第2章　CFDによる最適設計の動向

2.1 はじめに

流れの制御と最適化は人類文明にとって遙か昔より重要な問題であった。いやむしろ、流れの制御と最適化は、生命の進化にとって重要な問題であったといってもよいかもしれない。栄養の補給と老廃物の除去、呼吸、血流など体内の流れに始まり、遊泳や飛翔など生物はあらゆる場面で流れを利用している。長い進化の歴史の中で、さらに生命はビーバーのようにダムを作り環境を操作する生物を産み出してきた。人類の科学技術は、環境に対する大きな影響力を誇り、ダムによって川の流れをせき止め、航空機によって何百人という人を一度に地球の裏側まで運んでいる。

しかし、このような技術は、これまで主に流体の支配方程式を解くことなしに（しかもしばしば適切に）処理されてきた。もちろんビーバーがナビエ・ストークス方程式を知っていると思えないが、たとえば人類の発明した航空機の制御すら、流体の方程式と連立して解かれるのではなく、前もって与えられた空力係数によって外力を評価する機体の運動方程式に基づいて構築されており、それでも航空機は安全に飛んでいる。

一方、最適性という概念は、物理学によって自然を理解する上で重要な概念である。例えば次のような抵抗最小化問題を考えてみよう：「回転体が一定速度でその軸方向に運動するとき、最も抵抗が最小となるような形状を求めよ。」いかにも流体力学の教科書にでてきそうな問題であるが、この問題を提出し、流れに対するある種の仮定の下でこの問題を変分法により実際に解いたのは、17世紀末のニュートンであった！

19世紀には流体の支配方程式であるナビエ・ストークス方程式が完成し、20世紀の流体力学の発展に伴い、超音速から亜音速へ衝撃波を伴わずに減速する遷音速翼型の設計や吸い込み・吹き出しによる境界層制御など、高度の流体力学の知識を駆使した設計・制御・最適化も見られるようになってきた。このような研究は、最適化の理論を用いることなく主に流体の物理的性質を調べることで進められてきた。たとえば遷音速翼型で、超音速から亜音速への減速時に衝撃波を回避できれば造波抵抗はほぼ零となり、そのような翼型は衝撃波を伴う通常の翼型に比べはるかに抵抗が小さい。しかもそのような翼型の存在する条件が特殊なため、一つでも見つけられればそれが最適解であると見なされてきた。

航空機の制御に見られるような「流体なしの最適制御」・流体力学の理論的実験的研究に見られるような「最適化なしの流体設計」に対して、流体の支配方程式に基づく最適化問題の定式化を考えることができる。このような問題の定式化がこれまでされてこなかったのは、ひとえに流体の方程式を解くのが大変だったためである。

現在では、数値流体力学（CFD）によって流体の支配方程式が数値的に解けるようになってきたので、流体の支配方程式の基づく最適化が注目されている。また、CFDが設計現場で受け入れられるにつれ、性能向上・設計期間短縮など設計最適化への期待が高まっている。このような機運を受けて、CFDと最適化法を組み合わせた研究が最近学会などでも発表されるようになってきた。もっとも、現在でもまだリアルタイムでの非定常流体力の計算は大変なので、流体問題の最適化ではオフラインで行える形状最適化にもっぱら関心が集まっている。本稿では、CFDによる数値最適化について、いくつかの基礎的な話題を紹介しよう。

2.2 CFDソフト＋最適化ソフト＝流体問題最適化？

ところで、最適化の研究といえば、どのような最適化手法を用いるのかということに話題が集中しがちであるが、CFDと最適化法を組み合わせるというアイデア自体誰しも考えつくことである。CFDと最適化

アルゴリズムはいずれも必要な技術ではあるが、それが問題の本質だろうか？

2.3 問題設定について

まず問題設定について考えてみよう。設計変数の設定では、設計変数とその値の範囲を決めた時点で、解を求める設計空間を決めてしまったことになる。設計空間の大きさは問題の解きやすさ・結果の解釈のしやすさに直に関わっているので、なるべく小さな次元にとどめたい。しかし、本来最適であるべき答えがその設計空間に含まれていなければ、最適化をしても求める解は得られない。たとえば形状最適化を行う場合、3 次元形状は無限の自由度を持っている。しかし、実際に制作できる形状はコストも考えて比較的単純なものに限られる。本来無限の設計空間から有限の「比較的単純」な形状をどのように切り出してくるかは、決して自明な問題ではない。

