分散型問題解決に基づく実行可能解の探索

小野木 克 明*・川 上 茂 人**・竹 村 学***・西 村 義 行*

A Method of Seeking Feasible Solutions Based on Distributed Problem Solving

Katsuaki ONOGI*, Sighito KAWAKAMI**, Manabu TAKEMURA*** and Yoshiyuki NISHIMURA*

A problem of seeking feasible solutions under a large number of constraints is decomposed into a set of interconnected subproblems. If two subproblems share same variables, equality constraints arise between the subproblems. These equality constraints represent the interconnections of subproblems. This paper first shows that such a problem can further be transformed into a two-level optimization problem composed of small-scale optimization subproblems and one coordination problem. Auxiliary coordination variables are introduced to coordinate the common decision variables included in two or more subproblems. It second proposes a method of seeking feasible solutions based on cooperative in distributed problem solving. Feasible solutions are sought through the minimization with respect to the decision variables in each subproblem of the first level and through the minimization with respect to the coordination variables in the second level. If many feasible solutions exist, the coordination in the second level causes a multimodal problem. The paper also discusses a problem of separating peaks from multimodal function. The feasible solution seeking method proposed here seems to easily manage knowledge necessary to problem solving, to decrease effects of curse of dimensionality, and to have potential possibilities of parallel processing.

Key Words: distributed problem solving, cooperative approach, multimodal problem, feasible solutions seeking method

1. ま え が き

計画・設計問題においては、システムの仕様が与えられたとき、これを実現するようなシステムの構成とそれに含まれる構成要素の特性を決定することが求められる。その際、仕様を満たすシステム構成および構成要素がただ1通りに定まることはほとんどなく、システムの大規模化に伴いその数は組合せ論的に増大する。大規模なシステムの計画・設計問題では、問題がもつ構造を利用して問題をいくつかの部分問題に分散させ、それらを独立に解いて得られた解を協調を図りながら統合することによって問題解決を図ることが多い。分解法¹⁾はこのような考え方に基づいた最適化手法であり、大規模な数値計画問題の解法として現在広く用いられている。しかし、このような分散された知識源が協調的に問題解決を図る分散協調型問題解決の考え方^{2),3)}は最適化問題に限らず、計画・設計問題に頻出するような大規模な連立方程式の解を見出す問題にも適用できるものと思われる。

多数の制約条件のもとで実行可能解を見出す問題は、制約条件をいくつかの組に分割して得られた部分問題と、これらの部分問題間の干渉を干渉入力 x と干渉出力 y とに分け x と y の間に成り立つ付加的制約(協調条件) $x=y$ からなる問題に変換することができる。本論文では、最初に、この種の問題をさらにいくつかの下位最適化問題とこれらの解の統合を図る一つの上位最適化問題からなる2レベル最適化問題として定式化し、これを解くための一つの手法を提案する。この手法は、上位最適化問題に含まれる決定変数の初期値が1組与えられたとき、これをもとに両レベルの最適化計算を繰り返しながら、もとの問題の一つの実行可能解を見出そうとするものである。ここで、もとの問題に複数個の実行可能解が存在するならば、上位最適化問題に含まれる目的関数は多峰性関数となる。したがって、このような場合には、極値の探索・

* 豊橋技術科学大学 豊橋市天伯町字雲雀が丘 1-1

** 横河電機(株) 武蔵野市中町 2-9-32

*** 鶴岡工業高等専門学校 鶴岡市大字井岡字沢田 104

* Toyohashi University of Technology, Toyohashi

** Yokogawa Electric Corporation, Musashino

*** Tsuruoka National College of Technology, Tsuruoka

(Received October 12, 1989)

(Revised November 6, 1990)

域をあらかじめ絞り込んだのち、これらの領域内の点を初期値に設定しながら、提案する手法を繰り返し適用することが望ましい。本論文では、多峰性関数の極値探索問題についても考察する。

2. 分散協調方式に基づく実行可能解探索

2.1 問題の定式化

計画・設計問題においては、計画・設計をサブシステムごとに進め、それらの結果を統合することによってシステム全体の計画・設計を図るような問題解決方式がとられることが多い。その際、サブシステムがもつ目的は互いに競合することが多く、これを解消するためにサブシステム間でどのように協調をとるかが重要な問題となる。本章では実行可能解探索のための一つの分散協調方式を提案する。

