



Title	分散協調型設計支援システムのためのプロセスモデルと操作モデル（第2報, 設計操作のためのモデルとプロセス管理）
Author(s)	藤田, 喜久雄; 菊池, 慎市
Citation	日本機械学会論文集 C編. 2002, 68(666), p. 666-674
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/2922">https://hdl.handle.net/11094/2922</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

## 分散協調型設計支援システムのためのプロセスモデルと操作モデル (第 2 報 : 設計操作のためのモデルとプロセス管理)\*

藤田 喜久雄<sup>\*1</sup>, 菊池 慎市<sup>\*2</sup>

### Computational Models for Concurrent Design Process Support (2nd Report: Design Operation Model and Process Management)\*

Kikuo FUJITA<sup>\*3</sup> and Shin'ichi KIKUCHI

<sup>\*3</sup> *Department of Computer-Controlled Mechanical Systems, Osaka University,  
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan*

This research proposes computational models for concurrent design process of large complicated systems toward the development of agent-based distributed design support systems. In this second report, we mathematically formulate the partial design problem of each divided task in the distributed design process under the concepts introduced in the first report. Following them, the backward-chaining operation for refining a tentative design solution is modeled as an optimization problem. Its form belongs to a style of goal programming, since the goals, each of which is the play of an intermediate variable among divided tasks, are defined with a set of regions and domains. The associated mathematical programming method is configured by combining linear approximation, monotonicity analysis and active set strategy to support designer's decision making. The issues on design process management over such optimization-based support are further discussed. Finally, the coordinative design process among divided tasks is briefly demonstrated in an application to preliminary aircraft design.

**Key Words :** Design Engineering, Design Process, Goal-Driven Coordination, Computational Design Model, Optimization, Goal Programming, Monotonicity Analysis, Active Set Strategy

## 1 緒言

大規模なシステムの設計は組織によってあらかじめ想定された期間内で遂行されている。このようなプロセスにおいて、様々なコンピュータ援用設計支援技術が重要なツールとなりつつあるものの、各種の内容は設計における個別的な処理を効率化するものであって、設計プロセスの全体からみれば、いささか補助的なものに留まっていると言わざるを得ない。これに対して、設計シンセシスにおける各種情報間の関連性やタスク間の構造に対応した支援システムの構築を目標として、コンピュータ援用コンカレントエンジニアリングと称される試みが多方面から行われつつある。

本研究<sup>(1)(2)</sup>は、上記の動向を踏まえた上で、分散協調型の設計支援システムの実現に向けて、大規模なシステムの設計プロセスにおける規模性に起因する複雑さに着目した設計支援のためのモデルを確立することを目的としている。第 1 報<sup>(1)</sup>では、設計問題の内容が

大規模な集中定数系で表現できるものとした上で、あらかじめ定められた時間の範囲内ではそのような問題の完璧な解を求めることはできず、制限された状況下での何らかの優れた満足解を導出することが求められているものとの想定のもと、希求される設計プロセスの分散化における構造と協調処理の形態を導出した。

本報では、そのプロセス構造のもとで設計対象を操作するためのモデルを提案し、対象操作を通じてのプロセス管理について論じる。第 1 報<sup>(1)</sup>の内容を分散協調型設計支援に向けての「分散化」と「並行化」のための枠組みを与えるものであるとすれば、ここでの内容は、分散並行化のもとで合理的な設計を展開するための「協調化」に向けての枠組みを与えようとするものである<sup>(3)</sup>。なお、第 3 報<sup>(2)</sup>では、第 1 報と本報での内容をエージェント方式により実装する方法とその航空機の基本設計における展開例を示すことにより、一連のモデルの妥当性を検証する。

## 2 大規模システムにおける設計プロセスの構造

本節では、まず、準備として第 1 報<sup>(1)</sup>において導出したプロセス構造モデルの内容を整理する。

\* 原稿受付 2001 年 5 月 7 日

\*1 正員、大阪大学大学院工学研究科 (〒 565-0871 吹田市山田丘 2-1)。

\*2 富士写真フイルム (株) (〒 250-0193 神奈川県南足柄市中沼 210)  
(元 : 大阪大学大学院工学研究科)。

Email: fujita@mech.eng.osaka-u.ac.jp

**2.1 集中定数系としての形式化** 本研究では、分散協調型の設計問題における各種の困難性のなかでも、規模性に起因するものに着目するために、設定対象の内容が集中定数系として表現されており、その際の変数は連続変数であって、関数として表現されるそれらの間の関係式もかなりなめらかであるという仮定を導入している。この仮定を踏まえると、設計問題を構成する各種の変数は、設計の対象となるシステムがどのようなものであるかを表し、設計において決定できる内容に相当する決定変数 (decision variables)、対象システムがどのように振る舞うかに対応する性能変数 (performance variables)、さらに、両者の間の関係式が一連の部分関係式に分割される際にそれらの間に介在する媒介変数 (intermediate variables) とに分類できる。以下では、それぞれを  $x, z, y$  と表す。このとき、対象システムの振舞いを模擬するアナリシスに相当する操作は  $z = f(x)$  という式によって規定される。さらに、媒介変数が決定変数と性能変数の間に位置することから、この関係式は、決定変数の部分ベクトル  $x_{y_i}$  と媒介変数の部分ベクトル  $y_{y_i}$  からある媒介変数  $y_i$  の値を算出する  $y_i = f_{y_i}(x_{y_i}, y_{y_i})$  や、決定変数の部分ベクトル  $x_{z_j}$  と媒介変数の部分ベクトル  $y_{z_j}$  からある決定変数  $z_j$  の値を算出する  $z_j = f_{z_j}(x_{z_j}, y_{z_j})$  などとして分解できる。つまり、 $f(x)$  は  $f_{y_i}$  や  $f_{z_i}$  などの部分関係式の集合から構成されることになる。