目的関数の決定もまた自明なようでそうともいえない。たとえば航空機の形状最適化を考える場合、主翼形状の空力最適化は典型的な流体最適化問題である。しばしば用いられる目的に主翼の揚抗比を最大化するというものがあるが、実はこの揚抗比を目的関数に用いて望みの解を手に入れるのはかなり難しい。まず、揚抗比は翼の迎角の関数であり、ある迎角で局所最適解を与える。このため形状変化が翼の迎角変更にとどまってしまうことが多い。また、遷音速翼では造波抵抗も考慮しなければならないが、しばしば誘導抵抗とのトレードオフを持つ局所解にたどりついてしまう。このため、剥離・衝撃波なしで楕円荷重分布を持つ流体力学的に最適と考えられる解を求めるのは非常に難しい。すなわち、目的関数値がある変数にだけ強い感度を持つ・局所解が多数ある・関数値の成分にトレードオフがあるといった場合、最適化は困難であり、問題設定を考え直した方がよい。

このように最適化問題の本質は、実は最適化問題の定式化自体にある。何を設計変数とするのか、目的関数は何か、拘束条件は何か、これらを吟味した上で、得られた最適解はどのような意味を持っているのかを考える必要がある。

2.4 最適解について

次に、最適化によって得られた最適解をどう取り扱うか、3 つの立場を例にあげて考えてみよう。第 1 は、最適化によって以前考えもしなかったすばらしい設計ができることを期待する立場である。これは逆に言えば、何を最適化するのかよく分かっていないことを意味している。問題設定が十分吟味されておらず、このような状態でよい解が得られる見込みは、宝くじを買うようなものである。第 2 は、初期設計から多少の改善があれば良しとする立場である。このような立場は、問題設定というものが現実のモデル化であり、その解は多かれ少なかれ現実的ではないことを失念している。航空機の主翼を再び例に取れば、遷音速領域で抵抗を減らそうとすると大抵翼厚が減少する。ところが現実には、薄い翼で荷重を支えようとすると、構造重量やコストがかさむことになり、そのような翼は現実的ではない。これは空力最適化の際に構造の拘束条件を考慮しなかったためである。あるいは燃料タンクの容量が足りないかもしれない。脚の収納が、フラップの荷重が、と現実の問題では様々な拘束条件が現れるが、初めからそのすべてを考慮して適切な最適化問題を設定することは困難なことが多い。それでは得られた最適解は結局現実問題には役に立たないことになる。

最適解そのものを目的とする上記 2 つの立場に対し第 3 の立場として、最適解そのものを最終目的とするのではなく、最適解から実際の設計に役立つ情報を引き出すことを目的とする立場が考えられる。結果としての最適解を検討して問題設定に立ち返るのは重要なステップであるが、ここではそれをもう一歩進めて、最適化のプロセスから設計空間の構造を理解し実際の設計に役立つ情報を引き出すことを考える。最適解は所詮モデル上の最適解である。最適解そのものに大きな意味があるのではなく、最適解近

傍の設計空間の構造にもっと有用な情報が含まれていると考えられる。

たとえば最適解近傍の感度解析は、最適解のロバスト性に対して重要な情報を与える。製品生産では設計変数の値に常に誤差が混入されると考えられる。このとき、最適なパラメータからの変動によって製品の性能が大きく影響を受けることは実用上好ましくない。すなわち、目的関数に対する最適性より設計点におけるロバスト性が望ましいことが多い。その上、感度解析は設計変数の重要度も教えてくれる。拘束条件のパラメータが制御可能なときも同様である。

また、目的関数を単一目的のスカラー値関数ではなく多目的のベクトル値関数として定式化する場合は、単一の最適解を求めるのではなく「パレート解」と呼ばれる集合を求めることになる。パレート解とは、ある目的関数の値を改善するためには少なくとも他の 1 つの目的関数の値を改悪せざるを得ない非劣解であり、その名称は初めてこの概念の重要さを指摘したイタリアの経済学者 Pareto に因んでいる。多目的の場合でも、ある重み付けで複数の目的関数の和をとると、スカラーの目的関数を作ることができる。Fig. 2.1に示された2目的の最小化問題の例のように、重み付けされた単一目的最適解はパレート解の 1 点に対応しており、そのときの重み付けは目的関数間のトレードオフに対応している。ここで問題となるのは、通常あらかじめ目的関数間のトレードオフ情報があるわけではないので、どのような重み付けを与えれば望ましい解が求まるのか分からないことである。逆に、すべてのパレート解が求まって初めてトレードオフが定量的に評価できる。そこで、多目的最適化問題ではなるべく多数のパレート解を求めることが望ましい。この場合も 1 点の最適解を求めるのではなく、パレート解という集合を通して設計空間のトレードオフ情報を解析し、望ましい解を選択することになる。