以下では、つぎのような N 個の部分問題と M 組の協調条件からなる問題(P)を考える。

問題(P)：

つぎの制約条件を満たす実行可能解

$$(x_{I(i)}, y_{O(i)}, u^i), \quad i=1, 2, \dots, N$$

を求めよ。

1. 部分問題 i の制約条件

$$(x_{I(i)}, y_{O(i)}, u^i) \in \Omega^i, \quad i=1, 2, \dots, N$$

2. 部分問題間の協調条件

$$x_p = y_p, \quad p=1, 2, \dots, M$$

ただし

N ：部分問題の個数

M ：協調条件の組数

x_p ： p 番目の干渉入力を表わす入力変数ベクトル

y_p ： p 番目の干渉出力を表わす出力変数ベクトル

$I(i)$ ：部分問題 i への入力変数ベクトルにつけられた添字の集合

$O(i)$ ：部分問題 i からの出力変数ベクトルにつけられた添字の集合

$x_{I(i)}$ ：部分問題 i への入力変数ベクトルを並べたベクトル

$y_{O(i)}$ ：部分問題 i からの出力変数ベクトルを並べたベクトル

u^i ：部分問題 i 固有の決定変数ベクトル ■

この問題は、Fig. 1 に示すようないくつかのサブシステムとそれらの間に存在する等号制約条件から構成されるシステム全体の実行可能解を求める問題に対応しており、連立方程式の求解問題はこのような形に書き直すことができる。以下では、 u に含まれる決定変数は数値以外の値をとることもできるが、 x 、 y は実変数ベクトルとする。したがって、もとの問題のな

Fig. 1 System composed of N interconnected subsystems

かに実変数以外の変数が存在するならば、複数個の部分問題がこれらの変数を共通な変数として含まないように問題を分割しておくことが必要である。

問題(P)は

$$x = (x_1, x_2, \dots, x_M)$$

$$y = (y_1, y_2, \dots, y_M)$$

$$u = (u^1, u^2, \dots, u^N)$$

$$\Omega = \{(x, y, u) \mid (x_{I(i)}, y_{O(i)}, u^i) \in \Omega^i,$$

$$i=1, 2, \dots, N\}$$

とおくことによってつぎのように表わすこともできる。

問題(P)：

つぎの制約条件を満たす実行可能解を求めよ。

$$(x, y, u) \in \Omega$$

$$x = y$$
 ■

2.2. 2レベル最適化問題としての定式化

問題(P)は、つぎのような最適化問題(P1)において最小値が0となる最適解を求めることと同等である。

問題(P1)：

$$\text{Minimize}_{x, y} \left\{ \frac{1}{2} \|x - y\|^2 \mid (x, y, u) \in \Omega \right\}$$
 ■

恒等式

$$\begin{aligned} & \|x - v\|^2 + \|y - v\|^2 \\ &= \frac{1}{2} \|x - y\|^2 + 2 \left\| \frac{1}{2} (x + y) - v \right\|^2 \end{aligned} \quad (1)$$

から、(1)式の左辺は $v = (x + y)/2$ のとき v に関して最小となり、最小値は

$$\min_v \{ \|x - v\|^2 + \|y - v\|^2 \} = \frac{1}{2} \|x - y\|^2 \quad (2)$$

となる。したがって、 $\|x - y\|^2/2$ の最小値は

$$\min_{x, y} \frac{1}{2} \|x - y\|^2 = \min_{x, y} \min_v \{ \|x - v\|^2 + \|y - v\|^2 \} \quad (3)$$

で与えられることとなる。ここで(3)式右辺の最小化の順序を入れ替えることによって、問題(P1)はつぎのような2レベル最適化問題に書き直すことができる。

問題 (P2) :

$$\underset{v}{\text{Minimize}} \{F(v) | v \in E^n\}$$

ただし

$$F(v) = \min_{x, y} \{\|x - v\|^2 + \|y - v\|^2 | (x, y, u) \in Q\} \quad \blacksquare$$

問題 (P2) の最適解が $v = (x + y)/2$ を満たすならば、問題 (P1) は解けたことになる。この関係は 2.3 節に示す問題 (P1) の解法アルゴリズムの停止条件として利用される。

