ラフ集合による多目的最適化データマイニング（改訂版）

Data Mining for Multi-Objective Optimization Using Rough Sets

大林 茂（東北大流体研）

Shigeru Obayashi, Institute of Fluid Science, Tohoku University, Katahira 2-1-1, Aobaku, Sendai, Japan

Data mining technique based on Rough Set theory has been applied to non-dominated solutions obtained from four-objective optimization for supersonic wing design. To reveal design tradeoffs, multiobjective optimization was performed by using Evolutionary Algorithms. High dimensional data (design variables and the corresponding objective function values) are mapped onto the two-dimensional Self-Organizing Map (SOM) where global tradeoffs are visualized. The rule sets are derived by Rough Set theory so as to determine the importance of design variables corresponding to the clusters obtained from SOM.

Key Words: *Design, Evolutionary Computation, Data Mining, Rough Sets, CFD*

１．はじめに

最適設計問題は、工学的にも重要な問題であるが、最適解そのものより、最適化のプロセスから設計空間について如何に価値のある情報を引き出すかが重要である。進化的計算法を用いた多目的最適化では、設計トレードオフの検討を通じて、設計空間の特徴をつかむことができる。進化計算は、多くの目的関数評価を必要とする一方、それによって生成された解は、計算シミュレーションによって構築された仮想的な設計データベースと見なすことができる。本研究では、この仮想設計データベースから、価値のある設計情報を引き出すために、ラフ集合によるデータマイニングの適用を検討する。

２．ラフ集合

ラフ集合（Rough Sets）[1,2]は、ファジィ・ニューロ・GAと並ぶソフトコンピューティングの構成分野として成長しつつあり、近年データマイニングの一手法として脚光を浴びている。我が国では感性工学の分野でよく用いられている[3]。ラフ集合はファジィ集合にも近い概念であるが、データマイニング法としては、集合要素の分類を通じて、特定の属性を満たすための決定ルールを生成することに特徴がある。

ラフ集合は集合論の記号と演算を用いて数学的に説明されるが、ここでは簡単な例[1]をもとにその考え方を説明する。

表１　車種選好の決定表

対象 U	条件属性 C			決定属性 D
車種	エンジン	サイズ	色	選好
x1	propane	compact	black	good
x2	diesel	medium	gold	bad
x3	diesel	full	white	bad
x4	diesel	medium	red	bad
x5	gasoline	compact	black	good
x6	gasoline	medium	silver	good
x7	gasoline	full	white	bad
x8	gasoline	compact	silver	good

この表では、対象が 8 つあり、条件属性集合 C={エンジン、サイズ、色}で特徴づけている。エンジンにはさらに{propane, diesel, gasoline}の３つの属性値がある。サイズ{compact, medium, full}、色{black, gold, red, silver, white}など属性値を選ぶことで、対象 U を様々な形に分類できる。また、この例では簡単のため一つだけの決定属性 D={選好}を考える。選好の属性値{good, bad}によっても、対象 U は二つに分類される。

一般に U を条件属性 C によって分類するとき、その分類が決定属性による分類と一致する保証はない。図 1 のようにある決定属性値を持つ集合 X を指定したとき、U の分類で X を覆うような「上近似」と、かならず X に含まれるような「下近似」に分かれる。「上近似」と「下近似」の違いが「ラフ」な近似を意味しており、ラフ集合という名前の由来となった。例として、エンジンの属性による分類を考えると、U の分類は、X1 = {x1}, X2 = {x2, x3, x4}, X3 = {x5, x6, x7, x8}となる。X として good な分類{x1, x5, x6, x8}を考えたとき、下近似は X1、上近似は X1 と X3、残りが X2 となる。

