、いくつかの汎用的な数理計画問題記述システムが開発されている。MGRW^{7), 8)}, GAMMA^{9), 10)}, DATAFORM¹¹⁾, UIMP¹²⁾, OMNI^{13), 14)} などである。これらは、マトリックス・ジェネレータ (係数計算, MPS 形式データ作成), レポート・ライタ

† A Readily Available Problem Description Language for Mathematical Programming and Its Processing System by MINORU FUJII (Japan Atomic Energy Research Institute), HIROKAZU SAITO (Nuclear Energy Data Center), MITSUO YOKOKAWA, OSAMU SATO and SHIGERU YASUKAWA (Japan Atomic Energy Research Institute).

†† 日本原子力研究所
††† 原子力データセンター

の両方を記述できる言語とコンパイラ、エグゼキュータなどの処理ソフトから成る大規模な問題記述システムである。

最初に作成された MGRW は、9次元データまで記述できるなど幅広いユーザを対象にした問題記述シ

ステムであるが、記述が制御言語的で使いにくく、一般のプログラマからも敬遠された。

このため、その後 GAMMA, DATAFORM, UI-MP, OMNI などが石油精製モデル等の大規模業務用モデル向きに開発されている。これらは、

- 数値データをテーブル形式で統一的に管理できる、
- モデルの構成要素を種類別にグループ化して扱うことができ、テーブル・データの操作、名前の合成などに豊富な機能をもつ、
- 利用者の必要とする形式にレポート作成ができる、

などの長所をもつ。専従の LP 計算担当者がいて、一度作成したモデルを修正しながら長期間継続的に使用する場合に特に有用である。しかし、数理計画問題を数式主体に扱う科学計算分野の個人ユーザ（モデル作成者）から見ると、これらは、

- データをテーブル（2次元）形式でしか扱えないため、 e_{ijkm} のような3次元以上の配列データは、 $e_{(ij)(kl)m}$ のように2次元化した配列データに定義しなおす必要がある、
 - モデルの構成要素に一つずつ意味づけした名前をつけなければならないなど問題定義に時間がかかる、
 - 機能が豊富なため、言語体系も大きく、問題記述言語の習得に時間がかかる、
 - 使用料が高価である、
- など手軽に使用できない面も多い。

このため、係数計算機能をもたない、MPS 形式データ作成機能のみをもつ簡易な汎用ソフトウェアもいくつか開発されている^{15), 16)}。しかし、これらには係数計算機能がないため、多くのユー

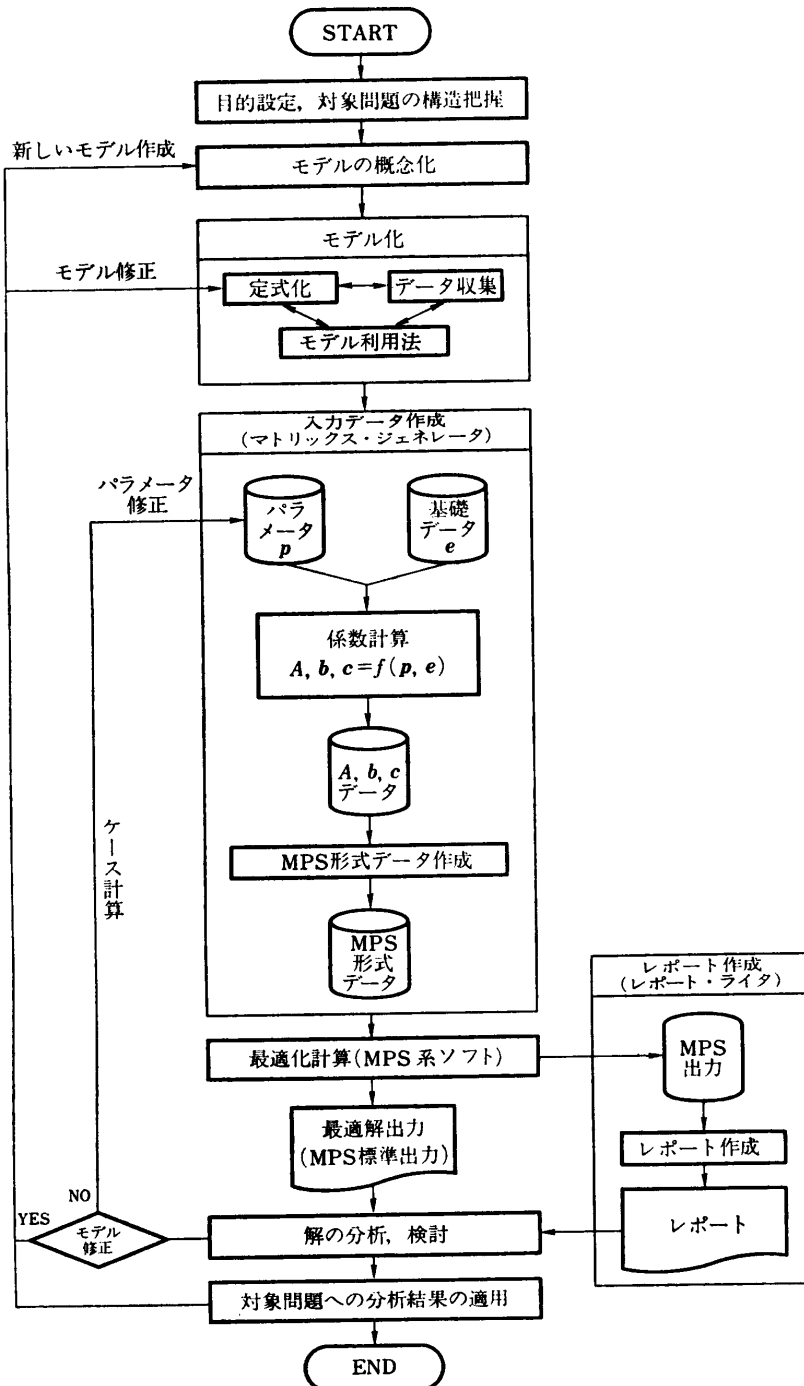


図 1 MPS 系ソフトを利用する場合の LP モデルの処理フロー
Fig. 1 Typical flow chart of LP model for MPS software.

