

LP 8000 リニア・プログラミング・システムの開発

Development of LP 8000 Linear Programming System

山 本 晃 司* 小 国 力* 倉 橋 昭*
Teruji Yamamoto Tsutomu Oguni Akira Kurahashi

要 旨

LP 8000 は、HITAC 8300/8400/8500 のために開発された線形計画計算のための総合プログラムであり、大規模な LP 計算が可能である本格的な LP コードとしては国産で最初に開発されたものである。本文では、LP 8000 の特長、アルゴリズム、プログラムの構成、実験結果についてその概要を紹介する。

1. 緒 言

LP (Linear Programming: 線形計画法) は周知のように、OR (Operations Research) の分野で広く用いられている問題解決のための一手法である。値を決定しなければならない変数が、線形な制約条件を構成するという前提のもとに、線形で表わされた目標関数を最大、または最小にすること(最適化)が LP の対象である。

1940 年代の後葉に George Dantzig 氏によって最初に開発されて以来、LP はその技術の応用面、および、取り扱う問題の大きさの面において長足の進歩をとげてきた。より複雑で大規模な LP モデルの作成は、電子計算機そのものの大形化、高速化と相呼応して、大規模な LP モデルを扱える LP コードの開発をうながしてきた。日立製作所においては、さきに LP 4010 (HITAC 4010 のための LP コード)⁽¹⁾、および LP 5020 (HITAC 5020 のための LP コード)⁽²⁾ を開発してこれらの需要にこたえてきたが、さらにこれらを機能面、および取扱可能な問題の大きさの面において強化し、HITAC 8300/8400/8500 のための線形計画計算の総合プログラムとして LP 8000 を開発し、すでに稼働の実績をあげている。

2. 特 長

LP 8000 は、すべてアセンブラ言語で書かれており、必要なすべての入出力操作に対してフィジカル・レベルの FCP (File Control Processor)⁽³⁾ を用いており、Tape Operating System (TOS), Tape Disk Operating System (TDOS)⁽⁴⁾、および Disk Operating System (DOS)⁽⁵⁾ の下で稼働する。

次に、LP 8000 の持つおもな特長を列挙し説明する。

(1) ジョブの指定法は、アジェンダム・カード形式である。

LP 8000 に対するジョブの指示は、大きなマクロ命令であるアジェンダムをパンチした一連のアジェンダム・カードによって行なわれる。すなわち、アジェンダム・カードが読み込まれると、それに対応するオーバーレイ・ルーチンがオーバーレイ・エリアにロードされ、ジョブが実行される。

(2) 採用しているアルゴリズムはシンプレクス 2 段階法であり、最大こう配法を用いている。またフェイズ I では、コンポジットアルゴリズムを用いている。

(3) コンポジット・プライシングを選択できる。

標準機能ではコンポジット・プライシングを行なわないが、CONTROLS アジェンダム・カード⁽⁶⁾により、コンポジット・プライシングを行なうように指示できる。

(4) サブオプティマイゼーションを選択できる。

標準機能ではサブオプティマイゼーションを行なわないが、CONTROLS アジェンダム・カードにより、サブオプティマイゼーションを行なうように指示できる。

(5) 初期基底の決定には、クラッシングの手法を用いている。

データの入力時にユーザーが初期基底を指定しない場合に限って、クラッシングの手法を用いて初期基底を決定する。

(6) 入力データ形式は LP/90 と互換性がある。

(7) 入力データの行名、列名の定義にカナ文字を用いることができる。

(8) 解かれる問題の式の数に応じてダイナミック・バッファリングを行なっている。

LP 8000 のロード時に LP 8000 の使用するエリアの大きさを指定できる。LP 8000 は指定されたエリアの大きさから、コモン・ワーク・エリア、および入出力バッファ・エリアとして使用可能なエリアを計算し、当該問題の計算のためにじゅうぶんかどうかを調べる。次に、そのエリアから、解かれる問題の式の数に応じてコモン・ワーク・エリアを割り当て、残りのエリアを四つの入出力バッファ・エリアに分割する。したがって、LP 8000 の使用するエリアが大きいほど、それだけインコアで解ける問題が大きくなると同時に計算速度も上昇する。