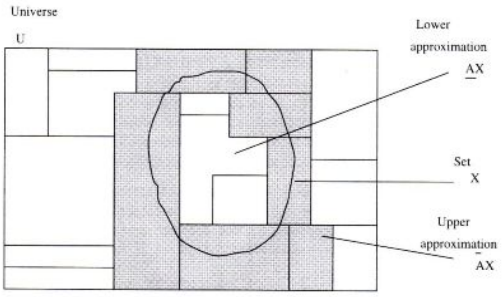

図 1　集合の分類と近似

条件属性による分類をした上で、決定属性による分類をする場合、条件属性集合 C={エンジン、サイズ、色}には冗長性があり、その部分集合｛エンジン、サイズ｝または｛エンジン、色｝をとっても最終的な分類は変わらない。具体例として、｛エンジン、サイズ｝による分類では、{{x1}, {x2,x4}, {x3}, {x5,x8}, {x6}, {x7}}となり、good な分類は{x1, x5, x6, x8}である。前者のうち後者の部分集合（下近似）となっているものは、{x1}, {x5,x8}と{x6}である。すなわち、{エンジン、サイズ}による分類で、good と判定されるものは、{x1, x5, x6, x8}である。{エンジン、色}の場合、{{x1}, {x2}, {x3}, {x4}, {x5}, {x6, x8}, {x7}}と分類され、good な分類の下近似をとると{x1}, {x5}と{x6,x8}を得ることになり、再び{x1, x5, x6, x8}を得る。すなわち、｛エンジン、サイズ｝と｛エンジン、色｝は決定属性に関して同じ分類を与えるので、エンジンに対してサイズと色のどちらかの属性のみを加味して考えれば十分である。このように、冗長性を排して、データ識別に必要最低限の属性集合をとることを縮約という。

また、表 1 は、

If {propane} and {compact} and {black} Then {good}

という形の 8 つの決定ルールを示していると見ることもできる。このままでは、汎用的なルールにならないので、データマイニングとしては、より単純化した If-Then ルールを得ることが目標となる。

表 1 で最も簡単なルール作成の例として、再びエンジンの

属性による分類を考える。U の分類は、X1 = {x1}, X2 = {x2, x3, x4}, X3 = {x5, x6, x7, x8}である。一方、決定属性による分類は、Y1 = {x1, x5, x6}, Y2 = {x2, x3, x4, x7, x8}である。これから生成されるルールは、Y1 に関して

$$\tau_{11} : X1 \cap Y1 = \{x1\}$$
$$\tau_{21} : X2 \cap Y1 = \phi$$
$$\tau_{31} : X3 \cap Y1 = \{x5, x6\}$$

ここで、$X1 \cap Y1 = X1$、$X3 \cap Y1 \neq X3$ より、

$$\tau_{11} : \textit{if}\ \text{propane}\ \textit{then}\ \text{good}$$

を採用すると良いことが分かる。また Y2 に関して

$$\tau_{12} : X1 \cap Y2 = \phi$$
$$\tau_{22} : X2 \cap Y2 = \{x2, x3, x4\}$$
$$\tau_{32} : X3 \cap Y2 = \{x7, x8\}$$

ここで、$X2 \cap Y2 = X2$、$X3 \cap Y2 \neq X3$ より、

$$\tau_{22} : \textit{if}\ \text{diesel}\ \textit{then}\ \text{bad}$$

を採用すると良いことが分かる。U の分類を変更すれば、異なるルールを生成することができる。この If-Then ルールで、If 条件部が単純であるほど、分かりやすいルールとなることが期待される。

ラフ集合によるデータマイニングでは、縮約を利用して決定属性を決めるための必要最小限の属性値の組み合わせを見つけ、決定ルールを抽出する。決定ルールの抽出は、結論に影響を与えない属性を取り除き、決定属性の識別に重要なものだけを見出すことになり、とりもなおさず知識獲得となっている。ラフ集合の演算自体は単純な集合演算であるが、データ数が増えれば組み合わせも膨大となり、コンピュータ処理が必要となる。本研究では、文献 2 で開発されインターネット上で公開されている Rosetta というソフト[4]を用いた。

3．超音速翼設計の自己組織化マップ(SOM)

以前に、SST 主翼の遷音速巡航、超音速巡航の抵抗係数及び超音速巡航時の翼根にかかる曲げモーメント、翼先端にかかる捻りモーメントの 4 つを最小化する多目的最適化を行った[5]。この最適化では 72 の設計変数を用い、目的関数の評価には、境界層の効果も考慮するために、遷音速・超音速ともにナヴィエ・ストークス方程式を用いた。Baldwin-Lomax の乱流モデルを用い、レイノルズ数 1.0×10^7 の全面乱流を仮定した。MOGA では一世代 64 個体とし、パレート面が変化しなくなるまで 75 世代進化させた。