ザは問題ごとに係数計算プログラムと MPS 形式データ作成プログラムを独自に作成，連結して処理しているのが現状である。

これらの問題点から，筆者らは，

“モデル作成者が直観的に理解でき，簡単に数理計画問題が記述できる言語”

の開発を行い，PDL/MP でこれを実現した。

以下，2章で PDL/MP とその処理システムを，3章で PDL/MP の適用例を示す。PDL/MP の言語仕様，使用方法等は，3章の適用例の中で具体的に説明する。

2. 数理計画問題記述言語 PDL/MP とその処理システム

2.1 基本設計

従来の数理計画問題記述システムの問題点と筆者らの必要性から，PDL/MP の設計では以下の目標を立てた。

- ①モデル作成者が直観的に理解でき，簡単に使える言語であること。
- ②数理計画問題の数式をそのまま記述できること。
- ③複雑な計算過程を経て算出されるデータや3次元以上の配列データが簡単に記述できること。
- ④ケース・スタディの実行は，パラメータを変化させるだけで行えること。また，パラメータや基礎データは，問題記述と別な箇所を設定できること。
- ⑤最適解の理解が容易にできること。
- ⑥LP 問題のみでなく，MPS 系ソフトで標準的にサポートされている MIP 問題も記述できること。
- ⑦処理ソフトの開発が短期間にできること。

これらの設計目標から，以下の基本設計がなされた。

- (i) 文の少ない言語とし，インタプリタ（解釈実行）形式の簡易な処理ソフトにする（①，⑦より）。
- (ii) 数式順に問題記述し，係数式に現れる $(u_{ijk} + v_i) * r_{ijk}$ のような表現は， $(U(I, J, K) + V(L)) * R(I, J, K, L)$ と添字をそのまま記述できるようにする（①，②，③より）。
- (iii) パラメータや基礎データから複雑な計算過程を経て算出される数式中のデータは，FORTRAN 等の既存の高級言語で作成できるようにする（③，④より）。
- (iv) MPS 系ソフトの標準出力で使用される変数名（MPS 形式入力データの変数名に対応する）をモデル作成者にわかりやすくする。数式中的変数名が

x_{ijk} ならば， $XIJKL \{X 1111, X 1112, \dots\}$ のように変数に添字を付ける形とする。この機能により，レポート・ライタは作成しない（⑥，⑦より）。

(v) 整数変数も定義できるようにする。

これらの設計のうち，(ii) の数式順の問題記述と (iii) の高級言語の利用については，GAMMA と DATAFORM でも実現されている。ただし，高級言語の利用に関しては，GAMMA と DATAFORM が消極的利用（テーブル操作で記述できない複雑な計算にのみ利用）であるのに対し，本論文ではデータ作成は問題記述言語で行うより既存の高級言語で行った方が簡単であるとする積極的利用の立場をとる。UIMP と OMNI は，データ作成もすべて問題記述言語で行う立場をとっている。(v) の整数変数の定義は，本論文と定義方法は全く異なるが GAMMA の拡張機能¹⁰⁾でも可能である。

2.2 PDL/MP の言語概要

前節の基本設計から，PDL/MP は，以下の 10 種類の文から成る簡単な言語として作成された。

- (1) ¥ PD 文——問題記述の開始を宣言する。
- (2) ¥ SIZE 文——添字や変数の値域定義に使われるスカラ整数とその値を定義する。
- (3) ¥ CLASS 文——添字とその値域を定義する。
- (4) ¥ DATA 文——係数式中に使用される定数データを定義する。
- (5) ¥ FORMAT 文——¥ DATA 文で定義された定数データの入力形式を定義する（FORTRAN の書式なし WRITE 文で定数データをファイルに作成する場合は，この文は省略できる）。
- (6) ¥ RVAR 文——実数変数を定義する。
- (7) ¥ IVAR 文——整数変数を定義する。
- (8) ¥ COND 文——制約式を定義する。
- (9) ¥ COST 文——評価関数を定義する。
- (10) ¥ END 文——問題記述の終了を宣言する。

これらの文は，(1)～(10)の順に定義しなければならない。使用されない文は省略できるが，(1)，(8)，(9)，(10)の文は必須である。コメントは，¥ PD 文の前と ¥ END 文の後，あるいは各文の終了記号「;」の後に記述できる。文の記述は，一行 80 カラムの中に行い，継続する場合は文の終了記号「;」を書かないで次の行に移ればよい。他の問題記述言語と違い，

記述カラムの制約や予約語は一切ない。

2.3 PDL/MP 処理システムの概要

図2に PDL/MP を利用した場合の処理フローを示す。本論文で新しく作成した部分を二重枠で示す。数理計画問題の数式中に使用される定数データ d は、モデル・パラメータ p や基礎データ e と必ずしも同一でない。 d が p と e の関数であることも多い^{6), 6)}。このため、PDL/MP では定式化されている問題記述のみを行い、定数データ d の作成は既存の高級言語で行う。

PDL/MP の処理ソフトは、問題解釈プログラム (PDL/INTPR) と MPS 形式データ作成プログラム

(PDL/MPSIN) から成る。ともに FORTRAN 77 で記述されている。

PDL/INTPR は、問題記述データを一文ずつ主記憶に読み込み、定数データと表1に示す内部表を使って問題記述の解釈、数式(添字)展開、係数計算を行う¹⁷⁾。そして、すべての文を処理した後、MPS 形式データ作成に必要な情報をファイルに出力する。同時に数式展開された出力をプリンタに出す。

PDL/MPSIN は、PDL/INTPR によってファイルに出力された情報から、MPS 形式データと MPS 系ソフトの制御言語プログラムを自動的に作成する。この MPS 形式データと制御言語プログラムを使って、