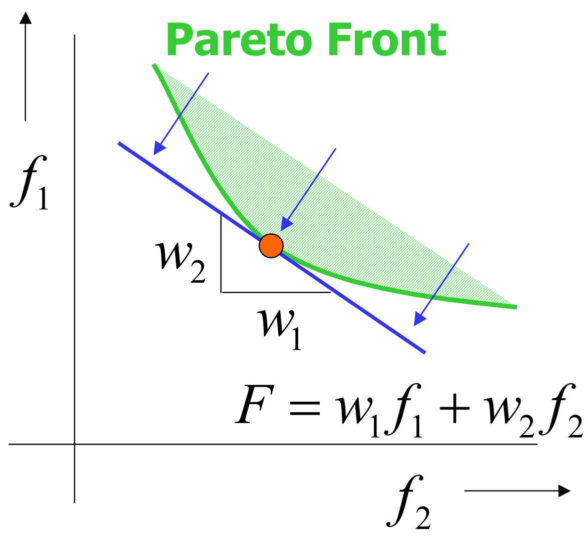

Fig. 2.1 パレート集合と重み付けによる単一最適解の関係

最適化問題におけるモデル化を現実問題の部分空間と見なすと、モデル化の仕方によって様々な部分空間がとれることになる。その部分空間ごとに固有の最適解があり、それらが一致する保証はどこにもない。一方、各部分空間における感度やトレードオフに関する大域的な情報は、もとの現実空間を推定するために役立つはずである。従って、モデル化された設計空間をよりよく理解すれば、実際の設計を改良する設計上の指針を与えることができる。そこで、第 3 の立場では、結果（最適解）そのものではなくその解釈が重視されることになる。

上の議論から最適化で大切なことは問題設定であり、また結果（最適解）そのものではなく結果の検討・解釈が重要であることが分かる。CFDソフト・最適化ソフトは見るべきポイントを探すために必要とされる技術に過ぎない。大切なことは何を見るのかという正しい問いを発することであり、とりもなおさず正しい

問題設定を行うことである。そのためには問題設定を繰り返し吟味し、結果の解釈を通じて設計空間の構造に関する情報を得ることが重要である。また、ここにこそ従来のCFD研究を越えた新しい研究開発の課題があると考えられる。

2.5 流体問題最適化法について

まず流体問題の最適化の数学的な背景について考え、次に具体的な最適化法について、最近の話題まで含めて紹介しよう。

2.5.1 数学的定式化

目的関数を $J(Q,g)$ (Q:流れの変数、g:設計変数)とする。流れの変数は当然流れの支配方程式を満たす必要があるので、流れの支配方程式 $F(Q,g)=0$ を考慮する必要がある。通常 J の値を求めるためにまず $F = 0$ を解いているのでこれが拘束条件であることを意識しないことが多いが、設計上の拘束条件がなくても流れの支配方程式が拘束条件となることに注意しよう。

最適解であることの数学的な条件を求めてみよう[1]。適当な内積を導入しラグランジュ係数 ξ を用いると、ラグランジュ関数

$$L(Q,g,\xi)=J(Q,g)-\langle F(Q,g),\xi\rangle \tag{2.1}$$

が定義できて、変分法によってこの最適解の条件が、

$$\text{流れの支配方程式:}\quad F(Q,g)=0 \tag{2.2}$$

$$\text{随伴方程式:}\quad \left(\frac{\partial F}{\partial Q}\right)^{*}\xi-\left(\frac{\partial J}{\partial Q}\right)^{*}=0 \tag{2.3}$$

$$\text{最適性条件:}\quad \left(\frac{\partial F}{\partial g}\right)^{*}\xi-\left(\frac{\partial J}{\partial g}\right)^{*}=0 \tag{2.4}$$