今回のように 4 目的の場合、計算結果は 4 次元目的関数空間における 3 次元曲面として表される。しかし、このトレードオフの様子を直観的に把握することは困難である。目的関数が 2 つや 3 つなら図示することは明白であるが、それ以上になると高次元空間の可視化となるからである。そこで、766 の解から、SOM を作ると図 2 のようになり、各目的関数を最小化する極限パレート解をそれぞれ含むようなクラスタができる。また同じマップを各目的関数で色づけすると図 3 が得られる。クラスタ間では、ピッチングモーメントの小さい翼と遷音速抵抗の小さい翼に類似性があり、また遷音速抵抗と超音速抵抗の小さい翼にも類似性があることが分かる。これらに共通することはアスペクト比が高いことである。図 2 の中の Pareto A, B は航空宇宙技術研究所で線形理論に基づいて設計した 2 次設計翼（NAL 2nd）より 4 目的すべてで優れた解であり、実用的な解はアスペクト比を抑えたものであることが確認できる。

図 2 のマップをいくつかの設計変数で色づけしてみる（図 4）。図 3 と図 4 を比較すると、たとえば dv02 とそれを含むクラスタは超音速抵抗とピッチングモーメントの変化に非常に関連していることが分かる。これに対して dv51 は、マップの色がランダムであり、どの目的関数にも寄与していないことが分かる。この設計変数は、設計上あまり重要でないことが予想できる。こうして設計変数とその役割についての知識が発見できる。

この方法は、人間が得意とするパターン認識に頼っている点で、明らかな傾向が見て取れる時には効率的であるという利点を持つが、傾向が曖昧なときには判断も曖昧になる・設計変数が多いとすべてを比較して見るのが大変といった欠点も持っている。そこで、データマイニングの自動化を図るため、ラフ集合の適用を考えることにした。