(9) 計算はすべて倍精度で行なわれ、 E_{1a} レコードは $1\frac{1}{2}$ 精度でたくわえている。

計算精度を上げるためには、終始一貫して倍精度で通すのが望ましいが、 E_{1a} レコードを倍精度でたくわえると、コア内に駐在できるレコード数が減少し計算速度が落ちる。そこで、計算精度も計算速度も落とさないように、このような対策がとられている。

(10) ランの中断、再開の機能が完備している。

大規模な LP 問題のランは一般に長時間要するものであるし、また所要時間を予測することも困難である。そこでデバイス・エラーや時間切れなどの理由でランの続行が不可能になった場合に対処するために、ランの中断、再開の機能が完備していることが必要になる。

LP 8000 は 2 種類のランの中断、再開機能を備えている。

(a) エグゼクティブのチェックポイント・リスタート機能による方法。

これは、指定された時点(コンソール・タイプライタからのリクエストがあった時点)で CKPT マクロ⁽³⁾ を発して、その時点のコアの内容を磁気テープ記憶装置にダンプすることによって、リスタート情報を保存する方法である。したがって、リスタートもエグゼクティブのリスタート機能による。

(b) LP 8000 のゲットオフ・リスタート機能による方法。

これは、指定された時点(GETOFF アジェンダム・カード⁽⁶⁾ が読み込まれた時点、または CONTROLS アジェンダム・カードで指定された回数ごと)で、そのときの基底と LP 8000 コミュニケーション・リージョンの内容の一部を磁気テープ記憶装置にダンプすることによって、リスタート情報を保存する方法

* 日立製作所神奈川工場

表1 取扱可能な問題の大きさ

主記憶装置の大きさ	式の数 (目的関数行も含む)	変数の数 (スラック変数は含まない)
65 KB	400 強	32,767
131 KB	1,200 強	32,767
262 KB	2,800 強	32,767

である。したがって、リスタートも LP 8000 のリスタート機能による。

(11) コンソール・タイプライタからの割り込みによって、ジョブの進行をコントロールできる。

(12) アウトプット・アナライザ LPTRAN⁽⁷⁾ が用意されている。

(13) FORTRAN 言語で書かれたプログラムで、入力データの作成、および解の解析ができる。

解の解析のためには、アウトプット・アナライザ LPTRAN が用意されているが、ユーザー独自のアウトプット・アナライザを FORTRAN 言語を用いて作成することがきる。

(14) 主記憶装置の大きさと、取扱可能な問題の最大の大きさとの関係は表1に示すとおりである。

3. 機器構成

(1) 最小機器構成

中央処理装置	1台	65 KB (8300/8400/8500)
コンソール・タイプライタ	1台	
カード読取機	1台	
ラインプリンタ	1台	
磁気テープ記憶装置	5台	

ただし、システム・デバイスは含まれていない。

(2) 追加機器

カードせん孔機	1台
---------	----

4. アルゴリズム⁽⁸⁾

ここでは、LP 8000 において用いているアルゴリズムについて説明する。以後の説明で用いる記号を定義しておく。

m : 制約行の数(目的関数行も含む)

n : 変数の数(スラック変数も含む)

X_j : j 番目の変数

a_{ij} : 入力時の制約行列の i 行 j 列要素。ただし、 a_{oj} は入力時の目的関数行の係数を表す。

α_{ij} : 現在の制約行列の i 行 j 列要素。ただし、 α_{oj} は現在の目的関数行の係数を表す。

d_j : リデュースト・コスト

b_i : 入力時の i 行の右辺要素

β_i : 現在の i 行の右辺要素

J : 基底変数番号の集合

π_{ij} : 現在の基底の逆行列の i 行 j 列要素

s : ピボット列

r : ピボット行

$\phi(i)$: 現在の基底解で β_i を解とする変数番号。 $\phi(i)=0$ は人為変数であることを示す。

4.1 最適化(最小化)のアルゴリズム

4.1.1 ピボット列の選択

$$d_s = \min_j \{d_j < 0\}$$

$$\text{ただし, } d_j = \sum_{i=0}^m \pi_i a_{ij}$$

ここで、フェイズ I では、

$$\pi_i = f\pi_{oi} + \pi_i^c \quad (f \geq 0)$$

$$\pi_i^c = \sum_{h \in M} \pi_{ih} - \sum_{h \in P} \pi_{ih}$$

$$M = \{h / \beta_h < 0\} \cap (h \geq 1)\}$$

$$P = \{h / (\beta_h > 0) \cap (\phi(h) = 0) \cap (h \geq 1)\}$$

フェイズ II では、

$$\pi_i = \pi_{oi}$$

[注] $A = \{i / P(i)\}$ は、 $P(i)$ が真であるような i の集合を表わす。 $\cap$ は論理積を表わす。

4.1.2 ピボット行の選択

(1) $R_1 = \{i / (\phi(i) = 0) \cap (\alpha_{is} \neq 0) \cap (\beta_i = 0) \cap (i \geq 1)\}$ が空集合でないとき、