のように求まる。(この条件はあくまで局所最適解についての必要条件である。)この条件から最適解を求めようとすると、元の流体の支配方程式に比べ約 3 倍の連立非線形偏微分方式系を解かねばならない。問題設定に拘束条件をさらに付加するならば、ラグランジュ係数が増えることによってこの方程式系はさらに大規模になっていく。

一般に流体方程式すら解析的に解けないことを考えれば、通常の場合、流体の最適化問題を解析的に解くことはできない。また、流体問題は非線形であるために、目的関数の応答が複雑になり、一般に多峰性を持つ目的関数を扱う問題となる。多峰性があると局所的な最適解がいくつも現れるため、真の最適解すなわち大域的最適解をどうやって求めるかが問題となる。しかし、大域的最適解であることを判断するための条件はない。解の信頼性を確認するには、最適化を繰り返すことによって、異なる初期値から出発しても同じ解に到達することを確認するしかない。したがって、流体問題の最適化とは、最適解の「一候補」を何らかの方法で近似的に求めることにほかならない。

2.5.2 数値的最適化法

前節のような事情から我々は解析的にアプローチするよりも、何らかの方法で数値的に流体の最適化問題を解くことになる。流体の最適化問題に用いられている解法にはいろいろな分類の仕方があるが、まず決定論的な方法と確率論的な方法に大別できる。決定論的な方法は、多変数に対するいわゆる非線形計画法を適用するもので、計算効率がよい・局所最適化なら正確に求まるなどの利点がある。一方、確

率論的な方法には、Simulated Annealing 法(焼き鈍し法・SA)[2]、Evolutionary Computation(進化的計算、遺伝的アルゴリズム(GA)・進化戦略(ES)・遺伝的プログラミング(GP)・進化的プログラミング(EP)等の総称)[3]などがあり、大域的な最適解が求められる・勾配情報がいらない・関数値の誤差に強いなどの利点を持つ。ただし、計算時間がかかる、収束判定が曖昧であるなどの欠点を持っている。Fig. 2.2 に決定論的な勾配法(山登り法)とSA、Fig. 2.3 に進化的計算法が多峰性を持つ目的関数の分布の中で、最適解を探索する様子を図示する。