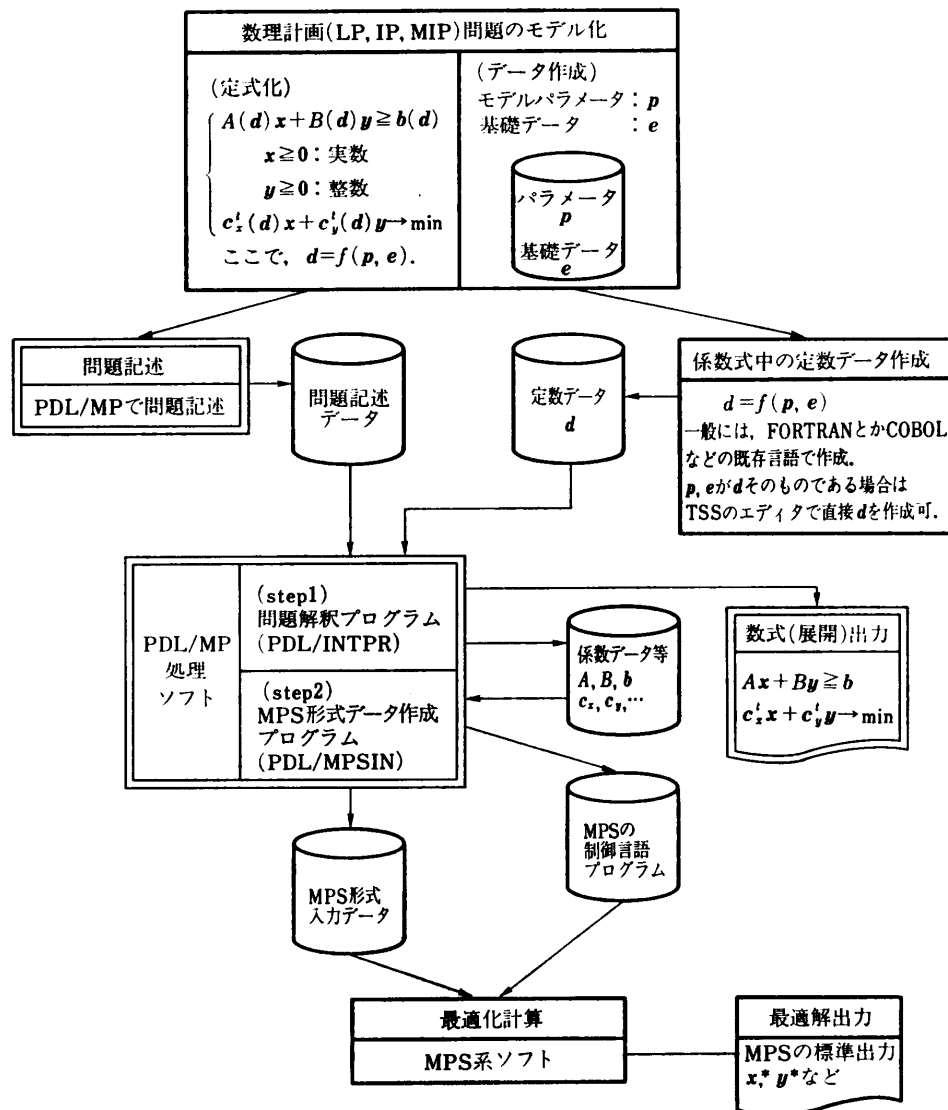


図2 PDL/MP を利用する場合の処理フロー

Fig. 2 Flow chart for PDL/MP.

表 1 PDL/INTPR が使用する内部表
Table 1 Tables used by PDL/INTPR.

No.	表 名	内 容
1	サイズ名表	サイズ名, 値
2	クラス名表	クラス名, 始値, 終値など
3	変数名表	変数名, 次元数, 各次元の添字*, 変数タイプなど
4	データ名表	データ名, 次元数, 各次元の添字*, データ表における先頭番地など
5	制約式名表	制約式名, 次元数, 各次元の添字*, 不等号など
6	データ表	データ値
7	式処理値表	式処理中の値
8	スタック表	処理記号, 左順位関数, 右順位関数
9	式文字列表	式文字列
10	FORMAT 表	FORMAT 文字列
11	制約式用添字処理表	添字名*, 添字処理用ワーク領域
12	変数用添字処理表	添字名*, 添字処理用ワーク領域
13	総和用添字処理表	添字名*, 添字処理用ワーク領域
14	係数行列表	係数行列
15	不等号表	不等号の向き
16	右辺値表	右辺値
17	変数表	変数タイプ, 上・下限値など
18	評価関数表	評価係数と問題のタイプ

*1 添字がクラス名の場合は, クラス名表のポインタが負で入る. 整数, サイズ名の場合は, その値がそのまま入る.

*2 クラス名表のポインタが入る.

MPS 系ソフトは最適化計算を行い, 最適解を出力する.

これらの一連の処理をジョブ制御文で標準的に定義しておけば, ユーザは問題記述と定数データを作成するだけで MPS 系ソフトが簡単に利用できる. MPS 系ソフト以外の数理計画ソフトを使用しているユーザにも, この言語と PDL/INTPR は利用できる. PDL/MPSIN の代わりに, それぞれのソフトが要求する入力データを作成するプログラムを作れば, この処理システムと同等なものが利用可能になる.

3. PDL/MP の適用例

3.1 複合計算機システムの最適設計モデルにおける使用例

3.1.1 複合計算機システムの最適設計モデルの概要

PDL/MP で記述したモデルは, 「ベクトル計算機を含む一地域複合計算機システムの最適設計」⁵⁾ で使用した MIP モデルである. 計算需要, 対象計算機の価格, 性能等を入力すると, 種々の運用制約のもとで, 与えられた計算需要を処理できるコスト最小の最適計算機構成 (システム数と各システムに割り当てる計算

機) と最適計算機構成のもとでの最適ジョブ負荷配分が同時に求められるモデルである. この複合計算機システムの最適設計問題を図 3 に示す.

3.1.2 PDL/MP による問題記述

図 3 の問題を PDL/MP で記述した例を図 4 に示す. ここでは, PDL/MP の言語仕様と使用方法を図 4 に沿って説明する.

① 添字と値域の定義

この問題に表れる添字は, $i, j, k, k1, t$ である. $\forall$ SIZE 文で, 添字と変数の値域定義に使われるスカラー整数 $n_i, n_j, n_k, n_{k1}, n_t, n$ を定義する. $\forall$ CLASS 文で添字 $i, j, k, k1, t$ の値域を定義する. (添字名) = (始値, 終値); の形で定義する. 現在使用している PDL/MP では, 増分値を 1 としている. 添字は 7 次元まで記述できる.

② 変数と定数データの定義

この問題の変数は, 実数変数 x_{k1j} と整数変数 y_{ij} である. これら以外の $r_{k1}, d_k, v_{ik}, e_{ij}, h_t, u_i, w, c_i$ はすべて定数データである.

$\forall$ DATA 文で, 定数データを定義する. 定義された定数データは, ファイル上に, 他的高级言語または TSS のエディタを使って作成しておく必要がある. 図 4 の例では, $d_k \rightarrow D(K), r_{k1} \rightarrow R(K), c_i \rightarrow C(I), u_i \rightarrow U(I), v_{ik} \rightarrow V(I, K), e_{ij} \rightarrow E(I, J, T), h_t \rightarrow H(T), w \rightarrow W$ と対応させ, FORTRAN プログラムでこれらの定数データを作成し, 以下のように書式なし WRITE 文でファイルに書き込んでいる. なお, r_{k1} と $R(K)$ の添字の違いは, ファイルに作成する $\{R(K), K=1 \sim 5\}$ のうち, このモデルではその先頭 4 個 $\{r_{k1}, k1=1 \sim 4\}$ を使うことを意味し, $k1 \leq K$ であるから, 問題はない.