問題 (P2) は、 $F(v)$ を求める問題を部分問題に分割することによって、つぎのような問題に変換できる。

問題 (S) :

$$\underset{v}{\text{Minimize}} \{F(v) | v \in E^n\}$$

ただし

$$F(v) = \sum_{i=1}^N F^i(v)$$

ここで、それぞれの $F^i(v)$ はつぎのような部分問題 (Q_i) の解によって定められる。

<問題 (Q_i)>

$$F^i(v) = \min_{x_{I(i)}, y_{O(i)}, u^i} \{\|x_{I(i)} - v_{I(i)}\|^2 + \|y_{O(i)} - v_{O(i)}\|^2 | (x_{I(i)}, y_{O(i)}, u^i) \in Q_i\}$$

ただし

$v_{I(i)}, v_{O(i)} : x_{I(i)}, y_{O(i)}$ の目標を表わすベクトル

このようにもとの実行可能解探索問題 (P) は、サブシステムごとに定義されるいくつかの最適化問題を含む 2 レベルの最適化問題 (S) として再定式化される。

2.3 分散協調アルゴリズム

問題 (S) を解くことによって原問題 (P) を解く手順はつぎようになる。

手順 1 : 問題 (S) を解くことによって、問題 (P1) を解く。この概略を Fig. 2 に示す。

- ① 部分問題間の協調条件 $x = y$ を取り除き、 x , y の共通の目標を表わす v の初期ベクトルを設定する。
- ② 現在の v のもとで各部分問題 (Q_i) を解き $x_{I(i)}$, $y_{O(i)}$, $F^i(v)$ および $F(v)$ を求める。
- ③ ② で得られた x , y が停止条件 $v = (x + y)/2$ を満たすならば手順 2 に移る。そうでなければ、 $F(v)$ の値が小さくなるように v の値を変更し ② に進む。

手順 2 : 手順 1 で得られた問題 (P1) の最適解が $\min_v F(v) = 0$ を満たすならば、問題 (P) の実行可能解が得られたこととなり終了する。そうでなければ手順 1 に戻る。

Fig. 2 Cooperative method of seeking feasible solutions

手順 1 中の条件 $v = (x + y)/2$ は問題 (S) の最適解が問題 (P1) の最適解にもなっていることを保証するものである。また、手順 2 中の条件 $\min_v F(v) = 0$ はその解がさらに協調条件を満たすことを保証するものである。このようにもとの問題 (P) の解法は、部分問題 (Q_i) を解くこと (分散) と、それらの解が実行可能解となるように v を調整すること (協調) に帰着できる。

2.4 分散型問題解決システムのアーキテクチャ

分散型問題解決法に類する求解法は、分散型問題解決法に含まれないこともある。しかし、その概念に従ってハードウェアが分割されそれらの間に通信を伴う場合には、それは分散型問題解決法とみなすことができる⁴⁾。この意味において、2.3 節で提案した手法も、上位最適化問題の解決と下位最適化問題 (部分問題) の解決をそれぞれ異なったハードウェア (プロセッサなど) で実現させるならば、一つの分散型問題解決法とみなすことができる。

提案した手法を実現するような一つの分散型問題解決システムのアーキテクチャを Fig. 3 に示す。システムの目的を達成するために必要な仕事の種類とそれらの実行順序は 2.3 節の手法に従ってあらかじめ定められており、これらに基づいてエージェント間の通信経路も固定される。 E_0 は上位最適化問題の解決を図るエージェントであり、そこにはそれに必要な情報のほかに、それぞれの部分問題の解決をどのエージェントに依頼するかという情報も備えられる。 E_i ($i=1$,

Fig. 3 Architecture of distributed problem solving system

$2, \dots, N$ は部分問題 (Q_i) の解決を図るエージェントであり、そこには問題 (Q_i) の解決に必要な情報が備えられる。システムの動作は、 E_0 がそれぞれの E_i に問題 (Q_i) の解決に必要な初期データ $v_{I(i)}$, $v_{O(i)}$ を送信することによって開始される。 E_i は E_0 から送信されたデータをもとに、