$$\alpha_{rs} = \max_{i \in R_1} \{|\alpha_{is}|\}$$

(2) R_1 が空集合であり、かつ、

$$R_2 = \{i / (\phi(i) \neq 0) \cap (\beta_i = 0) \cap (\alpha_{is} > 0) \cap (i \geq 1)\}$$

$$\text{が空集合でないとき, } \alpha_{rs} = \max_{i \in R_2} \{\alpha_{is}\}$$

(3) R_1, R_2 がともに空集合であり、かつ、

$$R_3 = \{i / (\alpha_{is} \cdot \beta_i > 0) \cap (i \geq 1)\} \text{ が空集合でないとき,}$$

$$\theta_r = \min_{i \in R_3} \{\beta_i / \alpha_{is}\}$$

[注] $\min$ や $\max$ を求める場合に‘対’が生ずれば、番号の若いものを選ぶ。

4.1.3 最大こう配法

これは、ピボット列の選択に際して、もし可能ならば二つの列を候補として選択し、その両者の中で目的関数値をより望ましい方向に変化させるほうをピボット列として選択しようとする方法である。

まず、ピボット列の候補として次のように列 s_1, s_2 を選ぶ。

$$d_{s_1} = \min_j \{d_j < 0\}$$

$$d_{s_2} = \min_{j \neq s_1} \{d_j < 0\}$$

$$\text{次に, } \alpha_{is_1} = \sum_{h=1}^m \pi_{ih} a_{hs_1}$$

$$\alpha_{is_2} = \sum_{h=1}^m \pi_{ih} a_{hs_2}$$

を計算し、 α_{is_1} と β_i, α_{is_2} と β_i を用いてピボット行選択の方法に従って、おののおの、 $\theta_{r_1}, \theta_{r_2}$ を求める。目的関数値は、 X_{s_1} を基底変数に選んだ場合には $\theta_{r_1} d_{s_1}$ だけ、 X_{s_2} を基底変数に選んだ場合には $\theta_{r_2} d_{s_2}$ だけ、おののおの望ましい方向に変化する。そこで、より望ましいほうを選ぶために、

$$\theta_r d_s = \max \{|\theta_{r_1} d_{s_1}|, |\theta_{r_2} d_{s_2}|\}$$

によりピボット要素を決定する。

ただし、LP 8000 では、人為変数をできるだけ早い時点で基底から追い出すほうが望ましいことにかんがみ、人為変数の基底からの追い出しを上基準よりも優先させている。たとえば、 X_{s_2} を基底に導入することによって人為変数が追い出される(すなわち、 $\phi(r_2)=0$) ならば、 $|\theta_{r_1} d_{s_1}| > |\theta_{r_2} d_{s_2}|$ であっても X_{s_2} を基底変数に選ぶ。

4.1.4 サブオプティマイゼーション

4.1.3 で述べた方法により、二つの列 s_1, s_2 が候補として選ばれ、最終的にピボット要素として $\alpha_{r_1 s_1}$ が選ばれたとする。 $\alpha_{r_1 s_1}$ による掃き出しの結果得られた列 s_2 の要素を $\bar{\alpha}_{is_2}$ とし、フェイズ I では、

$$\bar{d}_{s_2} = f\bar{\alpha}_{os_2} + \sum_{i \in M} \bar{\alpha}_{is_2} - \sum_{i \in P} \bar{\alpha}_{is_2} \quad (f \geq 0)$$