[FORTRAN プログラムによる定数データ作成例]

```
PARAMETER (NI=6, NJ=2, NK=5, NT=2)
DIMENSION D(NK), R(NK), C(NI), U(NI), ...
```

定数データ作成

```
WRITE (6) D, R, C, U, V, E, H, W
STOP
END
```

四角枠で囲まれた定数データ作成の箇所では, 以下のような計算が行われている. パッチ・ジョブクラス k の計算需要 $D(K)$ は, ケース・パラメータの $SXX.l_1l_2l_3l_4$ (XX が年度, l_k が計算需要割合の十分比) から (3.1) 式で算出される.

$$D(K) = D_{59} * \frac{I_k}{10} * (1 + \alpha) * (XX - 59) \quad (3.1)$$

ここで、 D_{59} は昭和 59 年度のバッチ計算需要、 α は計算需要の年増加率である。定式化された問題 (図 3) に陽に記述されない (3.1) 式のような定数データ作成用の式を数理計画問題記述言語で記述すると、問題構造の把握を難しくするし、問題記述言語を大がかりなものにする。このため、PDL/MP では、モデル作成者の使い慣れた高級言語でこれらを記述するようにしている。¥ DATA で定義するデータを書式付き (FORTRAN, PL/I, COBOL, TSS のエディタなど) でファイルに書き込んだ場合は、その書式を ¥ FORMAT 文で以下の例のように定義する必要がある。FORTRAN の FORMAT 文の形式がそのまま使用できる。

```
¥ FORMAT;
(2(5 F 7.2), 11(6 F 7.2),
3 F 7.2);
```

ただし、PDL/INTPR では、この FORMAT 文を FORTRAN の FORMAT 文に置き換えて解釈しているので、PL/I, COBOL などの高級言語や TSS のエディタで定数データをファイルに作る場合は、データを列順に作る必要がある。

変数は、¥ RVAR 文で実数変数 x_{kij} を、¥ IVAR 文で整数変数 y_{ij} を変数名 = (下限値, 上限値); の形で定義する。値域の定義がないと $(0, \infty)$ の標準値が設定される。NAME=SUBS は、MPS 系ソフトの最適解出力において変数名を $XKIJT$ ($X1111, X1112, \dots, X6252$) のように添字を後に付けた形で出力させる指定である。添字の値が 36 (1~9, 0, A~Z の 1 文字で表現できる数) を超えたり、NAME=SUBS の指定がないと、MPS 形式データの変数名は、 $X1, X2, \dots, X120$ と添字表現でなく連番の数字が付けられる。

[複合計算機システムの最適設計モデル]

1. 制約式

制約式名 (制約式数)

- (1) SYSTEM (1 本) : $\sum_{i=1}^{n_i} \sum_{j=1}^{n_j} y_{ij} \leq N$
- (2) BJOB (n_{k1} 本) : $\sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \sum_{t=1}^{n_t} x_{k1, i, j, t} = 1.0 \quad (k1=1, \dots, n_{k1})$
- (3) BDJOB (n_{k1} 本) : $\sum_{i=1}^{n_i} \sum_{j=1}^{n_j} x_{k1, i, j, 1} \geq r_{k1} \quad (k1=1, \dots, n_{k1})$
- (4) TDJOB (1 本) : $\sum_{i=1}^{n_i} x_{5, i, 1, 1} = 1.0$
- (5) CAP ($n_i \times n_j \times n_t$ 本) : $\sum_{k=1}^{n_k} \frac{d_k}{v_{i, k}} x_{k, i, j, t} \leq (1.0 - e_{i, j, t}) h_i u_i y_{ij}$
($i=1, \dots, n_i; j=1, \dots, n_j; t=1, \dots, n_t$)
- (6) $x_{k, i, j, t} \geq 0$ の実数 } ($k=1, \dots, n_k; i=1, \dots, n_i; j=1, \dots, n_j; t=1, \dots, n_t$)
- (7) $0 \leq y_{i, j} \leq N$ の整数 }

2. 評価関数

$$COST = w \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} c_i y_{ij} + \sum_{k=1}^{n_k} \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \sum_{t=1}^{n_t} \frac{d_k x_{k, i, j, t}}{(1.0 - e_{i, j, t}) u_i v_{i, k}} \Rightarrow \text{最小}$$

図 3 複合計算機システムの最適設計問題

Fig. 3 An optimal system design problem for a computer complex.

```
¥PD ;
¥SIZE ;
NI=6 ;
NJ=2 ;
NK=5 ;
NK1=4 ;
NT=2 ;
N=4 ;
¥CLASS ;
(I)=(1,NI) ;
(J)=(1,NJ) ;
(K)=(1,NK) ;
(K1)=(1,NK1) ;
(T)=(1,NT) ;
¥DATA ;
D(K),R(K),C(I),U(I),V(I,K),E(I,J,T),H(T),W ;
¥RVAR ;
X(K,I,J,T),NAME=SUBS ;
¥IVAR ;
Y(I,J)=(0,N),NAME=SUBS ;
¥COND,NAME=SUBS;
@SYSTEM.LE.N;
Y(I,J)=1.0;
@BJOB(K1).EQ.1.0;
X(K1,1,J,T)=1.0;
@BDJOB(K1).GE.R(K1);
X(K1,1,J,1)=1.0;
@TDJOB.EQ.1.0;
X(5,1,2,1)=1.0;
@CAP(I,J,T).LE.0.0;
X(K,I,J,T)=D(K)/V(I,K);
Y(I,J)=-(1.0-E(I,J,T))*H(T)*U(I);
¥COST.N.MIN ;
Y(I,J)=W*C(I);
X(K,I,J,T)=D(K)/((1.0-E(I,J,T))*U(I)*V(I,K));
¥END;
```

図 4 PDL/MP による問題記述 (複合計算機システムの最適設計問題)

Fig. 4 Description of a system design problem for a computer complex by PDL/MP.