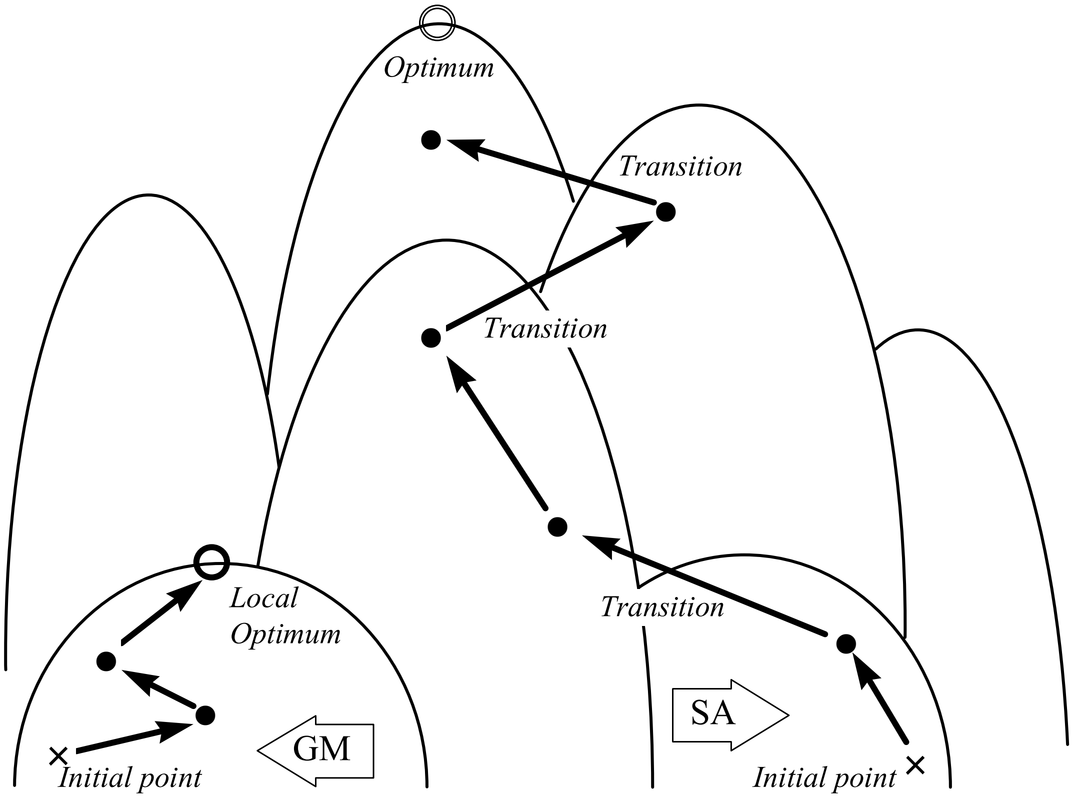

Fig. 2.2 勾配法(GM)と SA の作業状況

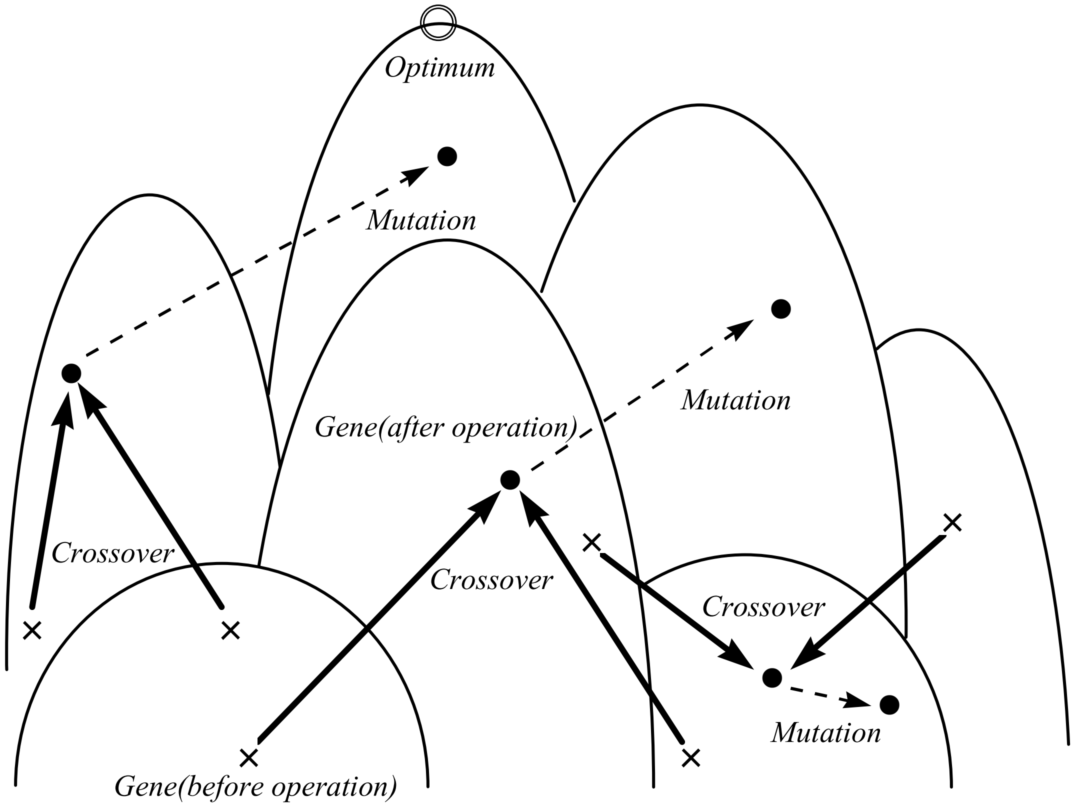

Fig. 2.3 進化的計算法の作業状況

2.5.3 決定論的最適化アルゴリズム

決定論的な方法はさらに、勾配情報（目的関数の微分）がいらないものと、それを利用するものに分けられる。最適化というとすぐに目的関数の微分を計算するものと思っている人もいるが、微分による勾配情報を利用する方法は最適化法全体の中では一つの手法に過ぎない。決定論的な手法の中で、微分による勾配情報を必要とせず手軽に結果を得られるもっとも良い方法として Nelder-Mead の Downhill simplex 法[4]が知られている。（ちなみに線形計画法の Simplex 法とはことなるので注意していただきたい。）プログラムも短く、高いお金を出さなくても手にはいるので、最適化を始めるにはお薦めの方法である。実際にドイツ・エアバス社でも設計に用いている。また、この方法は確率論的な手法であるSA法と容易にハイブリッド化できてより大域的な解を探せるようになるので、是非ライブラリに用意しておきたい。