を、フェイズ II では

$$\bar{d}_{s_2} = \bar{\alpha}_{os_2}$$

を計算し、 $\bar{d}_{s_2} < 0$ ならば、列 s_2 を次の反復計算のピボット列として選択する。ただし、LP 8000 ではフェーズ I での条件をきびしくし、

$$f\bar{\alpha}_{os_2} \leq 0, \quad \text{かつ},$$

$$\sum_{i \in M} \bar{\alpha}_{is_2} - \sum_{i \in P} \bar{\alpha}_{is_2}$$

の場合に限り、 s_2 をピボット列として選択する。

4.2 右辺パラメトリック LP のアルゴリズム

右辺パラメトリック LP のために用いられるチェンジ・ベクトルの現在の要素を r_i とする。

4.2.1 ピボット行の選択

- (1) $R_1 = \{i / (r_i \neq 0) \cap (\phi(i) = 0) \cap (i \geq 1)\}$ が空集合でないとき

$$r_r = \max_{i \in R_1} \{|r_i|\}$$
- (2) R_1 が空集合であり、かつ、
 $R_2 = \{i / (r_i > 0) \cap (\beta_i = 0) \cap (i \geq 1)\}$ が空集合でないとき、

$$r_r = \max_{i \in R_2} \{r_i\}$$
- (3) R_1, R_2 がともに空集合であり、かつ、
 $R_3 = \{i / (r_i > 0) \cap (i \geq 1)\}$ が空集合でないとき、

$$\theta_r = \min_{i \in R_3} \{\beta_i / r_i\}$$

4.2.2 ピボット列の選択

- (1) $\phi(r) = 0$ で、かつ、 $K_1 = \{j / (j \in J) \cap (d_j = 0) \cap (\alpha_{rj} \neq 0)\}$ が空集合でないとき、

$$\alpha_{rs} = \max_{j \in K_1} \{|\alpha_{rj}|\}$$
- (2) $\phi(r) = 0$ で、かつ、 K_1 が空集合であり、かつ、
 $K_2 = \{j / (j \in J) \cap (d_j \neq 0) \cap (\alpha_{rj} \neq 0)\}$ が空集合でないとき、

$$\phi_s = \min_{j \in K_2} \{d_j / |\alpha_{rj}|\}$$
- (3) $\phi(r) \neq 0$ で、かつ、 $K_3 = \{j / (j \in J) \cap (d_j = 0) \cap (\alpha_{rj} > 0)\}$ が空集合でないとき、

$$\alpha_{rs} = \max_{j \in K_3} \{\alpha_{rj}\}$$
- (4) $\phi(r) \neq 0$ で、かつ、 K_3 が空集合であり、かつ、
 $K_4 = \{j / (j \in J) \cap (d_j \neq 0) \cap (\alpha_{rj} < 0)\}$ が空集合でないとき、

$$\phi_s = \min_{j \in K_4} \{-(d_j / \alpha_{rj})\}$$

4.3 コスト・パラメトリック LP のアルゴリズム

コスト・パラメトリック LP のために用いられるチェンジ・ロー・ベクトルの現在の要素を α_{oj} とする。

4.3.1 ピボット列の選択

- (1) $K_1 = \{j / (j \in J) \cap (\alpha_{oj} = 0) \cap (\alpha_{oj} < 0)\}$ が空集合でないとき

$$\alpha_{os} = \max_{j \in K_1} \{|\alpha_{oj}|\}$$
- (2) K_1 が空集合であり、かつ、
 $K_2 = \{j / (j \in J) \cap (\alpha_{oj} \neq 0) \cap (\alpha_{oj} < 0)\}$ が空集合でないとき

$$\phi_s = \min_{j \in K_2} \{-(\alpha_{oj} / \alpha_{oj})\}$$

4.3.2 ピボット行の選択

最適化のアルゴリズムのピボット行の選択法と同じである。

5. まるめ誤差の集積の検出と処理

大規模な LP 計算に耐え得る LP コードの開発における重要な問題点の一つは、まるめ誤差の集積の検出とその処理である。ここでは、LP 8000 が用いている処理方式を簡単に説明する。

最適化の過程においてピボット列 s_1, s_2 を選択したのちに、変数 X_{s_1}, X_{s_2} のいずれかが基底変数であるかどうかを調べる。基底変数に対する d_j の値はゼロでなければならぬにもかかわらず、変数 X_{s_1} 、あるいは X_{s_2} が基底変数の候補として選択されたということ