③ 制約式の定義

制約式は、¥ COND, NAME=SUBS; の次行から

```

<< RESTRICTIONAL EQUATION >>
( EQ NO.      1 : NAME= SYSTEM )
0.10000E+01*(Y11 ) + 0.10000E+01*(Y12 ) + 0.10000E+01*(Y21 ) + 0.10000E+01*(Y22 ) +
0.10000E+01*(Y31 ) + 0.10000E+01*(Y32 ) + 0.10000E+01*(Y41 ) + 0.10000E+01*(Y42 ) +
LE      4.00000

( EQ NO.      2 : NAME= BJOB1 )
0.10000E+01*(X1111 ) + 0.10000E+01*(X1112 ) + 0.10000E+01*(X1121 ) + 0.10000E+01*(X1122 ) +
0.10000E+01*(X1211 ) + 0.10000E+01*(X1212 ) + 0.10000E+01*(X1221 ) + 0.10000E+01*(X1222 ) +
0.10000E+01*(X1311 ) + 0.10000E+01*(X1312 ) + 0.10000E+01*(X1321 ) + 0.10000E+01*(X1322 ) +
0.10000E+01*(X1411 ) + 0.10000E+01*(X1412 ) + 0.10000E+01*(X1421 ) + 0.10000E+01*(X1422 ) +
0.10000E+01*(X1511 ) + 0.10000E+01*(X1512 ) + 0.10000E+01*(X1521 ) + 0.10000E+01*(X1522 ) +
0.10000E+01*(X1611 ) + 0.10000E+01*(X1612 ) + 0.10000E+01*(X1621 ) + 0.10000E+01*(X1622 ) +
EQ      1.00000

:
:
:

< OBJECTIVE FUNCTION >>
0.51783E+02*(X1111 ) + 0.51783E+02*(X1112 ) + 0.95280E+02*(X1121 ) + 0.95280E+02*(X1122 ) +
0.51783E+02*(X1211 ) + 0.51783E+02*(X1212 ) + 0.95280E+02*(X1221 ) + 0.95280E+02*(X1222 ) +
0.28156E+02*(X1311 ) + 0.28156E+02*(X1312 ) + 0.32276E+02*(X1321 ) + 0.32276E+02*(X1322 ) +
0.51783E+02*(X1411 ) + 0.51783E+02*(X1412 ) + 0.62684E+02*(X1421 ) + 0.62684E+02*(X1422 ) +
0.28156E+02*(X1511 ) + 0.28156E+02*(X1512 ) + 0.31137E+02*(X1521 ) + 0.31137E+02*(X1522 ) +
0.51783E+02*(X1611 ) + 0.51783E+02*(X1612 ) + 0.59550E+02*(X1621 ) + 0.59550E+02*(X1622 ) +
0.25891E+02*(X2111 ) + 0.25891E+02*(X2112 ) + 0.47640E+02*(X2121 ) + 0.47640E+02*(X2122 ) +
0.28768E+02*(X2211 ) + 0.28768E+02*(X2212 ) + 0.52933E+02*(X2221 ) + 0.52933E+02*(X2222 ) +
0.21658E+02*(X2311 ) + 0.21658E+02*(X2312 ) + 0.24828E+02*(X2321 ) + 0.24828E+02*(X2322 ) +
0.39833E+02*(X2411 ) + 0.39833E+02*(X2412 ) + 0.48191E+02*(X2421 ) + 0.48191E+02*(X2422 ) +
0.28156E+02*(X2511 ) + 0.28156E+02*(X2512 ) + 0.31137E+02*(X2521 ) + 0.31137E+02*(X2522 ) +
0.51783E+02*(X2611 ) + 0.51783E+02*(X2612 ) + 0.59550E+02*(X2621 ) + 0.59550E+02*(X2622 ) +
0.25891E+02*(X3111 ) + 0.25891E+02*(X3112 ) + 0.47640E+02*(X3121 ) + 0.47640E+02*(X3122 ) +
0.32364E+02*(X3211 ) + 0.32364E+02*(X3212 ) + 0.59550E+02*(X3221 ) + 0.59550E+02*(X3222 ) +
0.39106E+02*(X3311 ) + 0.39106E+02*(X3312 ) + 0.70390E+02*(X3321 ) + 0.70390E+02*(X3322 ) +
0.71920E+02*(X3411 ) + 0.71920E+02*(X3412 ) + 0.87061E+02*(X3421 ) + 0.87061E+02*(X3422 ) +
0.70390E+02*(X3511 ) + 0.70390E+02*(X3512 ) + 0.77843E+02*(X3521 ) + 0.77843E+02*(X3522 ) +
0.12944E+03*(X3611 ) + 0.12944E+03*(X3612 ) + 0.14887E+03*(X3621 ) + 0.14887E+03*(X3622 ) +
0.14384E+01*(X4111 ) + 0.14384E+01*(X4112 ) + 0.26467E+01*(X4121 ) + 0.26467E+01*(X4122 ) +
0.25891E+01*(X4211 ) + 0.25891E+01*(X4212 ) + 0.46927E+01*(X4221 ) + 0.46927E+01*(X4222 ) +
0.86304E+01*(X4311 ) + 0.86304E+01*(X4312 ) + 0.53794E+01*(X4321 ) + 0.53794E+01*(X4322 ) +
0.14078E+02*(X4511 ) + 0.14078E+02*(X4512 ) + 0.10447E+02*(X4421 ) + 0.10447E+02*(X4422 ) +
0.25891E+02*(X4611 ) + 0.25891E+02*(X4612 ) + 0.15569E+02*(X4521 ) + 0.15569E+02*(X4522 ) +
0.12500E+02*(X5111 ) + 0.12500E+02*(X5112 ) + 0.29775E+02*(X4621 ) + 0.29775E+02*(X4622 ) +
0.67967E+01*(X5311 ) + 0.67967E+01*(X5312 ) + 0.23000E+02*(X5221 ) + 0.23000E+02*(X5222 ) +
0.12500E+02*(X5121 ) + 0.12500E+02*(X5122 ) + 0.23000E+02*(X5221 ) + 0.23000E+02*(X5222 ) +
0.67967E+01*(X5411 ) + 0.67967E+01*(X5412 ) + 0.77913E+01*(X5321 ) + 0.77913E+01*(X5322 ) +
0.12500E+02*(X5421 ) + 0.12500E+02*(X5422 ) + 0.15132E+02*(X5421 ) + 0.15132E+02*(X5422 ) +
0.67967E+01*(X5511 ) + 0.67967E+01*(X5512 ) + 0.75163E+01*(X5521 ) + 0.75163E+01*(X5522 ) +
0.12500E+02*(X5621 ) + 0.12500E+02*(X5622 ) + 0.14375E+02*(X5521 ) + 0.14375E+02*(X5522 ) +
0.20000E+06*(Y11 ) + 0.20000E+06*(Y12 ) + 0.15000E+06*(Y21 ) + 0.15000E+06*(Y22 ) +
0.18400E+06*(Y31 ) + 0.18400E+06*(Y32 ) + 0.11500E+06*(Y41 ) + 0.11500E+06*(Y42 ) +
0.16000E+06*(Y51 ) + 0.16000E+06*(Y52 ) + 0.10000E+06*(Y61 ) + 0.10000E+06*(Y62 )

```

図 5 PDL/INTPR によって展開された数式出力

Fig. 5 Output expressions expanded by PDL/INTPR.