**2.2 設計局面と設計知識** 上記のもとで設計プロセスを構成する各局面を以下のように解釈できる。

設計 (design) :  $x$  を (仮に) 設定する。

検証 (verify) :  $z$  を求める。

評価 (critique) : 求めた  $z$  についての判断を行う。

修正 (modify) : 評価の結果に基づいて  $x$  を調整する。

さらに、この分類を  $z = f(x)$  なる関係に照し合せれば、設計に関連する知識は以下のように分類できる。

前向き知識 (forward-chaining knowledge) : 検証の局面において  $f(x)$  を計算することに対応する知識であり、いわゆるアナリシスのための知識となる。

後向き知識 (backward-chaining knowledge) : 設計と修正の局面で用いられる知識であり、形式的には  $f^{-1}(z)$  に対応するが、その陽関数としての表現は明示的には得ることができない。

評価知識 (critique knowledge) : 評価の局面で用いられる知識であり、性能変数  $z$  が相互の調整に向けて適切であるかどうかを判定するものである。

なお、評価知識はいわゆる満足化において  $z$  を洗練化の上で不可欠なものであるが、その内容は  $z = f(x)$

という関係式に関連づけられるものではない。

**2.3 設計プロセスの分散構造** 以上の形式化や解釈のもとでは、大規模なシステムの設計問題は以下の2つの観点から分割することができる。

(1) 垂直分割構造 まず、決定変数と性能変数との間の関係における操作性と忠実度 (fidelity) とを調整しながら設計を進めるためには、前向き知識を複数の粒度 (granularity) レベルに渡って多重に保持することが必要である。このとき、決定変数  $x$  は各粒度レベル  $g_i$  において意味をもつ部分変数ベクトル  $x_{g_i}$  に分割され、一連の部分変数ベクトルの間には  $\{x_{g_1}\} \subset \{x_{g_2}\} \subset \dots \subset \{x_{g_i}\} \subset \dots$  なる関係が成立する。さらに、これに対応して、前向き知識も  $f|_{x_{g_1} \rightarrow z}, f|_{x_{g_2} \rightarrow z}, \dots, f|_{x_{g_i} \rightarrow z}, \dots$  として多重に存在し、つまり、設計問題は垂直に分割される。

関連して、後向き知識については、ある粒度レベル  $g_i$  における決定変数  $x_{g_i}$  を  $\{\widehat{x}_{g_i}\} = \{x_\ell | x_\ell \in \overline{x_{g_{i-1}}} \cap \{x_{g_i}\}\}$  という関係<sup>†1</sup>のもとで  $x_{g_{i-1}}$  と  $\widehat{x}_{g_i}$  という2つの部分に分割した場合に、 $\widetilde{f}_{g_i}^{-1}|_{z_j \rightarrow x_{g_i}}$  の具体的な内容は、一般に、 $\widetilde{f}_{g_i}^{-1}|_{z_j \rightarrow x_{g_{i-1}}}$  と  $\widetilde{f}_{g_i}^{-1}|_{z_j \rightarrow \widehat{x}_{g_i}}$  に分けて整備されていて、 $\widetilde{f}_{g_i}^{-1}|_{z_j \rightarrow x_{g_{i-1}}}$  は  $\widetilde{f}_{g_i}^{-1}|_{z_j \rightarrow \widehat{x}_{g_i}}$  に対して支配的である (dominate) とすることができる。

(2) 水平分割構造 一方、設計問題の規模性に対応して、設計問題を構成する対象モデルは、全体を粗く代表する主要目とそれらに関わる内容とから構成される大域モデル (global model) と、それ以外の内容である領域モデル (domain models) とに分割される。前者は設計における起点となるものであり、個々の設計問題において唯一のモデルが存在する。後者は背後にある技術領域に対応して適切なサイズの複数の部分に分割される。2.1項の内容に従えば、大域モデルを構成する各種変数は、起点になるものであるものの、最終的な決定は領域モデルにおいて行われることから、いずれかの領域モデルに含まれることになる。また、領域モデル間の関係については、因果関係のもとにある2つの領域モデル間の境界には特定の媒介変数が存在していて、両側のモデルはそれらを共有している。その変数の値を算出する側の領域モデルにおいてはそれを媒介出力変数と呼び、その値を参照する側の領域モデルにおいては媒介入力変数と呼ぶ。なお、上記の垂直分割は領域モデルの内部におけるものである。

<sup>†1</sup> 本論文では、その都度特記はしないが、一貫して下付添字の  $\cdot_\ell$  を変数ベクトルの任意の要素を表すために用いる。

**2.4 ゴールの導入と協調処理の形態** 前項に示した分散構造のもとで設計を進めて適切な結果を得るためには、それぞれの部分が適切な処理を並行的に進めるに加えて、領域相互や粒度レベル相互の関係が適切に処理される必要がある。本研究ではこれを実現するために各種の変数についての相互の関係に対して「ゴール (goal)」という考え方を導入する。その詳細については後述するが、各部分に分散的に定義された変数の情報に対してある種のあそびとしての範囲を与えて、相互の処理を通じてそれらを段階的に縮小していくことにより、最終的な設計を確定しようとするものである。このゴールの段階的な絞り込みを円滑に進めるためには、分散構造に呼応して、媒介変数を介して連結されている領域間での領域間協調 (inter coordination)、全体としてのバランスに優れた設計解を得ることを保証するための大域モデルによる大域的協調 (global coordination)、個別領域内における異なる粒度レベル間での関係を処理するための領域内協調 (inner coordination) という3つの協調処理形態が必要となる。

### 3 設計操作における数理モデルとその処理

**3.1 一般化部分問題** 2.3項に示したプロセス構造のもとでは様々な部分設計問題が構成される。設計操作におけるモデルについて議論するに当たって、領域モデルにおける特定の粒度レベルについての部分問題を一般形として取り上げることにする。なお、それが他の部分問題に展開できることに關しては後述する。