TOS/TDOS/DOS

LP 8000 コントロール・エリア	約 4 K バイト
LP 8000 コミュニケーション・リージョン	
LP 8000 オーバレイ・エリア	約 8 K バイト
LP 8000 コモンワーク・エリア	
LP 8000 入出力バッファ・エリア	可 変

図 1 LP 8000 のメモリ配置

はまるめ誤差の集積の結果にほかならない。同じ考え方に基づいてパラメトリック LP の場合には、基底変数に対する d_j がゼロかどうかを調べる。ゼロでなければまるめ誤差の集積の結果と判断する。このような事態が発生した場合には、LP 8000 はユーザーの指定に従って、再逆化を行なってまるめ誤差の集積を緩和するか、次のアジェンダム・カードを読み取るか、あるいはランを終了させる。

また、実行可能解が得られたのちにも 1 回の反復計算が終了するごとに解の実行可能性を調べ、実行不可能になれば自動的にフェイズ I の過程にはいる。

LP 8000 では、まるめ誤差の処理の目的と計算のスピード化を図るために、次の 5 種類のゼロトレランスを用いている。すなわち、LP 計算に伴う諸計算において、ある数の絶対値がゼロトレランスに等しいかそれよりも小さければ、その数をゼロと見なして処理する。

TOLDJ: d_j のゼロトレランス。標準値は 10^{-5} 。

TOLPV: ピボット要素のゼロトレランス。標準値は 10^{-8} 。

TOLELM: E_{ia} ベクトル要素のゼロトレランス。標準値は 10^{-10} 。

TOLRHS: 右辺要素のゼロトレランス。標準値は 10^{-5} 。

TOLZE: その他の場合のゼロトレランス。標準値は 10^{-10} 。トレランスとしてはその他、解の最大許容誤差である TOLSOL (標準値は 10^{-5}) と、ピボット要素選択のためのトレランス (標準値は 10^{-5}) がある。これらのトレランスは普通、標準値にセットされているが、トレランス・アジェンダム・カード⁽⁶⁾を用いて変更することができる。

6. プログラムの構造

LP 8000 は、1 個のルート・セグメントと 20 個のオーバレイ・セグメントから成っている。そのメモリ配置は図 1 に示すとおりである。ここで各エリアについて簡単に説明する。

6.1 LP 8000 コントロール・エリア

これは、LP 8000 コントロール・ルーチン (ルート・セグメント) が常駐するエリアである。各オーバレイ・ルーチンはコントロール・ルーチンの制御の下で働く。

コントロール・ルーチンは次の二つの部分から構成されている。

(1) LP モニタ

これは、LP 8000 のジョブをコントロールする部分であり、次に示す機能を持っている。

- (a) カード読取機からアジェンダム・カードを読み取り、コールすべきオーバレイ・ルーチン名を識別し、オーバレイ・ルーチンをオーバレイ・エリアにロードする。
- (b) オーバレイ・ルーチンの実行すべき機能を制御する。
- (c) オーバレイ・ルーチンからコントロールを受け取って次のアジェンダム・カードを読み込むか、あるいは新たなオーバレイ・ルーチンをコールする。
- (d) コンソール・タイプライタからの割り込みを解析し、要求された機能を実行する。
- (e) 与えられたアジェンダム・カードの順序が正しいかどうか

表2 実験結果

式の数	変数の数	密度	所要時間	反復回数	処理装置	メモリの大きさ	備考
60	117	4.78	45 s	70	8400	131 KB	C, S
100	240	3.28	3m30 s	141	8400	131 KB	C, S
131	280	2.69	11m00 s	327	8400	131 KB	C, S
266	823	1.56	2h05m	873	8400	131 KB	
266	823	1.56	44m	873	8500	131 KB	
283	594	1.15	1h30m	950	8400	131 KB	
283	594	1.15	32m	950	8500	131 KB	
575	1,388	0.54	6h41m	1,983	8400	131 KB	S
575	1,388	0.54	2h10m	1,983	8500	131 KB	S
846	3,756	0.35	9h04m	2,310	8400	131 KB	

かをチェックする。

(2) LP 入出力コントロール・ルーチン

LP 8000 の標準的な入出力動作をコントロールする。

6.2 LP 8000 コミュニケーション・リージョン

これは、コントロール・ルーチンと各オーバーレイ・ルーチン、あるいは各オーバーレイ・ルーチン間の連絡情報や、各種パラメータ、各種エリアのアドレス・テーブルなどをたくわえるエリアである。その内容に従って、次の二つの部分から構成されている。