微分による勾配情報を利用する場合には逐次 2 次計画法がよく用いられるが、勾配法は勾配計算の手法によってさらに分類される。1 つは、直接有限差分法によって $\partial J / \partial g$ を計算する方法である。この方法は設計変数や拘束条件の数に比例して差分計算の数が増えていくため、3 次元では計算効率が悪くなると考えられている。なお、Downhill simplex 法と直接有限差分法の計算効率はほぼ同程度である。これに対して、1 回の解析ですべての設計変数の勾配情報を計算できる画期的な方法として、Adjoint 法がある。この方法は上の随伴方程式を利用する。まず流れの支配方程式（2.2）を解く。その情報を利用して随伴方程式（2.4）を解きラグランジュ係数 ξ を決定する。そして最適性条件式（2.4）によって $\partial J / \partial g$ を計算する。こうして、得られた勾配情報を用いて非線形計画法を実行する。設計変数の次元がどんなに増えようと1度にすべての設計変数の勾配計算ができるというのは、多くの設計変数を必要とする 3 次元の複雑形状の設計に対してすばらしい利点である。ただし、目的関数や設計変数を変えるたびに定式化からコーディングまでやり直さなければならないという重大な欠点を持つ。

Adjoint 法は、偏微分方程式に基づく連続形で定式化するか、CFDで離散化された方程式に基づく離散形で定式化するかでさらに 2 つに分類される。連続形の場合、方程式自体の定式化は比較的簡単だが、乱流モデルの扱いなど、困難な点もある。この方法を使用するのは、同じ問題を繰り返し解くような非常に特殊なプロジェクトに限られるであろう。

一方離散形による Adjoint 法は離散化した流体の方程式を扱うが、さらに直接差分法と自動微分法に分けられる。直接差分法で自ら差分式を書いて勾配情報を作る手間をかけるより、コンピュータがやってくれた方が楽だし単純なミスも防げるので自動微分法を利用することが望ましい。自動微分法も大規模なコードを取り扱うにはまだ制約があるようだが、このアプローチにより将来性があるといえよう。

2.5.4 確率論的最適化アルゴリズム

この節では、確率論的な最適化アルゴリズムとして特に進化論を模擬した計算手法を考えてみよう。このような計算法には先に挙げたようにGA・ES・GP・EPといった手法があるが、これらの名称の区別は主として過去の発展の経緯を反映している。最近はこうした流派にこだわらず統一的な見方をしようという動きもあり、その一つの現れとして進化的（型）計算（Evolutionary Computation）という名称が Journal of Evolutionary Computation・IEEE Transactions on Evolutionary Computation のように学会誌の名称などに用いられるようになってきている。

一般的な進化的計算では、設計候補からなる集団を考え、その各個体に最適化の目的関数に応じた適応度を与える。適応度に応じて親となる個体を選び遺伝情報に交叉や突然変異を施して子を作る。親と子を適当に入れ替えることで世代を交代させる。再び適応度を与え、親を選び子を作り、新しい世代と入れ替える。こうして世代を進めることでより目的関数をよりよくするような設計候補を進化させるのである。また、進化のアルゴリズムに比べるとCFDの計算が圧倒的に重いので、集団の個体数分の関数評価（すなわちCFD計算）さえ並列化すれば効率よく計算できることも利点の一つである。

このような進化論に基づく手法のアイデアは 1960 年代や 70 年代に提案されたが、工学問題の最適化手法として注目されるようになったのは、80 年代にコンピュータがふんだんに使えるようになってきた時期と重なっている。なかでも、Goldberg による天然ガスパイプラインの制御や Powell らによる航空機エンジンの概念設計は、GAの成功例としてよく引用されている。Goldberg はその後文献[5]を書き、GAの発展に決定的な役割を果たした。今もこの分野のリーダーとして盛んに活動している。一方、Powell らの作ったソフト EnGENEous は、GEで実際の設計に適用され、そのエンジンが後にボーイング 777 に採用されたため特に有名である。

これらパイプラインの問題にしても航空機エンジンの問題にしても、流体の方程式は解いていないものの流体運動を最適化しているところが興味深い。さらに進化的計算の歴史をひもとくと ES もまた、ドイツで風洞実験によって翼の最適設計をするための自動設計手法として考えられたのがそもそもの始まりであった。このように、流体の支配方程式を解かなくても流体問題の最適化は非常に困難であったため、その困難さを克服する手段として確率論的な最適化法が登場し、今や流体問題以外の様々な分野でも広く用いられるようになったのである。