書く. NAME=SUBS は制約式名を添字付きにする指定である. 図3の(5)の制約式を例に説明する. @の後に制約式名 CAP と制約式の展開添字 (I, J, T) を書く. 続けて, 不等号と右辺制約値 (式も可) を書く. この式は, 変数を左辺に記述する標準記法では,

$$\sum_{k=1}^{n_k} \frac{d_k}{v_{ik}} x_{kij} - (1.0 - e_{ij}) h_{ij} u_i y_{ij} \leq 0.0 \quad (3.2)$$

となるので, 不等号 .LE. と右辺値 0.0 を書く. 左辺式は変数ごとに区切り, 変数にかかる係数式を変数名 (添字列) = 係数式; の形で変数の種類だけ書く. (3.2) 式の x_{kij} の係数式は,

$$\sum_{k=1}^{n_k} \frac{d_k}{v_{ik}} x_{kij} = \frac{d_1}{v_{i1}} x_{1ij} + \frac{d_2}{v_{i2}} x_{2ij} + \dots \quad (3.3)$$

NAME	MODEL1		
ROWS			
L SYSTEM			
E BJOB1			
E BJOB2			
.			
.			
L CAP621			
L CAP622			
N OBJECT01			
COLUMNS			
X1111	BJOB1	0.1000E+01	
X1111	BDJOB1	0.1000E+01	
X1111	CAP111	0.4764E+02	
X1111	OBJECT01	0.5178E+02	
X1112	BJOB1	0.1000E+01	
.			
.			
X5622	CAP622	0.1150E+02	
X5622	OBJECT01	0.1250E+02	
INTORG	'MARKER'		'INTORG'
Y11	SYSTEM	0.1000E+01	
Y11	CAP111	-0.9200E+01	
.			
.			
Y62	CAP622	-0.1196E+02	
Y62	OBJECT01	0.1000E+06	
INTEND	'MARKER'		'INTEND'
RHS			
RHS	SYSTEM	0.4000E+01	
RHS	BJOB1	0.1000E+01	
.			
.			
RHS	BDJOB4	0.1000E+00	
RHS	TDJOB	0.1000E+01	
BOUNDS			
UP BOUNDS	Y11	0.4000E+01	
UP BOUNDS	Y12	0.4000E+01	
.			
.			
UP BOUNDS	Y61	0.4000E+01	
UP BOUNDS	Y62	0.4000E+01	
ENDATA			

図6 PDL/MPSIN によって作成された MPS 形式データ
Fig. 6 MPS data generated by PDL/MPSIN.

であるから, $X(K, I, J, T) = D(K)/V(I, K)$; と書く. 同様に y_{ij} の係数式を $Y(I, J) = -(1.0 - E(I, J, T)) * H(T) * U(I)$; と書く.

係数式に総和計算が含まれる場合は, 以下のように記述する. 総和は, 7 重まで記述できる.

$$\sum_{l=1}^{n_l} (f_{lk} * g_{lk}) x_{kij} \Rightarrow X(K, I, J, T) \\ = \langle L : F(L, K) * G(L, K) \rangle;$$

④ 評価関数の定義

¥ COST .N. の後に最大問題の場合は MAX を, 最小問題の場合は MIN を書く. 評価関数の式の定義は, ③の制約式の左辺の定義と同じ方法で行う.

3.1.3 PDL/MP 処理システムにおける出力

図2の PDL/MP 処理システムから出力される情報について説明する. MPS 系ソフトは, MPS/X⁴⁾を使用した.

① 数式展開出力

図5に問題解釈プログラム PDL/INTPR によって展開された数式出力を示す. 図4の問題記述から 34 本の制約式と一つの評価関数が作成される.

② MPS 形式データと MPS/X の制御言語プログラム

図6, 図7に MPS 形式データ作成プログラム PDL/MPSIN が作成する MPS 形式データと MPS/X の制御言語プログラムを示す. 図6の MPS 形式データは, この問題の場合, 合計 503 行が自動的に作成された. 図7の制御言語プログラ

```

MCALL INITIALZ
ADATA = 'MODEL1'
APBNAME = 'MODEL1'
CALL INPUT(NEW,LIST)
AMINMAX = 'MIN'
AOBJ = 'OBJECT01'
ARHS = 'RHS'
ABOUND = 'BOUNDS'
CALL SETUP
CALL OPTIMIZE
CALL SOLUTION
MCALL INIMIX
FTOLIMX = 0.0001
CALL MIXSTART(COST)
CALL MIXFLOW
CALL MIXSTATS
CALL MIXSAVE
STOP
END

```

図7 PDL/MPSIN によって作成された MPS/X 用の制御言語プログラム

Fig. 7 A MPS/X control program generated by PDL/MPSIN.

ラムは、評価関数を MIN 化する MIP 問題用である。図中の四角枠で囲われた箇所のみ、PDL/MPSIN が問題に応じて書き換えている。MAX 化のときには MIN が MAX に、LP 問題のときには MCALL 以下の四角枠内が削除になる。その他は標準名がセットされる。

③ MPS/X の出力

図 8 に MPS/X (MIP) の最適解出力の一部を示す。変数名 X_{kijt} は x_{kijt} に、 Y_{ij} は y_{ij} に対応している。このため、モデル作成者には、例えば $X1311 = 0.320235$ が、スカラジョブ ($k=1$) のうち、計算機 $I - I(i=3)$ のパッチ専用処理システム ($j=1$) で昼間 ($t=1$) に処理される割合である⁵⁾ことが容易に理解できる。

3.1.4 計算機処理時間

PDL/MP は、インタプリタで処理される。このため、ケース・スタディごとに問題解釈、MPS 形式データ作成が行われる。しかし、この節で示した複合計算

機システムの最適設計モデルにおいては、これらの処理に要した演算 (CPU) 時間は、FACOM M-380 計算機で平均 1 秒/ケース程度であり、ほとんど負担にならない。

3.2 長期核燃料サイクルモデル (JALTES-II) への適用

複雑な係数式構造をもつ LP モデル、JALTES-II の記述を PDL/MP で試みた。JALTES-II は、日本原子力研究所で開発された長期核燃料サイクルモデル⁶⁾でエネルギー需給モデルの一種である。PDL/MP で記述した問題は、図 9 に示す JALTES-II のテスト計算用の簡易モデルである。しかし、この簡易モデルは、制約式の数が少ないだけで、評価関数に見られるように係数式の複雑さは、JALTES-II 全体の係数式構造をほぼ説明している。図 10 にこの問題を PDL/MP で記述した例を示す。制約式(2)は、CAP(T) と制約式名を付け、 $X(i, t)$ の係数を $A(i, t)$ としている。PDL/MP では、FORTRAN 言語でデータ作成ができ