- (1) GETOFF 時にリスタート情報として採出される部分。
- (2) GETOFF 時にリスタート情報として採出されない部分。

6.3 LP 8000 オーバレイ・エリア

これはオーバーレイ・ルーチンがロードされるエリアである。

6.4 LP 8000 コモン・ワーク・エリア

これは、各オーバーレイ・ルーチンが使用する共通のワーク・エリアである。このエリアの大きさは与えられた問題の式の数によって異なり、式の数が m であるとき、

$$(50m + 656 + \alpha) \text{ バイト} \quad (0 \leq \alpha \leq 6)$$

になる。

6.5 LP 8000 入出力バッファ・エリア

これは、入力データ、および E_{ta} レコードの入出力のためのバッファ・エリアであり、おのおの2個のバッファ・エリアから成る。

このエリアの大きさは、LP 8000 システムに割り当てられたエリアの大きさによって決まる。すなわち、LP 8000 システムに割り当てられたエリアから、6.1, 6.2, 6.3, 6.4 で説明した各エリアを差引いた残りのエリアを入出力バッファ・エリアに割り当てる。ただし与えられた問題の式の数が m であるとき、このエリアは少なくとも $(32m + 24)$ バイトの大きさでなければならない。

7. 実験結果

表2はLP 8000 で実際に行なった実験結果を示すものである。変数の数にはスラック変数も含まれている。備考欄のCは、コンポジット・プライシングを指定したことを示し、Sは、サブオプティマイゼーションを指定したことを示している。この表からわかるように、8500 でのランでは、8400 でのランの約3倍のスピードになる。

8. 結 言

LP 8000 リニア・プログラミング・システムは、昭和43年3月に完成し現在稼働中である。

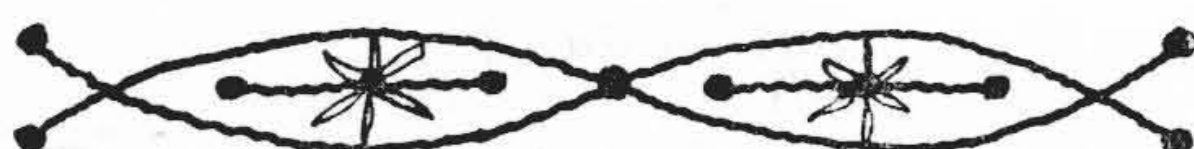
終わりに、本システムの開発に当たってご指導、ご協力をいただいた日立製作所神奈川工場の河野部長、松原主任技師をはじめ関係者のかたがたに感謝の意を表する。

参 考 文 献

- (1) HITAC 4010 Linear Programming System
- (2) HITAC EDP LP 5020 I Linear Programming System
- (3) HITAC 8300/8400/8500 TOS/TDOS/DOS FCP とエグゼクティブ連絡マクロ
- (4) HITAC 8300/8400/8500 TDOS コントロール・システム
- (5) HITAC 8300/8400/8500 DOS コントロール・システム
- (6) HITAC 8300/8400/8500 LP 8000 Linear Programming System
- (7) HITAC 8300/8400/8500 LP Translator (LPTRAN)
- (8) W. Orchard-Hays, 'Matrices. Elimination and the Simplex Method', C-E-I-R



特 許 の 紹 介



特許第522354号 (特公昭42-21536号)

高橋 長一郎・今井 利宣
太田 健三

波付アルミ被OFケーブルの端末金具取付方法

この発明は波付きアルミ被OFケーブルの端末に取り付けられる端末キャップ、仮油止管、ブーリングアイなどの端末金具の取付方法に関するものであり、波付きアルミ被OFケーブルの端末部分のシース1の表面に、アルミハンダによりメッキ2を設けたのち、ケーブルを切断し、その直後メッキを設けたシース端に、そのシース表面の凹凸みぞに合致する凹凸みぞを内面に有する筒状もしくは、半割筒状の端末金具3をネジ込みまたはかぶせ、さらに端末金具をその外周から絞り、さらにこの上のバインド線4を締め付けてから端末金具の片端部とその付近のシースとを鉛工5するものである。この発明の方法によるとシース表面の凹凸みぞは、従来のように鉛工ハンダによって埋めるのではないからケーブル線心に対する熱的な影響はまったくなく、またシース端にネジ込まれた端末金具はその外周から絞られ、しかもバインド線により締め付けられているため、キャップ内面とシース表面の接触は非常に気密となり、両面の接触間からの油の流出はまったくなく、この結果端末金具とシースの鉛工5作業は非常に容易となる。(斎藤)