その一般形は領域モデル  $D_j$  における特定の粒度レベル  $g_i$  を対象とすると、次式により表現される。

$$\left. \begin{aligned} z_{D_j}^{(g_i)} &= f_{z_j, g_i}(x_{D_j, g_{i-1}}, \widehat{x_{D_j, g_i}}, y_{D_j}^{in}) \\ y_{D_j}^{out(g_i)} &= f_{y_j, g_i}(x_{D_j, g_{i-1}}, \widehat{x_{D_j, g_i}}, y_{D_j}^{in}) \end{aligned} \right\} \quad (1)$$

ここで、 $x_{D_j, g_{i-1}}$  は、粒度レベル  $g_{i-1}$  で決定済みの決定変数であり、上式の操作においては定数となる。 $\widehat{x_{D_j, g_i}}$  は  $x_{D_j, g_i}$  から  $x_{D_j, g_{i-1}}$  を除いたこの粒度レベルで操作可能な決定変数である。 $z_{D_j}^{(g_i)}$  はこの領域が担っている性能変数のこの粒度レベルにおける算出値である。 $y_{D_j}^{in}$  は媒介入力変数、 $y_{D_j}^{out(g_i)}$  は媒介出力変数の算出値である。なお、厳密には  $y_{D_j}^{in}$  と  $y_{D_j}^{out}$  についても、 $x_{D_j}$  や  $z_{D_j}$  と同様、粒度レベルによる差異を加味する必要があるが、ここでの展開においては省略する。

**3.2 各設計局面における操作** 2.2項に示した設計局面と設計知識の内容は、上記の一般化部分問題における操作という観点から、以下のようにとらえ直すことができる。

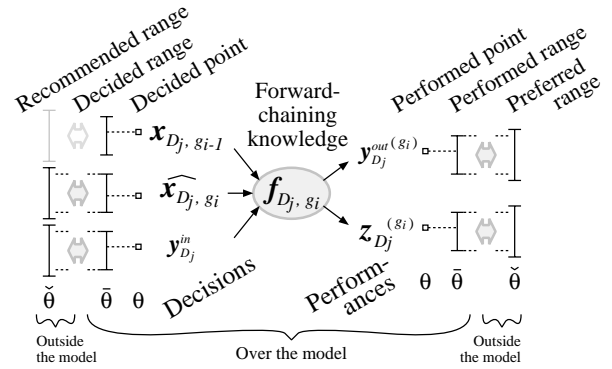


Fig. 1 Generalized partial model and related goals

初期設計 (initial design) … 操作可能な決定変数  $\widehat{x_{D_j, g_i}}$  の値を初期設定する。

前向き操作 (verification / forward-chaining) … 性能変数  $z_{D_j}^{(g_i)}$  と媒介出力変数  $y_{D_j}^{out(g_i)}$  の値を算出する。

評価 (critique) … 求めた  $z_{D_j}^{(g_i)}$  の内容について、評価知識や付随するゴールに照らして、何らかの判断を行う。

後向き操作 (refinement / backward-chaining) … 評価結果に問題がある場合には  $\widehat{x_{D_j, g_i}}$  の内容を調整する。

移行 (shift) … 評価結果が妥当であれば、他の粒度レベルに推移するなどして、設計を進展させる。

以上のうち、初期設計と前向き操作は何らかの明示的な知識に従って行えるものと期待することができる。評価に関しては、設計目標やゴールに対する照合は評価知識に従って行えるが、その結果のもとでの移行か後向き操作かという判断は意思決定の範疇に属するものであり、設計者に委ねられるべきものである。後向き操作に関しては、最終的な決定は設計者に委ねられるが、それに向けての後向き知識が明示的に存在することは一般には期待できないため、前向き知識に基づいた何らかの支援が必要である。一方、移行については機械的な操作として処理することができるはずである。つまり、各設計操作のなかでも、後向き操作に対する支援が本質的であるということになる。

**3.3 一般形のもとでのゴール** 第1報<sup>(1)</sup>で導入したゴールの考え方は性能変数と媒介変数に対するものであったが、式(1)の関係に従えば、左辺にあそびを認める上で右辺にもあそびを認める必要があり、決定変数についてもあそびとしてのゴールを定義する必要がある。図1は一般化部分問題からみた一連のゴールの状況を示したものである。ある変数  $\theta_\ell$  に着目するとき、そのゴールについての情報を、図1にも示すように、以下の内容をまとめた単位として定義し直す。

決定値・性能値 (decided point・performed point) … その変数の値として対象モデルに与える値, あるいは, それから算出される値.  $\theta_\ell$  と表す.

決定範囲・性能範囲 (decided range・performed range) … その値が取り得る範囲として対象モデルに与える範囲, あるいは, それから算出されるその値が取り得る範囲.  $\bar{\theta}_\ell = [\theta_\ell^L, \theta_\ell^U]$  と表す.

推奨範囲・選好範囲 (recommended range・preferred range) … 対象モデルには関与せず, その外部から推奨されたり選好されたりするその値が取るべき範囲.  $\check{\theta}_\ell = [\check{\theta}_\ell^L, \check{\theta}_\ell^U]$  と表す.

なお, 性能値と性能範囲については各粒度レベルのもとで異なる算出値(範囲)が存在することになる. また, すべての変数に関して, ゴールの内容について以下の各関係式が成立するべきであると考えられる.

$$\theta_\ell \in \bar{\theta}_\ell \quad (2)$$

$$\bar{\theta}_\ell \subset \check{\theta}_\ell \quad (3)$$

なお, 理想的な設計展開においては,  $\check{\theta}_\ell$  と  $\bar{\theta}_\ell$  の範囲の大きさは段階的に縮小されていくことにより,  $\theta_\ell$  の値も最終的に確定することになる.

**3.4 後向き操作についての一般形** 3.1項に示した一般形は順問題としての記述であり, それを 3.2項における後向き操作という局面に展開するためには逆問題としての形式に置き換える必要がある. その際の逆問題を規定する条件として以下のものを考える.