SECTION 1 - ROWS				PRIMAL-DUAL OUTPUT				
NUMBER	..NAME..	AT	...ACTIVITY...	SLACK	ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	..DUAL ACTIVITY
1	SYSTEM	UL	4.000000	.	.	NONE	4.000000	.32401166E-10
2	BJ081	EQ	1.000000	.	.	1.000000	1.000000	-37.402030
3	BJ082	EQ	1.000000	.	.	1.000000	1.000000	-28.770000
4	BJ083	EQ	1.000000	.	.	1.000000	1.000000	-32.360000
.	.	.	.	.	.	.	.	.
33	CAP621	UL	.	.	.	NONE	.	.
34	CAP622	UL	.	.	.	NONE	.	.
35	OBJECT01	FR	694089.780087	-694089.780087	.	NONE	NONE	1.000000
SECTION 2 - COLUMNS				PRIMAL-DUAL OUTPUT				
NUMBER	..NAME..	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	..REDUCED COST.	
36	X1111	LL	.	51.780000	.	NONE	27.317970	
37	X1112	LL	.	51.780000	.	NONE	27.317970	
38	X1121	LL	.	95.280000	.	NONE	57.877970	
39	X1122	LL	.	51.780000	.	NONE	27.317970	
40	X1211	LL	.	51.780000	.	NONE	14.377970	
41	X1212	LL	.	51.780000	.	NONE	14.377970	
42	X1221	LL	.	95.280000	.	NONE	57.877970	
43	X1222	LL	.	51.780000	.	NONE	14.377970	
44	X1311	BS	.320235	28.160000	.	NONE	.	
45	X1312	BS	.138203	28.160000	.	NONE	.	
46	X1321	LL	.	32.280000	.	NONE	.000783	
.	.	.	.	.	.	.	.	
153	X5612	LL	.	12.500000	.	NONE	96166.346153	
154	X5621	LL	.	14.370000	.	NONE	5.342385	
155	X5622	LL	.	12.500000	.	NONE	12.500000	
156	Y11	IV	1.000000	199999.999999	.	4.000000	199994.252510	
157	Y12	IV	.	199999.999999	.	4.000000	199996.751418	
158	Y21	IV	1.000000	149999.999999	.	4.000000	149999.999999	
159	Y22	IV	.	149999.999999	.	4.000000	149999.999999	
160	Y31	IV	1.000000	183999.999999	.	4.000000	183992.449625	
161	Y32	IV	.	183999.999999	.	4.000000	183994.144890	
162	Y41	IV	.	114999.999999	.	4.000000	114999.999342	
163	Y42	IV	.	114999.999999	.	4.000000	114999.999628	
164	Y51	IV	.	159999.999999	.	4.000000	159992.449625	
165	Y52	IV	1.000000	159999.999999	.	4.000000	159993.720954	
166	Y61	IV	.	99999.999999	.	4.000000	.	
167	Y62	IV	.	99999.999999	.	4.000000	99999.999999	

図 8 MPS/X による最適解出力

Fig. 8 Output of optimal solutions by MPS/X.

[長期核燃料サイクルモデル JALTES-II の簡易モデル]

1. 制約式

(1) Annual construction

$$XMIN(i, t) \leq X(i, t) \leq XMAX(i, t) \quad (i=1, \dots, n_i; t=1, \dots, n_t)$$

(2) Total demand for electricity or capacity (unit=MWeh/yr)

$$QE(t) \leq \sum_{i=1}^{n_i} h \cdot I(i, t) \cdot \sum_{s=t-t_L(i)+1}^t n(i) \cdot X(i, s) \quad (t=1, \dots, n_t)$$

2. 評価関数

$$FOBJ = \sum_{t=1}^{n_t} \sum_{i=1}^{n_i} [WTU \cdot \{N1(i, t) + N2(i, t) - NR1(i, t) - NR2(i, t) - NR3(i, t)\} \\ + WTH \cdot \{H1(i, t) + H2(i, t) - HR1(i, t) - HR2(i, t) - HR3(i, t)\}] \cdot X(i, t) \\ \Rightarrow \text{最小}$$

ここで,

$$N1(i, t) = (1 + I_E)(1 + I_W)(1 + I_{FV}(i)) \cdot EF\{e1(i, t), e_T(t - t_{E1}(i))\} \cdot k1(1, i, t) \cdot \frac{1}{w(i, t)} \\ \vdots \\ NR2(i, t) = \sum_{s=t+t_{E2}(i)+t_D(i)}^{t+t_{L2}(i)-1} (1 - I_{S2})(1 - I_{R2}) \cdot EF\{e2(i, s - t_{D2}(i)), e_T(s + t_{UC}(i))\} \cdot kr2(1, i, s - t_{D2}(i)) \cdot \frac{d \cdot I(i, s - t_{D2}(i))}{b(i, s - t_{D2}(i))} \\ \vdots \\ HR3(i, t) = (1 - I_{S3})(1 - I_{R3}) \cdot kr3(3, i, t + t_{L3}(i)) \cdot \frac{1}{w(i, t + t_{L3}(i))}$$

である。ただし、 EF は以下の関数である。

$$EF\{e(i, t), e_T(t)\} = \{e(i, t) - e_T(t)\} / \{e_N - e_T(t)\} \quad \text{when } e(i, t) \geq e_N \\ = 0 \quad \text{otherwise}$$

また、 $1 \leq \text{all time parameter} \leq T$ である。

図 9 長期核燃料サイクルモデル JALTES-II の簡易問題

Fig. 9 A simple version of JALTES-II (a long-term strategy model for nuclear fuel cycle system).

```

*PD ;
*SIZE ;
  NI=6 ;
  NT=21 ;
*CLASS ;
  (I)=(1,NI) ;
  (T)=(1,NT) ;
*DATA ;
  WTU,N1(I,T),N2(I,T),NR1(I,T),NR2(I,T),NR3(I,T),
  WTH,H1(I,T),H2(I,T),HR1(I,T),HR2(I,T),HR3(I,T),
  XMIN(I,T),XMAX(I,T),QE(T),A(I,T) ;
*RVAR ;
  X(I,T),NAME=SUBS ;
*COND,NAME=SUBS ;
  @XLOW(I,T).GE.XMIN(I,T);
  X(I,T)=1.0;
  @XHIGH(I,T).LE.XMAX(I,T);
  X(I,T)=1.0;
  @CAP(T).GE.QE(T);
  X(I,T)=A(I,T);
*COST.N.MIN ;
  X(I,T)=WTU*(N1(I,T)+N2(I,T)-NR1(I,T)-NR2(I,T)-NR3(I,T))
  +WTH*(H1(I,T)+H2(I,T)-HR1(I,T)-HR2(I,T)-HR3(I,T));
*END;

```

図 10 PDL/MP による問題記述 (JALTES-II の簡易問題)

Fig. 10 Description of a simple version of JALTES-II by PDL/MP.