$$\|x_{I(i)} - v_{I(i)}\|^2 + \|y_{O(i)} - v_{O(i)}\|^2$$

を最小にする $(x_{I(i)}, y_{O(i)}, u^i) \in Q^i$ を求め、このうち $x_{I(i)}$, $y_{O(i)}$ を E_0 に送信する。この仕事は互いに他のエージェントの仕事の進行状況に影響されことなく、非同期に並行して行われる。 E_0 は他のすべてのエージェントからデータ $x_{I(i)}$, $y_{O(i)}$ が到着すると、それらをもとにシステムの目的が達成されたかどうか (停止条件が満たされたかどうか) を調べる。その結果、目的が達成されていない場合は、システム全体として協調がとれるような $v_{I(i)}$, $v_{O(i)}$ を適当な最適化手法を用いて再検討し、それらをそれぞれの E_i にあらためて送信する。システムは、 E_0 と各 E_i が必要に応じて通信しながらそれぞれの仕事を進めることによって一つの実行可能解を求める。

もとの問題 (P) が複数個の実行可能解をもつならば、システム全体の協調をとることができるような $v_{I(i)}$, $v_{O(i)}$ も複数組存在する。このような場合、問題解決の効率化を図るためには、 v の探索空間において、目的を達成することができるような v が存在すると思われる領域をあらかじめ絞り込んだのち、得られた個々の領域内の v を初期データとして、上に示した一連の動作を繰り返し実行することが望ましいと考えられる。3章では多峰性関数の極値を絞り込むための一つの手法について考察する。なお、もとの問題を部分問題に細かく分散しすぎると、 E_0 と E_i の通信に要する負荷が増加するとともに、システムの協調を図るのに要する E_0 の負荷も増加する。一方、分散が大まかすぎると、部分問題の解決に要する E_i の負荷が増加する。もとの問題をどの程度まで分散させればよ

いかについては、今後の問題であると思われる。

提案した手法の特徴として、つぎのことが挙げられる。

(1) 問題解決に必要な知識 (制約条件) を部分問題ごとに分散させることによって知識の整理・管理が容易となる。このため、一部の知識の修正・拡張で問題の変更に対応することができる。

(2) 大規模な問題をいくつかの小規模な部分問題に分散させて解くことができるため、次元の呪いの影響を小さくすることができる。

(3) それぞれの部分問題は互いに独立に解くことができるため、並列処理が期待できる。

3. 多峰性関数の極値探索

3.1 多峰性関数の単峰分離

計画・設計問題においては、一般に複数個の実行可能解が存在するため、2章で示した問題 (S) に現れる関数 $F(v)$ は多峰性関数となることが多い。しかし、従来のほとんどの最適化手法は目的関数が単峰であることを仮定しており、このため、それらの手法を多峰性最適化問題に適用するためには、試行の始点をいろいろ変えることが必要となる。したがって、この種の最適化問題では、(1) あらかじめ多峰を単峰に分離したのち、(2) 分離された峰ごとに従来の最適化手法を適用することが有効であると思われる。

(1) を目的とする手法として、沖野らは探索領域内に一様に分布する探索点を乱数を用いて作成し、それらをもとに各峰の広がり具合を近似的に表わす指標と、これを活用した単峰分離手法を提案した⁵⁾。以下では、この考え方に基き問題 (S) 中の多峰性関数

$$Z = F(v) = F(v_1, v_2, \dots, v_n) \quad (4)$$

がもつ特徴を活かし、その最小点を探索するための手法について考える。なお、 v の成分 v_i はいずれも

$$0 \leq v_i \leq 1, \quad i = 1, 2, \dots, n \quad (5)$$

に標準化されているものとする。

n 個の変数 v_i ($i = 1, 2, \dots, n$) と Z が張る $n+1$ 次元空間を S とし、 S 内の

$$Z = Z_0, \quad Z_0 > 0 \quad (6)$$

で表わされる超平面 π_0 を考える。 $Z = F(v)$ がつくる超曲面を π_0 で切断すると、Fig. 4 に示すようないくつかの閉領域 $a_1, a_2, \dots$ ができる。これらの閉領域を π_0 上での極小領域という。極小領域には必ず極小点が含まれるが、その個数は一つとは限らない。しかし、極小領域の面積が小さくなればなるほど、その領域にただ一つの極小点が含まれる確率は高くなるものと考えられる。

Fig. 4 Minimum region

3.2 極小領域の大きさと探索点数

システムの計画・設計問題においては、既存の部品、機械、装置などを組み合わせることによって問題解決を図ることが多い。このような状況のもとでは、決定変数は離散的となり、 $F(v)$ はつぎのような性質をもつ関数となる。