- (i) 評価知識に照し合せて性能変数  $z_{D_j}^{(g_i)}$  の内容を適切なものにする. これについては, 一般に, 目的関数  $|z_\ell^{(g_i)} - z_\ell^{obj}|^2 \rightarrow \min$  と不等式制約条件  $h_\ell(z_{D_j}^{(g_i)}) \leq 0$  として規定されるものとする. なお,  $z_\ell^{obj}$  は  $z_\ell$  についての目標値である. 前者は, 性能範囲に基づけば, 次式により表すことができる.

$$\left| \frac{\max_{\substack{x_{D_j, g_i} \in x_{D_j, g_i}^U \\ y_{D_j}^in \in y_{D_j}^in}} z_\ell^{(g_i)} + \min_{\substack{x_{D_j, g_i} \in x_{D_j, g_i}^L \\ y_{D_j}^in \in y_{D_j}^in}} z_\ell^{(g_i)}}{2} - z_\ell^{obj} \right|^2 \rightarrow \min \quad (4)$$

また, 後者も次式により表すことができる.

$$\forall x_{D_j, g_i} \in x_{D_j, g_i}^-, \forall y_{D_j}^in \in y_{D_j}^in, h_\ell(z_{D_j}^{(g_i)}) \leq 0 \quad (5)$$

- (ii) 現時点での決定変数の値が例えば初期設計知識に基づくなどにより大局的には望ましい方向にあるとの前提のもとで, それらからの変更をできる

だけ抑制する. これについては, それぞれの変数  $\widehat{x_{D_j, g_i}^\ell}$  をその現在値に対してできるだけ拘束することに対応することから, 式 (2) のもとでその決定範囲について以下の目的関数を導入する.

$$\left| \widehat{x_{D_j, g_i}^\ell}^U - \widehat{x_{D_j, g_i}^\ell}^{cur} \right|^2 + \left| \widehat{x_{D_j, g_i}^\ell}^L - \widehat{x_{D_j, g_i}^\ell}^{cur} \right|^2 \rightarrow \min \quad (6)$$

なお, 上付添字の  $^{cur}$  はある変数  $\cdot$  の現時点での値を表すものとする.

- (iii) 大域モデルに含まれる決定変数が大域モデルに基づく推奨範囲を満たす. この条件は領域モデルにおける推奨範囲を調整することによって式 (3) に帰着することができる.
- (iv) 媒介出力変数  $y_{D_j}^{out(g_i)}$  については, 関連している各領域モデルから推奨されるすべての選好範囲を満たすとともに, それらの各領域モデルでの設計自由度を確保するために対応可能なゴールの範囲をできるだけ拡大する. 前者は式 (3) に帰着ができ, 後者は定まる性能範囲を広く保つことに対応することから, それぞれの変数  $y_{D_j}^{out(g_i)}_\ell$  について以下の目的関数として記述する.

$$\left| y_{D_j}^{out(g_i)}_\ell^U - y_{D_j}^{out(g_i)}_\ell^L \right| \rightarrow \max \quad (7)$$

- (v) 媒介入力変数  $y_{D_j}^in$  については, 関連している各領域モデルから通知されるすべての推奨範囲内で自身における範囲とその中での代表値を定める. この条件は式 (2) と式 (3) そのものである.
- (vi) 定められた期間内に設計を終了するように, 各ゴールにおける範囲の幅を縮小していく. この条件も式 (3) に帰着することができる.

**3.5 最適化問題としての定式化** ゴール概念のもとでの後向き操作に向けての支援機能として, 式 (1) における  $\widehat{x_{D_j, g_i}^\ell}$  と  $y_{D_j}^in$  の決定値を定めるのではなく, 上記の各内容のもとでの可能な決定範囲を定めることにし, 最終的な値としての決定は設計者に委ねることにする. 範囲を決定する問題は以下の最適化問題として定式化することができる.

設計変数:  $\widehat{x_{D_j, g_i}^\ell}^L, \widehat{x_{D_j, g_i}^\ell}^U, y_{D_j}^in^L, y_{D_j}^in^U$  の各変数の値.

制約条件:  $\widehat{x_{D_j, g_i}^\ell}, z_{D_j}^{(g_i)}, y_{D_j}^in, y_{D_j}^{out(g_i)}$  の各変数について与えられる推奨範囲や選好範囲のもとでの式 (3) の内容に対応する不等式制約条件. 式 (5) の不等式制約条件.

目的関数: 式 (4)・式 (6)・式 (7) の内容が目的関数に相当するが, 多目的最適化問題となるため, 各

式の項をそれぞれ  $\vartheta_\ell^1, \vartheta_\ell^2, \vartheta_\ell^3$  とするとき、線形加重和により以下の目的関数を定める。

$$w_1 \sum_\ell k_\ell^1 \vartheta_\ell^1 + w_2 \sum_\ell k_\ell^2 \vartheta_\ell^2 - w_3 \sum_\ell k_\ell^3 \vartheta_\ell^3 \rightarrow \min \quad (8)$$

ここで、 $w_1, w_2, w_3, k_\ell^1, k_\ell^2, k_\ell^3$  はそれぞれ、重み付けのための正の定数である。

なお、以上のような範囲を用いた最適化は妥協解を求めるためのゴールプログラミング<sup>(4)</sup>でも用いられる。

### 3.6 線形化と単調性解析による一般形の展開

上記の最適化問題を解くに当って、式(1)における  $f_{z_j, g_i}$  や  $f_{y_j, g_i}$  の内容は一般には大きな計算コストが必要であると考え、最適化計算においては、それらの式を直接用いるのではなく、その時点における決定値や性能値の周辺での線形化近似を用いることにする。つまり、式(1)を次式に置き換える。

$$\left. \begin{aligned} z_{D_j}^{(g_i)} &= f_{z_j, g_i}^{cur} + \left\{ \frac{\partial f_{z_j, g_i}}{\partial x_{D_j, g_i}} \right\}^{cur T} (\widehat{x_{D_j, g_i}} - \widehat{x_{D_j, g_i}}^{cur}) \\ &\quad + \left\{ \frac{\partial f_{z_j, g_i}}{\partial y_{D_j}^{in}} \right\}^{cur T} (y_{D_j}^{in} - y_{D_j}^{in, cur}) \\ y_{D_j}^{out(g_i)} &= f_{y_j, g_i}^{cur} + \left\{ \frac{\partial f_{y_j, g_i}}{\partial x_{D_j, g_i}} \right\}^{cur T} (\widehat{x_{D_j, g_i}} - \widehat{x_{D_j, g_i}}^{cur}) \\ &\quad + \left\{ \frac{\partial f_{y_j, g_i}}{\partial y_{D_j}^{in}} \right\}^{cur T} (y_{D_j}^{in} - y_{D_j}^{in, cur}) \end{aligned} \right\} \quad (9)$$