るため、 $A(i, t)$ と $N1(i, t), N2(i, t), \dots$ の算出式は FORTRAN で記述する。UIMP, OMNI では、このような複雑な係数式も問題記述言語の機能のみで記述できるようになっているが、例えば評価関数中の $NR2(i, t)$ の計算式を記述するだけでも大変な作業となる。しかし、こうした科学計算的記述は、FORTRAN の

最も得意とする分野であり、関数、条件文なども簡単に記述できる。このため、図 9 で示した複雑な係数式をもつ問題も FORTRAN を使用できるモデル作成者ならば、PDL/MP で簡単に問題記述できる。

4. おわりに

数理計画問題を簡単に記述できる PDL/MP とその処理システムについて記述した。既存の数理計画問題記述システムは、個人が使用するには高価であり、言語の習得に時間がかかる、記述に手数がか

かるなど手軽に利用できない面があった。特にモデル作成に興味を集中する人々にとっては、新しい言語を習得することは非常に負担を感じることであり、簡易な数理計画問題記述言語が望まれていた。

PDL/MP は、FORTRAN ユーザの多い科学計算分野のモデル作成者に最も使いやすい言語として作成さ

れている。3.1 節で示した複合計算機システムの最適設計モデルでは約 100 ケースのケース・スタディを行ったが、PDL/MP なしでは 3~4 か月かかったであろう作業も PDL/MP で問題記述を行ったため 2~3 週間で終了した。図 4 程度の問題記述は、少し慣れれば数分でできるようになる。また、PDL/MP の処理ソフトは、筆者らが技術計算用簡易言語のインタプリタの作成経験をもっていたため、約 5 人月（設計に 2 人月、コーディングとデバッグなどに 3 人月）で実用化できた簡単なものである。既存の問題記述言語と比べると、PDL/MP は、10% の作業量で 90% の満足を得ようとしたもので、まだ不備な点も多い。

今後の課題としては、添字集合の定義方法の多様化、簡易なレポート作成機能の追加などが残されている。

謝辞 最後に、本言語作成にあたり有益なコメントをいただいた三菱総合研究所の藤石克朗主任研究員、日本 IBM 社の富真一主任システムズ・エンジニアの両氏に感謝いたします。

参 考 文 献

- 1) 前田活郎ほか：コンピュータマネジメント・サイエンスハンドブック，オーム社，東京（1971）。
- 2) 中原啓一ほか：新版システムズエンジニアハンドブック，オーム社，東京（1982）。
- 3) (株)日本 IBM: IBM Mathematical Programming System Extended/370, Program Reference Manual SH 19-1095-3 (1979. 12).
- 4) (株)富士通: MPS/X 解説書，富士通マニュアル，70 AR-0500-2 (1978. 5).
- 5) 藤井 実，横川三津夫：ベクトル計算機を含む一地域複合計算機システムの最適設計，情報処理学会論文誌，Vol. 26, No. 5, pp. 807-814 (1985. 9).
- 6) 佐藤 治，安川 茂：原子力開発長期戦略のためのシステム解析モデル：JALTES-II，日本原子力研究所，JAERI-M 85-129 (1985. 9).
- 7) (株)日本 IBM: Matrix Generator and Report Writer (MGRW), Program Reference Manual SH 19-5014-0 (1972. 10).
- 8) (株)富士通: MPS/X-MGRG 使用手引書，富士通マニュアル，78 SP-7620-1 (1984. 3).
- 9) (株)三菱電機: MELCOM GAMMA 3.2 説明書，三菱電機マニュアル，NM-SR 00-32 A(53 A0) (1975).
- 10) Sanders, R. L. and Smith, M. G.: A Description of Bonner & Moore's GAMMA SYSTEM, Bonner & Moore Management Science, Hous-ton.
- 11) Creegan, J. B., Jr.: DATAFORM: A Model Management System, KETRON, Inc., Arlington (1985).
- 12) Ellison, E. F. D. and Mitra, G.: UIMP: User Interface for Mathematical Programming, *ACM Trans. Math. Soft.*, Vol. 8, No. 3, pp. 229-255 (Sep. 1982).
- 13) 木下順隆ほか：数理計画モデル記述システム OMNI, ソフトウェア流通，No. 9, pp. 98-103 (1981. 10).
- 14) 井上 斉：石油精製 LP と OMNI の利用，*bit*, Vol. 13, No. 13, pp. 1646-1653 (1981. 11).
- 15) (株)CCS: RELACS データ・マニュアル
- 16) (株)日本 IBM: MPSX/370 MODEL MAINTENANCE UTILITY.
- 17) 中田育男：コンパイラの技法，日科技連，東京（1972）。

(昭和 61 年 2 月 17 日受付)
(昭和 61 年 6 月 18 日採録)



藤井 実 (正会員)

昭和 25 年生。昭和 48 年広島大学工学部経営工学科卒業。昭和 50 年同大学院修士課程修了。同年日本原子力研究所入所。現在，同研究所計算センターにて計算機システムの解析評価，原子炉解体評価等の研究に従事。日本経営工学会会員。



斎藤 博一

昭和 29 年生。昭和 52 年弘前大学理学部物理学科卒業。昭和 55 年同大学院修士課程修了。同年日本ソフトウェア開発(株)入社。現在，原子力データセンター (NEDAC) にて，主として日本原子力研究所の核融合関係のコード開発に従事。

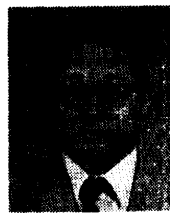


横川三津夫

昭和 35 年生。昭和 57 年筑波大学第一学群自然科学類卒業。昭和 59 年同大学院修士課程修了。同年日本原子力研究所入所。現在，同研究所計算センターに勤務。数値解析や実験データ処理の研究に従事。

**佐藤 治**

昭和 23 年生。昭和 46 年名古屋大学工学部原子核工学科卒業。昭和 48 年同大学院修士課程修了。昭和 51 年日本原子力研究所入所。現在、同研究所核エネルギーシステム研究室にて長期核エネルギーシステムの解析研究に従事。「Energy After the Eighties」(共著)。日本原子力学会会員。

**安川 茂**

昭和 11 年生。昭和 34 年東北大学工学部通信工学科卒業。同年日本原子力研究所入所。動力炉開発部、国産動力炉開発室、原子炉設計部を経て現在動力炉開発・安全性研究管理部核エネルギーシステム研究室長。工学博士。「原子力発電」、「エネルギー用語辞典」、「物理学用語辞典」、「Energy After the Eighties」(ともに共著)。日本原子力学会会員。