(i) $F(v^*)=0$. $v \neq v^*$ では、 $F(v)>0$

(ii) v^* 近傍では、 $F(v)=2\|v^*-v\|^2$

ここで、 (x^*, y^*, u^*) は問題 (P) の実行可能解を、 v^* はこれに対応する v を表わす。(i) は実行可能解では、 $x^*=y^*=v^*$ となることより明らかである。また、(ii) は x, y が離散的で、 v が v^* に十分近いならば、

$$\min_x \|x-v\|^2 = \|x^*-v\|^2 = \|v^*-v\|^2$$

$$\min_y \|y-v\|^2 = \|y^*-v\|^2 = \|v^*-v\|^2$$

となることよりわかる。したがって、 Z_0 が小さいとき、 π_0 上での極小領域は中心が v^* で、

$$2r^2 = Z_0 \quad (7)$$

を満たす半径 r の超円の内部で与えられる。

(5) 式で定義される π_0 上の超正方形の超面積は 1 である。したがって、この超正方形内に一様に分布する探索点を K 個発生させたとき、これらの探索点のうち少なくとも 1 個が超面積 A の超円の内部に入る確率 p は

$$p = 1 - (1-A)^K \quad (8)$$

となる。これより信頼度 p で v^* を含む極小領域が見い出されるために必要な探索回数 K は

$$K \geq \log(1-p)/\log(1-A) \quad (9)$$

となる。

3.3 単峰分離アルゴリズム

Fig. 2 に示した分散協調方式に従って問

題 (P) を解くためには、まず初めに $F(v)$ の多峰を単峰分離し、峰ごとに v の初期ベクトルを設定することが必要である。単峰分離のための手順を Fig. 5 に示す。

- 1: 極小領域の大きさを表わす超円の半径 r をもとに、切断面 $\pi_0: Z=Z_0$ を (7) 式より、探索回数 K を (9) 式より定め、 $j=1$ とおく。
- 2: 探索点 $v^{(j)}$ を乱数を用いて設定し、この点における Z の値 $Z^{(j)}=F(v^{(j)})$ を求める。 $v^{(j)}$ を中心とする半径 r の超円を $a_{(j)}$ で表わす。
- 3: $Z^{(j)} < Z_0$ ならば 4 へ進む。そうでなければ、5 へ進む。
- 4: 現在までに見い出された極小領域の候補を表わす超円

$$a_1, a_2, \dots, a_T$$

(ただし a_i の中心を v_i とし、 $Z^i=F(v^i)$ とすると、各 a_i は $Z^1 \geq Z^2 \geq \dots \geq Z^T$ の順に並べられている) について、 a_1 から順に $a_{(j)}$ との重なりが存在するかどうかを調べる。もし $a_{(j)}$ と a_i が

Fig. 5 Peak separating algorithm

重なりをもつならば、すなわち

$$\|v^{(j)} - v^i\| < 2r \quad (10)$$

ならば、 $Z^{(j)}$ と Z^i のうち大きい値をもつ超円を取り除く。これを $a_{(j)}$ が取り除かれるまで、またはすべての a_i との比較が完了するまで続ける。こうして残った超円をあらたな極小領域の候補として Z の値の大きいものから順に再配列し、5に進む。

5: $j \geq K$ ならば終了する。そうでなければ、 $j = j + 1$ とおき、2へ戻る。 □

(10)式は二つの異なった極小領域の候補が同じ極小点を含むとみなすための一つのヒューリスティックな基準を表わすものであり⁹⁾、この関係が成立すると、大きい Z の値をもつ超円は小さい Z の値をもつ超円に吸収される。

問題(P)の実行可能解は、こうして分離された峰ごとに v^i を v の初期ベクトルとして Fig. 2 の分散協調方式を適用することによって求められる。その際、部分問題(Q_i)中の関数 F^i は必ずしも連続微分可能な関数になるとは限らない。したがって、以下では $F(v)$ の最小化手法として、Nelder-Mead のシンプレックス法を用いることとした。

3.4 探索効率に関する考察

超円の半径 r を大きく設定すると超面積 A が大きくなりその結果(9)式より探索点は少なくなるが、いくつかの極小点が接近している場合にはそれらが完全に分離できないこともある。逆に、 r を小さくしすぎると探索点が多くなり最適化に費やす時間が増加する。そこでつぎのようなテスト問題(T)を解くことによって、 r と単峰分離の効率との関係について考察した。