この式は議論の簡便のために次式として書き直す。

$$\begin{bmatrix} z_{D_j}^{(g_i)} \\ y_{D_j}^{out(g_i)} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \widehat{x_{D_j, g_i}} \\ y_{D_j}^{in} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (10)$$

なお、線形化のための1次微分は近傍での数値微分や設計履歴に対する応答曲面近似などによって求めることができるものとする。

このとき、 $z_{D_j}^{(g_i)}$  や  $y_{D_j}^{out(g_i)}$  は3.4項の各内容と3.5項の定式化の随所に現れるが、設計変数を値ではなく範囲としていることから、それらの決定値に対する性能値ではなく、決定範囲に対する性能範囲が必要となる。この範囲から範囲への写像は元来は組合せ的なものであって、一般には容易に計算することはできないが、線形化によって得られる式(10)に単調性解析(monotonicity analysis)<sup>(5)</sup>を適用することにより、範囲から範囲への写像における特定の道筋を事前に想定する。すなわち、式(10)の特定の行に含まれる  $z = [a_1 \ a_2 \ \cdots \ a_n] [x_1 \ x_2 \ \cdots \ x_n]^T + b$  に着目するとき、各変数  $x_i$  の取り得る範囲が  $[x_i^L, x_i^U]$  であれば、その  $z$  の下限値  $z^L$  と上限値  $z^U$  は次式により定めること

ができる。

$$\left. \begin{aligned} z^L &= [a_1 \ a_2 \ \cdots \ a_n] [x_1^{*L} \ x_2^{*L} \ \cdots \ x_n^{*L}]^T + b^L \\ z^U &= [a_1 \ a_2 \ \cdots \ a_n] [x_1^{*U} \ x_2^{*U} \ \cdots \ x_n^{*U}]^T + b^U \end{aligned} \right\} \quad (11)$$

ただし  $x_i^{*L}, x_i^{*U}$  ( $i = 1, 2, \dots, n$ ) は次式により定める。

$$x_i^{*L} = \begin{cases} x_i^L & (a_i \geq 0) \\ x_i^U & (a_i < 0) \end{cases}, \quad x_i^{*U} = \begin{cases} x_i^U & (a_i \geq 0) \\ x_i^L & (a_i < 0) \end{cases}$$

以上により、線形化と単調性解析による展開により得られる  $z_{D_j}^{(g_i)}$  や  $y_{D_j}^{out(g_i)}$  についての計算式を用いれば、前項の内容が簡便に計算できるようになる。例えば、式(4)は  $\left| \frac{z_{\ell}^{(g_i)U} + z_{\ell}^{(g_i)L}}{2} - z_{\ell}^{obj} \right|^2 \rightarrow \min$  として、式(5)においては関数  $h_{\ell}(\cdot)$  を式(9)により線形化した式の係数に対して式(11)の単調性解析を適用することにより、簡便な計算式を得ることができる。

**3.7 最適化計算による方向付け** 前項の変形を通じて3.5項に示した定式化は2次計画問題となり、例えば双対法<sup>(8)</sup>を用いればその解を求めることができる。得られる結果は後向き操作に向けて設定された一連のゴールのもとでの妥当な決定の範囲を指し示すものであることから、設計者にとって有効な支援になるものと期待することができる。

なお、複合領域の最適化では、大規模な問題を分散的に扱ったり<sup>(6)</sup>、それらの間で目標レベルの相互調整を行ったり<sup>(7)</sup>する試みも行われているが、ここでの最適化法の援用は想定している問題の規模や利用における非同期性などの面でそれらとは異なるものである。

以下の各節では、以上の数理的な枠組みに基づいた具体的な設計操作の方法について論じることとする。

## 4 大域モデルに基づく操作

**4.1 大域モデルの内容と役割** 前述のように、大域モデルは設計問題において唯一存在して、すべての設計の起点となるものである。ここではそれが以下のものによって構成されているものとする。

大域決定変数 … 決定変数のうち、大域モデルに含まれるもの。  $x_G$  と表す。

大域性能変数 … 性能変数のうち、大域モデルに含まれるもの。  $z_G$  と表す。

要求仕様 … 設計対象などの使用条件などの  $z_G$  についての設計目標。  $z_G^{spec}$  と表す。

大域決定変数初期設計知識 …  $z_G^{spec}$  のもとで、 $x_G$  の初期値を定めるための知識。

大域前向き知識 … 定めた  $x_G$  の振舞いとしての  $z_G$  を粗く推定するための知識。

大域評価知識 … 推定した  $z_G$  が  $z_G^{spec}$  などに照し合せて適切であるかどうかを判断するための知識。

**4.2 初期設計のための処理** 上記の大域決定変数初期設計知識は、一般には、過去の設計データなどによる経験式やマクロな物理モデルを展開して得られる簡易式であり、個別の設計問題に依存してそれぞれに整備されていて、大域決定変数についての適切な値を導き出すものではある。しかしながら、具体的な領域モデルによる設計を進めるに先立って、特定の設計対象に照し合せて各種の設計目標間でのバランスを調整しておくことは、全体として優れた設計結果を得るためには不可欠なタスクである。