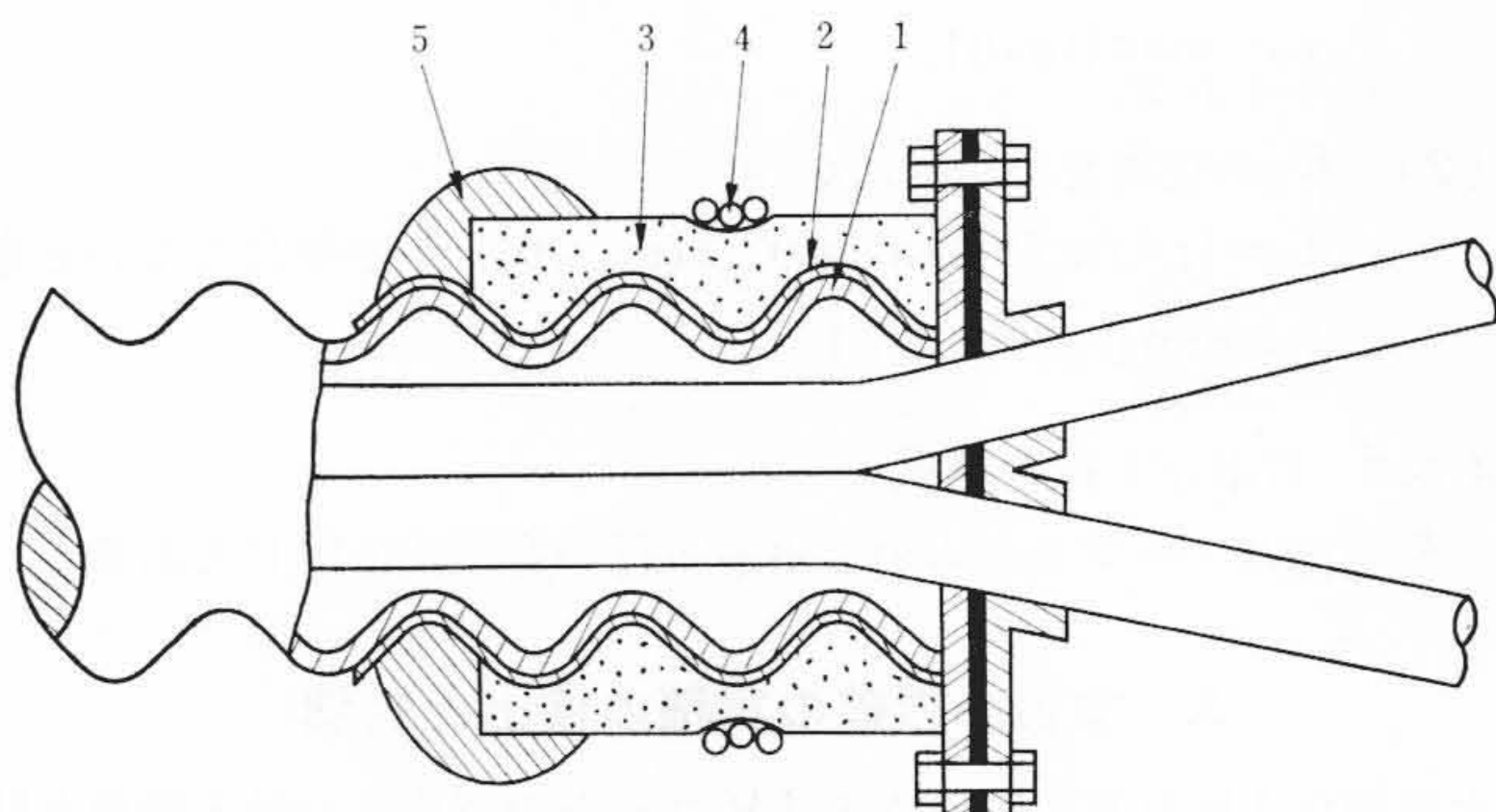


図 1