問題(T)：

$$\underset{v}{\text{Minimize}} \{F(v) | v \in E^2\}$$

ただし

$$F(v) = F^1(v) + F^2(v)$$

は部分問題 Q_1 , Q_2 の解によって定められる。

$\langle Q_1 \rangle$

$$F^1(v) = \min \|y - v\|^2 | y = u^1, u^1 \in \Gamma_1\}$$

ただし、 $\Gamma_1 = \{\alpha_1, \dots, \alpha_L | \alpha_i \in E^2\}$

$\langle Q_2 \rangle$

$$F^2(v) = \min \|x - v\|^2 | x = u^2, u^2 \in \Gamma_2\}$$

ただし、 $\Gamma_2 = \{\beta_1, \dots, \beta_L | \beta_i \in E^2\}$ ■

数値実験では、 $L=50$ とし、 α_i , β_i はそれぞれ(0,1)の一様乱数を用いて作成した。 $x=y$ を満たす実行可能解は、いく組かの α_i と β_i が $\alpha_i = \beta_i$ となるように、それらに同じベクトルを割り当てることによって作成した。

Fig. 6 Effect of r on separating performance
 N_m : Number of minimum region candidates
 N_f : Number of feasible solutions found by experiments

実験によって求められた実行可能解の個数 N_f と r との関係を図. 6 に示す。図中のパラメータは真の実行可能解の個数を表わす。また、データは50回の実験の結果を平均したものであり、 $p=0.9$ とした。この図より、実行可能解の個数が4または8ならば、 r が0.04以下で N_f はほぼ一定となり、実行可能解の個数が16ならば、 r が0.02以下で N_f はほぼ一定となることがわかる。したがって、 r の値は前者では0.04以下に、後者では0.02以下に設定すればよいものと思われる。

このような実験を繰り返した結果、つぎのことが確かめられた。

(1) 得られる極小領域の候補数 N_m は r の適当な値で最大となる。 r が大きいとき極小領域の候補数が小さくなるのは、一つの超円に複数個の極値が含まれやすくなるためである。

(2) r を適当に小さくすると大部分の実行可能解を求めることができる。しかし、 r を必要以上に小さくしてもその効果はほとんどない。

(3) 実行可能解が得られる確率は、実行可能解が一樣に分布するならばその数にほとんど影響されない。

以上より、 r は適当に小さい値に設定すればよいものと思われる。

4. 適用例

2 章において提案した実行可能解探索手法をつぎのような設計問題に適用した。

“Fig. 7 に示すような二段減速歯車列において、入力軸の回転数 N_i 、出力軸の回転数 N_o 、および入力軸と出力軸との間の距離 H が与えられたとき、これを実現するような各歯車の歯数とモジュールを求めよ。”

この設計問題は、軸間の距離および速比に関する二つの制約条件のもとで、4 個の歯車の歯数と 2 組の歯車列に含まれる歯車のモジュールからなる 6 個の決定変数の値を定める問題として定式化される。さらにこの問題は、これらの決定変数を 2 組に分割することによって、つぎのような条件のもとでそれぞれの歯車列を独自に設計するような小規模な問題に書き直すことができる。

1. 歯車列 1 の設計条件 C1

$$h_1 = \frac{1}{2} M_1 (Z_a + Z_b), \quad 0 < h_1 < H$$

(ピッチ円の半径, 歯数, モジュールの関係)

$$n_1 = N_i \frac{Z_a}{Z_b} \quad (\text{速比})$$

$$M_1 \in \Omega_M, \quad Z_a, Z_b \in \Omega_Z \quad (\text{歯車の規格})$$

2. 歯車列 2 の設計条件 C2

$$h_2 = \frac{1}{2} M_2 (Z_c + Z_d), \quad 0 < h_2 < H$$

$$n_2 = N_o \frac{Z_c}{Z_d}$$

$$h_3 = H - h_2$$

$$M_2 \in \Omega_M, \quad Z_c, Z_d \in \Omega_Z$$

3. 歯車列の関係 (協調条件)