この初期設計における問題は、式 (1) に対応させれば、次式の形式を取る。

$$z_G = f_G(x_G) \quad (12)$$

前節での内容をこれらの両式の形式的な差異に従って翻訳すれば、上記の調整に向けての支援機能とすることができる。その際、大域モデルでの各変数の意味は方向付けのためのものであって、実際の内容はそれぞれの領域モデルにおいて確定されるため、 $x_G$  については決定値ではなく決定範囲に重点を置いた処理を行うことになる。

**4.3 領域モデルへの展開** 大域モデルにより想定された設計の方向は各領域モデルに展開されるが、その際、各変数についてのゴールを以下のように設定する。すなわち、領域モデル  $D_j$  においては、 $\check{x}_{D_j} \subset \bar{x}_G$  を満たすように推奨領域を設定し、また  $\check{z}_{D_j} \subset \bar{z}_G$  を満たすように選好領域を設定する。ただし、領域モデルにおいて後者の関係を満たすことができない場合には、大域モデルの側の  $\bar{z}_G$  を調整する必要を生じる場合もある。このことが大域的協調を要請する。

**4.4 大域的協調のための処理** 上記の内容を踏まえば、大域的協調とは、事前に想定した設計全体についての方向付けが具体的な設計展開に伴って崩れつつある場合に、それを適切な方向に補正するためのものである、と言える。その際の操作には 4.2 項の後半に示したもので対応可能であるが、方向付けにずれを生じる一因は式 (12) の忠実度が不十分であることから、そのずれを吸収することが必要となる。

いま、 $x_G$  と  $z_G$  についての各領域モデルでの決定値と対応する性能値を  $x_G^{cur}$  と  $z_G^{cur}$  とするとき、上記のずれのもとでは  $z_G^{cur} \neq f_G(x_G^{cur})$  となる。 $f_G(\cdot)$  が示す傾向は信頼するものとしつつ、このずれを吸収するために、式 (12) をその都度次式で置き換えることにする。

$$z_G = f_G(x_G) + (z_G^{cur} - f_G(x_G^{cur})) \quad (13)$$

これを用いて大域決定変数や大域性能変数についてのゴール情報を更新し、それを各領域モデルへと展開することによって、大域的協調に向けての方向付けを行うことができる。

## 5 領域モデルに基づく操作

**5.1 領域モデルの内容と基本的な操作** 領域モデルの内容については、3.1項に示したとおりであるが、4.1項の場合と同様に列挙すれば、(領域固有) 決定変数、(領域関連) 媒介変数、(領域固有) 性能変数、(領域関連) 要求仕様、(領域固有) 決定変数初期設計知識、(領域固有) 前向き知識、(領域関連) 評価知識、から構成されることになる。また、3.2項に列挙した各操作についても、それぞれ、3節に示した方法で進めることができる。以下では、なかでも、後向き操作において 2.4項に示した 3 つの協調処理が実現される仕組みについて示す。

**5.2 大域モデルとの関連** 大域決定変数  $x_G$  は、大域モデルに含まれると同時に、対応するそれぞれの領域モデルにも含まれている。つまり、領域モデルは各種の領域固有の決定変数をそれらに併せることによって対象部分のモデルを表現する。このことから、大域モデルにおける情報は大域的な協調に向けての補助的なものであって、 $x_G$  の各値の決定権は領域モデルの側にあるものとする。なお、 $x_G$  に含まれる決定変数は、領域モデル内では粒度レベルの最も粗い変数  $x_{D_{j,81}}$  に含まれる。そのもとで、大域モデルにおけるゴール情報は領域モデルでは 4.3項に示した意味を持ち、領域モデルでの設計を方向付けることになる。しかしながら、その方向付けが万全であるとは限らないため、領域モデルに基づいて展開される決定値や決定範囲についての情報は随時、大域モデルとも照し合せて、場合によっては、適宜 4.4項の方法により方向付けを補正する必要も生じる。

**5.3 領域間協調に向けての操作** 領域モデルとは対象システムの全体を何らかの基準に従って複数の部分に分割した際の各部分の内容であり、それらの相互干渉についての経路が媒介変数である。このとき、ある媒介変数を介した 2 つの領域モデルの間では、一方の領域における媒介出力変数はもう一方の領域では媒介入力変数になっている。つまり、3.3項で定義したゴールのもとでは、一方の領域における媒介出力変数  $\theta_\ell^{out}$  についてのゴールは、もう一方の領域における媒介入力変数  $\theta_\ell^{in}$  についてのゴールに結び付いている。したがって、決定における自由度と結果における自由度との関係を踏まえば、 $\theta_\ell^{out} := \theta_\ell^{in}$ 、 $\theta_\ell^{out} := \bar{\theta}_\ell^{in}$ 、

$\theta_\ell^{in} := \theta_\ell^{out}$  という代入関係によって相互のゴールを連結することができ、それによって領域間の協調を操作することができる。

#### 5.4 領域内での粒度レベルの遷移における操作

一方、個別の領域モデルの内部には複数の粒度レベルにおける対象モデルが重複して保持されている。

仮に、式(1)の粒度レベル  $g_i$  の対象モデルに対して、その下位に位置する一段細かい粒度レベル  $g_{i+1}$  の対象モデルは以下のように記述することができる。

$$\left. \begin{aligned} z_{D_j}^{(g_{i+1})} &= f_{z_j, g_{i+1}}(x_{D_j, g_i}, \widehat{x_{D_j, g_{i+1}}}, y_{D_j}^{in}) \\ y_{D_j}^{out(g_{i+1})} &= f_{y_j, g_{i+1}}(x_{D_j, g_i}, \widehat{x_{D_j, g_{i+1}}}, y_{D_j}^{in}) \end{aligned} \right\} \quad (14)$$

なお、 $\{x_{D_j, g_i}\} = \{x_{D_j, g_{i-1}}\} \cup \{\widehat{x_{D_j, g_i}}\}$  である。