2.6 多目的進化的アルゴリズムとデータマイニング

2.5 節で一つの最適解ではなく、設計空間を表すより大域的な情報が必要な例として、多目的最適化をあげた。ここで必要とされる多数のパレート解を求めるために、従来の勾配法などの決定論的な方法では、結局のところ重み付けを変えながら異なる単一目的関数最適化問題の答えを、一つ一つ地道に求めていくしかなかった。しかし、多目的進化的アルゴリズムでは、多数のパレート解を一度にサンプリングでき

る。常に集団を用いて最適化計算を行う進化的計算法の特徴を活かし、集団の個体がパレート集合になるべく一様に分布するような遺伝的操作を加味しながら進化をさせることができるのである。（この手法の概要については、筆者の web ページを参考にしてほしい、流体研の URL、http://www.ifs.tohoku.ac.jp/の教官一覧より「大林 茂」をクリック。）

進化的計算では集団の個体数かける世代数分の関数評価を行う必要があるので、非常に多くの関数評価が必要となり、CFDと組み合わせた場合は膨大な計算時間がかかることになる。それなのにたった一つの最適解しか得られないのではあまりに計算効率が悪い。そこで、特に流体問題では、一度にたくさんのパレート解を求められる多目的進化的アルゴリズムを多目的の流体問題最適化へ適用することが一つのトレンドとなっている。

こうして、多目的進化的アルゴリズムで多数のパレート解を得ることができるようになると、別の問題が生じる。1つあるいはほんの数個の解を詳細に検討することはさほど手間ではないが、進化的アルゴリズムによって数百のパレート解が得られたとき、これらすべてを詳細に検討することはかなりの苦痛である。また、Fig. 2.1 のように目的関数空間でトレードオフを表示するにしても、目的関数が2つ3つのうちは容易に可視化が可能であるが、4つ以上になれば高次元空間の可視化を行うことになり、これも容易ではない。これはとりもなおさず、データ数が少ないうちは詳細な検討も可能であるが、データ数が膨大になったときにどのように意味のある解釈を引き出すかという、IT技術の今日的な課題に対応している。

この課題への今日的処方箋は、データベース化とデータマイニングである。その典型的な手法の一つとして自己組織化マップ[6]があげられる。Fig2.3 に、筆者らのグループで行った超音速翼多目的最適化問題のパレート解に対する自己組織化マップの例を示す。この問題では、超音速抵抗・遷音速抵抗・ベンディングモーメント・ピッチングモーメントの4目的最小化を、3次元翼を表現する72の設計変数を用い、ナビエ・ストークス方程式を用いて最適化した。その結果、766個のパレート集合が4目的関数空間における3次元曲面上の分布として得られた。この分布を2次元平面に表現したものが Fig. 2.4 である。このマップ上に、個々の目的関数の分布やその他の量の分布を示すことで、結果の解釈に役立てることができる。また、設計変数のマップを作ると、設計変数のクラスタリングができ、重要な変数やその組み合わせを発見することができる。このように、機械工学の分野でもデータマイニングは有効なツールである。