$$h_1 = h_3, \quad n_1 = n_2$$

ただし

Z_a, Z_b, Z_c, Z_d : 対応する歯車の歯数

M_i : 歯車列 i に含まれる歯車のモジュール

h_1 : 入力軸と中間軸の間の距離

h_2 : 出力軸と中間軸の間の距離

Fig. 7 Double reduction gear

n_1, n_2 : 中間軸の回転数

ここで, 2.2 節に示したように

$$u^1 = (Z_a, Z_b, M_1), \quad u^2 = (Z_c, Z_d, M_2)$$

$$x = (h_3/H, n_2/N_i), \quad y = (h_1/H, n_1/N_i)$$

とおくことによって, この問題はつぎのような 2 レベル最適化問題 (E) として定式化される。

問題 (E):

$$\text{Minimize } \{F(v) | v \in E^2\}$$

ただし

$$F(v) = F^1(v) + F^2(v)$$

はつぎの部分問題の解によって定まる。

〈部分問題 1〉

$$F^1(v) = \min_y \{\|y - v\|^2 | y \text{ は C1 を満たす}\}$$

〈部分問題 2〉

$$F^2(v) = \min_x \{\|x - v\|^2 | x \text{ は C2 を満たす}\}$$

問題 (E) は, 本来 6 変数からなる実行可能解探索問題を二つの 3 変数最適化問題を含む 2 レベル最適化問題として再定式化したものであり, 2.3 節の手法はこれ

(a) Example a

(b) Example b

Fig. 8 Distribution of feasible solutions ($H=126$, $N_i=1600$, $N_o=400$)

Table 1 Numerical example result

Example	r	0.017	0.020	0.025
	K	2535	1831	1170
a	N_m	5.73	6.63	9.45
	N_f	3.78	3.75	3.65
b	N_m	7.97	10.2	11.2
	N_f	3.25	2.93	2.40

に最適化手法を適用し実行可能解を組織的に求めようとするものである。

四つの実行可能解が Fig. 8(a)のように探索領域内に一様に分布する場合、および(b)のように集中して分布する場合について、問題(E)を解いた結果を Table 1 に示す。それぞれの実行可能解は、3章で示した極値探索手法を使って $F(v)$ の極値が存在すると予想される探索領域をあらかじめ絞り込んだのち、これらの領域内の v を初期値として2章で示した手法を繰り返し適用することによって求めた。ただし、 $p=0.9$ であり、いずれの数値も50回の数値実験の結果を平均したものである。(a)の場合、超円の半径 r が小さくなると極小領域の候補数は減少するが、得られる実行可能解の個数は r の大きさにはほとんど依存せず一定となる。これより、複数個の極小点が一様に分布するならば、 r をある程度小さくしさえすれば単峰分離が可能であることがわかる。一方、(b)のように集中して存在する実行可能解を得るためには、(a)の場合に比べ r をかなり小さい値に設定することが必

要である。

5. あとがき

多数の制約条件のもとで実行可能解を見出す問題は、いくつかの部分問題とそれらの間に課せられた協調条件からなる問題に変換することができる。本論文では、この問題をさらに2レベル最適化問題としてとらえ、この解法としてある分散協調方式に基づく手法を提案した。また、多峰性関数の極値探索手法についても考察した。提案した実行可能探索手法は問題解決に必要な知識を分散させることによって、(1)知識の整理・管理が容易となる。(2)次元の呪いの影響が小さくできる、(3)並列処理が期待できる、の特徴をもつ。なお、この手法を大規模な問題に適用するためには、対象となる問題をどのように分散させるかが重要となる。これが残された課題であると思われる。

参考文献

- 1) G. Dantzig and P. Wolfe: Decomposition Principle for Linear Programs, Oper. Res., 8, 101/111 (1960)
- 2) R.G. Smith: Framework for Cooperative in Distributed Problem Solving, IEEE Trans. Systems, Man, and Cybernetics, SMC-11-1, 61/70 (1981)
- 3) R.G. Smith: Report on 1984 Distributed Artificial Intelligence Workshop, The AI Magazine, 234/243 (1985)
- 4) 市川惇信: システム分散化を支える技術, 計測と制御, 26-1, 11/18 (1987)
- 5) 沖野, 佐々木: モンテカルロ積分を利用した多峰性関数の各峰分離, 計測自動制御学会論文集, 8-6, 676/686 (1972)