上記の2つの粒度レベルの間で  $g_i$  から  $g_{i+1}$  へ移行する場合には、 $\widehat{x_{D_j, g_i}}$  を含めて  $x_{D_j, g_i}$  を固定し、操作の対象を  $\widehat{x_{D_j, g_{i+1}}}$  に移す。このとき、各変数についての推奨範囲や選好範囲は基本的には現在のものを継承することにする。

一方、 $g_{i+1}$  から  $g_i$  へ移行する場合には、大域モデルが領域モデルでの設計結果を踏まえる場合と同様、 $g_i$  での  $f_{z_j, g_i}$  と  $f_{y_j, g_i}$  の内容にずれが生じることになる。これについては、式(13)と同様、式(1)をその都度、次式で置き換えることにする。

$$\left. \begin{aligned} z_{D_j}^{(g_i)} &= f_{z_j, g_i}(x_{D_j, g_{i-1}}, \widehat{x_{D_j, g_i}}, y_{D_j}^{in}) \\ &\quad + (z_{D_j}^{(g_{i+1})cur} - f_{z_j, g_i}(x_{D_j, g_i}^{cur}, y_{D_j}^{incur})) \\ y_{D_j}^{out(g_i)} &= f_{y_j, g_i}(x_{D_j, g_{i-1}}, \widehat{x_{D_j, g_i}}, y_{D_j}^{in}) \\ &\quad + (y_{D_j}^{out(g_{i+1})cur} - f_{y_j, g_i}(x_{D_j, g_i}^{cur}, y_{D_j}^{incur})) \end{aligned} \right\} \quad (15)$$

その上で、操作の対象を  $\widehat{x_{D_j, g_{i+1}}}$  から  $\widehat{x_{D_j, g_i}}$  に移す。この場合も、各変数についての推奨範囲や選好範囲は基本的には現在のものと継承することにする。

## 6 ゴールの収斂に向けて設計プロセスの管理

**6.1 時間限定性とプロセス管理** 大規模なシステムの設計問題における規模性に起因する特質は、可能な限り時間を投入して最適な設計解を求めるのではなく、あらかじめ想定された時間内に何らかの満足な設計解を求めるということに象徴されている<sup>(1)</sup>。分散協調型設計に向けての本研究での考え方もそれに対応したものである。ここまでで示した設計操作のためのモデルと方法はそれに向けてのものであるが、ゴールにおける範囲を特定の設計解へと収斂させるためには、3.3項でも述べたように、分散協調的に展開される各設計操作に呼应させて  $\theta_\ell$  と  $\bar{\theta}_\ell$  との範囲を段階的に

縮小させていくように、設計プロセスを管理する必要がある。あわせて、領域モデル内での粒度レベルを適切に移行させ、設計を終了する時点ですべての決定変数の内容つまりは最も粒度の細かいレベルの内容が適切に確定しているように、設計プロセスを管理する必要もある。以下の各項では、両者の管理のためのスケジューリングについて論じる。

なお、何れの面に関してもスケジューリングのための絶対的な基準が存在しないため、第3報<sup>(2)</sup>での実装においては、何らかの方向性は提示するものの、プロセスの管理は設計者の判断に委ねることにしている。

### 6.2 粒度レベルの移行についてのスケジューリング

粒度レベルの移行を考える場合、粗いレベルでは設計操作の自由度が大きく大域的に設計解を探索できる一方で、設計解を明確に定めるためには細かいレベルでの設計操作が不可欠であることを踏まえる必要がある。粒度レベルによる多重性を設計プロセスにおける段階性に対応させた上で、設計の初期段階で十分な検討を行う必要があるというコンカレントエンジニアリングにおける方針<sup>(9)</sup>に従えば、粗いレベルでの操作を十分に繰り返すことにより概略についての良好な解を得ておくことが良好な最終解に結び付くということにはなるが、細部における検討が不足するようでは本も子もなくなることにもなる。つまり、レベル移行についてのスケジューリングは戦略に関わる内容であると言わざるを得ない。しかしながら、設計終了時にはすべての粒度レベルにおける設計が適切に完了している必要があるということ踏まえれば、設計終了時刻から逆算した各粒度レベルにおける設計についての最遅開始時刻をもとにして設計者に対する何らかの支援機能を構成することは可能であると考えられる。

### 6.3 ゴール範囲の縮小についてのスケジューリング

ゴール範囲の大きさに関しては、それが上記の粒度レベルの移行と関連して、ゴール範囲が広い段階では粒度レベルも粗く、ゴール範囲が狭くなるにつれて粒度レベルも細かいものに移行するものと想定することには、それほど無理はない。したがって、探索における大域性(とそれに向けての効率的な設計操作)と結果における局所的な質(とそれに向けての適切な忠実度)との間にはある種のトレードオフの関係が成り立つことを踏まえれば、このスケジューリングに関しても戦略に関わる内容であると言わざるを得ない。しかしながら、最終的には範囲を値にまで狭めることが必要であることを考えれば、現時点での範囲と終了時刻までの時間を加味して単位設計操作に希求される範囲の縮小率を提示していくことは、設計者に対する支援