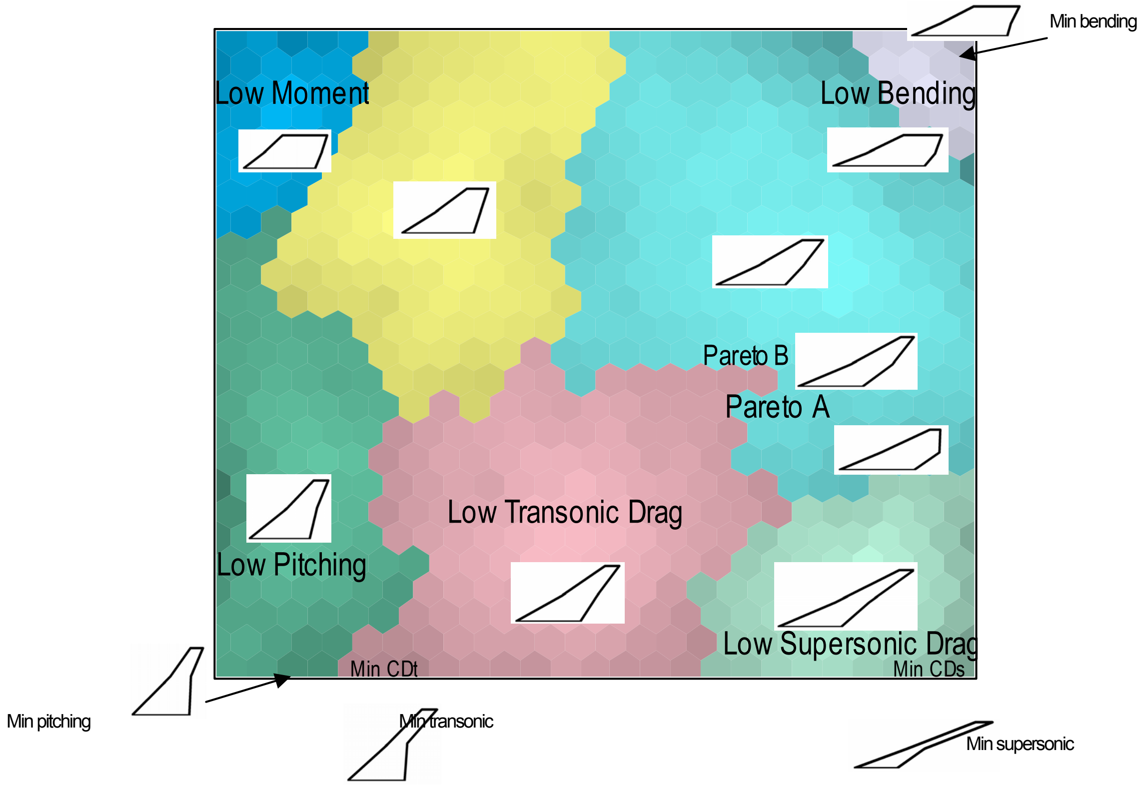

Fig. 2.4 超音速翼パレート解の自己組織化マップ

2.7 おわりに

本稿の中で出てきた流体以外のいくつかのキーワードを取り出してみると、自動微分・進化的計算法・多目的進化的アルゴリズム・データマイニング・自己組織化マップなど、流体問題最適化という分野は、情報科学、特にソフトコンピューティングの分野に関連深いことが分かる。すなわち、この分野は従来の流体工学と情報科学の学際領域に成立しており、今後流体工学と情報科学の融合を目指す「流体情報学」の重要な要素として発展していくと予想される。

また、流体問題の最適化がより実用的な問題に適用されるにつれ、この分野の研究開発は、流体としての単一分野の最適化にとどまらず、流体と構造・制御・推進など複合領域の最適設計に進んでいくものと予想される。このような統合化システムの設計最適化問題では、分野間のインターフェースをうまくとることが重要であり、IT技術の利用が期待される。IT革命を経た現在では、流体問題の最適化を流体分野に閉じて考えるのではなく、CFD技術を核とした統合化システムのシミュレーション技術の確立とその最適化を目指すことになろう。

現在享受している生活の快適性を損なうことなく人類の活動から地球環境を守るため、機械や機械からなる人工システムの性能を落とすことなく環境適合性を高めることは21世紀の機械工学の大きな課題である。高次の物理モデルに基づく流れの制御と最適化によって、人工システムのライフサイクルにわたる環境適合性を高めることが今後ますます重要になるであろう。このために、CFDとそれを利用した最適化技術が果たす役割は非常に大きい。

参考文献

[1] Gunzburger, M. D., “A Prehistory of Flow Control and Optimization,” *Flow Control*, pp. 185-195, Springer-Verlag, New York, 1995.

[2] Kirkpatric, S., “Optimization by Simulated-Annealing,”*Journal of Statistical Physics*, Vol. 34, No. 5/6, pp. 975-986, 1984.

[3] 伊庭 斉志, *遺伝的アルゴリズムの基礎*, オーム社, 東京, 1994.

[4] Press, W. H. *et al.*, “Minimization or Maximization of Functions,” *Numerical Recipes in FORTRAN : the art of scientific computing*, Cambridge University Press, Cambridge, pp. 387-448, 1992.

[5] Goldberg, D. E., “*Genetic Algorithms in Search, Optimization & Machine Learning*,” Addison-Wesley, Reading, MA, 1989.

[6] T.コホネン，*自己組織化マップ*, シュプリンガー・フェアラーク東京，東京，1996.