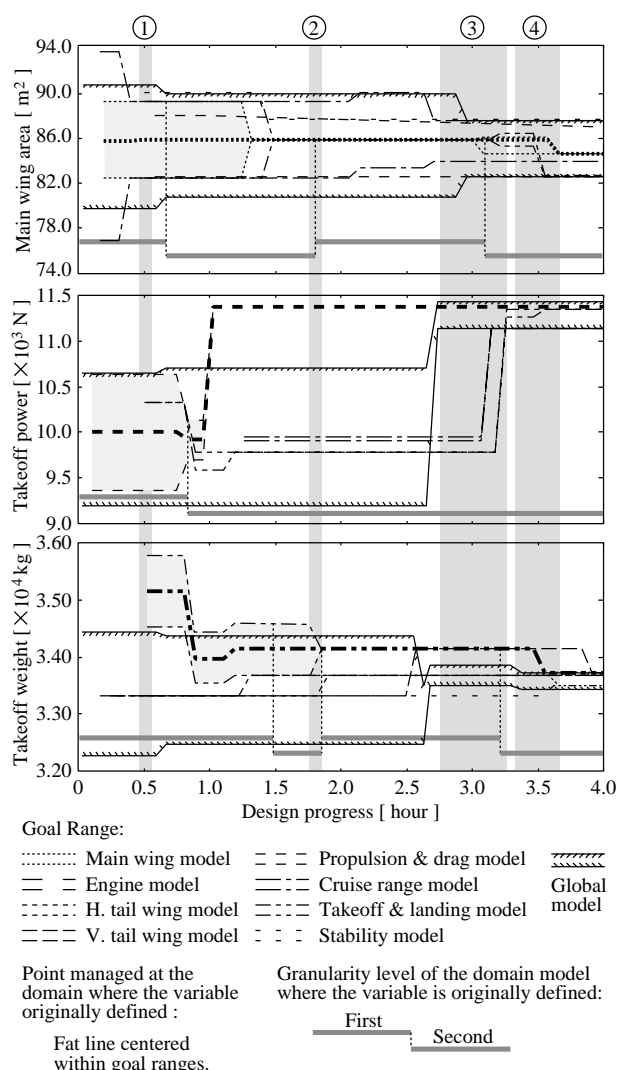


Fig. 2 An example of goal convergence history

機能として有効であるものと考えられる。

## 7 航空機の基本設計における協調処理の実施例

最後に、本研究で提案するプロセスモデルと操作モデルをもとに構成した設計支援システムを航空機の基本設計に展開した事例<sup>(2)</sup>における協調処理の状況を示すことにする。事例におけるプロトタイプシステムにおいては、対象モデル<sup>†2</sup>を、胴体、主翼、水平尾翼、垂直尾翼、エンジン、揚力・抵抗、航続性能、離着陸性能、安定性、の9つの領域モデルに分割しており、大域モデルを含めた合計10個のモデルによる設計処理が分散並行的に行われることになる。

図2は、質量と揚力のバランスに関わる3つの変数について、それらのゴール情報の履歴を示したもので

ある<sup>†3</sup>。①の局面では、離着陸性能についてのモデルに従い、まず、離陸質量を算出した上で、離陸距離と着陸距離を適正なものとするのできる媒介入力変数、具体的には主翼質量や離陸推力の決定範囲、すなわち媒介出力変数の側からみた選好範囲を定めている。また、②の局面では、第2粒度レベルでの設計内容では主翼強度を確保することが困難であることが判明したことに基いて、主翼についてのモデルを第1粒度レベルのものに戻している。③の局面では、離陸推力に関して、エンジンについてのモデルから算出されるものが離着陸性能や航続性能についてのモデルから必要とされるものを大きく上回るという状況が継続しているため、離陸推力についての推奨範囲や選好範囲を大域モデルに基づいて調整している。この調整の結果は離陸推力の増加と離陸質量の増加が相殺され離着陸性能を適切なレベルに維持するものである。④の局面では、事前に想定した設計終了時刻が近づいたため、各領域モデルに基づいた設計展開を大域的に確認して、最終的な設計結果への方向付けを行っている。

## 8 結 言

本報では、分散協調型設計支援システムの構築に向けて、第1報<sup>(1)</sup>で抽出したプロセス構造モデルに基づいて、設計操作のためのモデル、数理モデルと最適化計算による支援機能を提案し、さらに、プロセス管理について論じた。また、航空機の基本設計における展開例<sup>(2)</sup>を通じて、それらの有効性を示した。ここでの内容は、分散協調型設計におけるすべての処理を支援できるまでには至っていないが、その本質的な内容を明確にしたことによるものであり、この方面の設計支援システムの展開に向けての基盤となるものである。

なお、本研究の一部は日本学術振興協会未来開拓学術研究推進事業(96P00702)の援助によるものである。

## 文 献

- (1) 藤田・菊池, 分散協調型設計支援システムのためのプロセスモデルと操作モデル(第1報: プロセス構造についてのモデル), 日本機械学会論文集C編, Vol. 68, No. 666, (2002), (論文番号 01-0551).
- (2) 藤田・菊池・南, 分散協調型設計支援システムのためのプロセスモデルと操作モデル(第3報: エージェント方式による実装と航空機の基本設計への展開), 日本機械学会論文集C編, (投稿中: 論文番号 01-0553).
- (3) 藤田・赤木, システム構造に着目したエージェント方式による分散並行型設計支援システムの構成方法, 日本機械学会論文集C編, Vol. 65, No. 630, (1999), pp. 813-820.

<sup>†2</sup> 対象モデルの具体的な内容は、一部ではあるが、第3報<sup>(2)</sup>での航空機の基本設計への展開において示す。

<sup>†3</sup> 履歴のうち、①と③の局面における具体的な前後関係は第3報<sup>(2)</sup>での航空機の基本設計への展開において示す。

- (4) Lee, S. M., *Goal Programming for Decision Analysis*, (1972), Auerbach Publishers.
- (5) Papalambros, P. Y. and Wilde, D. J., *The Principles of Optimal Design (Second Edition)*, (2000), Cambridge University Press, pp. 87-127.
- (6) Kroo, I. and Manning, V., Collaborative Optimization: Status and Directions, *Proceedings of 8th AIAA/USAF/-NASA/ISSMO Symposium on Multi-Disciplinary Analysis and Optimization*, (2000), Paper Number AIAA-2000-4721.
- (7) Kim, H. M., Michelena, N. F., Papalambros, P. Y. and Jiang, T., Target Cascading in Optimal System Design, *Proceedings of the 2000 ASME Design Engineering Technical Conferences*, (2000), Paper Number DETC2000/DAC-14265.
- (8) 茨木・福島, FORTAN77 最適化プログラミング, (1991), 岩波書店, pp. 87-132.
- (9) Ullman, D. G., *The Mechanical Design Process (Second Edition)*, (1997), McGraw Hill, pp. 2-6.