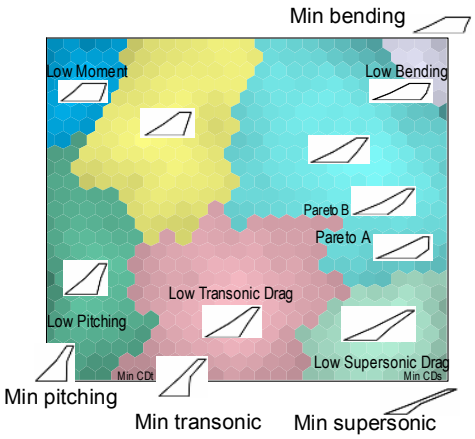

図 2　近似パレート解の目的関数値による SOM

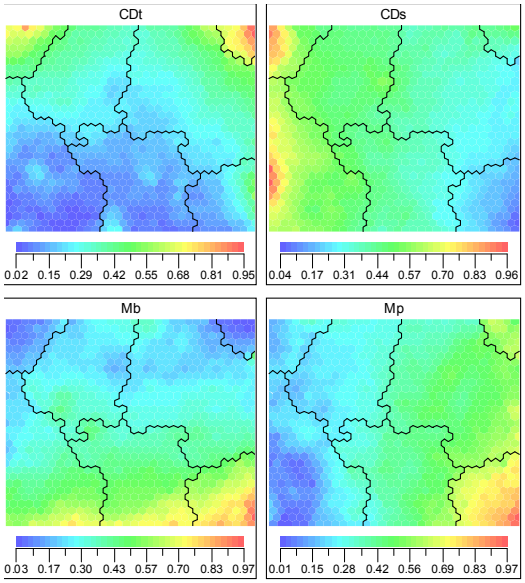

図 3　各目的関数値で色分けされた SOM

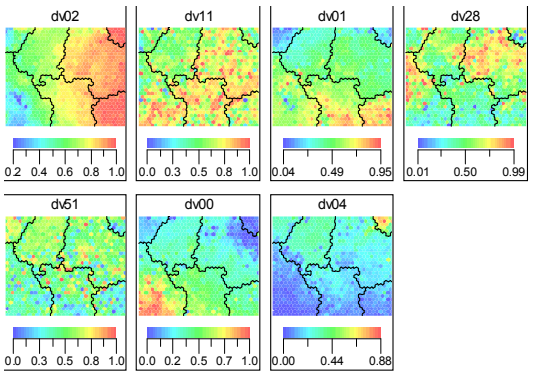

図 4　いくつかの設計変数で色分けされた SOM

４．ラフ集合によるルール生成

ラフ集合によるデータマイニングの流れは図5のようにまとめることができる。前節のデータは、766 の近似パレート解に対し、72 の設計変数値と、4 つの目的関数値からなる数値データである。対象 U が近似パレート解、条件属性 C が設計変数、決定属性 D が目的関数となる。

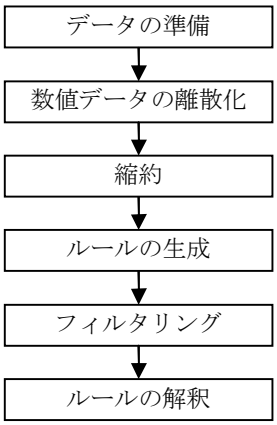

図 5　ラフ集合によるデータマイニングの流れ

766 の結果は進化計算の結果であって、全体としてパレート面を近似しているが、その分布には粗密があり、データとして偏りがある。そこで、データマイニングに入る前に、データのクリーニングを行った。具体的には、766 の解から近似曲面を生成し、Latin Hypercube により 100 個のデータを均等にサンプリングした。これにより、得られたパレート面を偏りなく表す解のサンプル集合が用意できた。

設計変数や目的関数が連続値であるのに対して、ラフ集合で扱う属性値は離散値である。そこで、まずデータの離散化が必要となる。設計変数は、変数毎に領域を 4 分割し、各区間でデータ数が同じになるように、分割幅を自動的に定めた。これより各変数について、{小}{中の小}{中の大}{大} の 4 つの属性値を与える。決定属性については、単純な離散化では 4 目的関数の関係を同時に考慮することが難しいので、100 個のデータについて SOM を作り直し、図 6 の 7 つのクラスタのどれに属するかを属性値とすることにした。そこで、決定属性として、{C1} ~ {C7} の 7 つの属性値を与える。

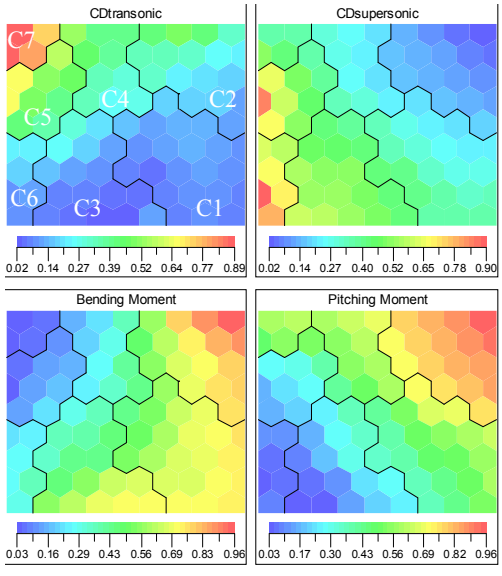

図 6　準備された 100 個のデータによる SOM とクラスタ分析結果（C1~C7）

100 個のデータに対して、上記の離散化を施した後、{C1} ~ {C7} の 7 つの決定属性値に対して、それぞれ縮約とルールの生成を行う。この部分はほぼ自動的に計算される。問題は、機械的に生成されるルールが数千にも及ぶことである。そこから意味のあるルールを見つけるべく、さらに絞り込みが必要となる。このプロセスがフィルタリングである。ルールの当てはまりを見るために様々な指標が提案されているが、ここでは単純にルールが当てはまるデータ数でフィルタリングした。すなわち、ルールが成立するようなデータが 4 つ以下しかないようなルールは自動的に切り捨てた。その結果合計 164 のルールが得られた。

これらのルールから、トレードオフの傾向をつかむために、極限パレート解を含むようなクラスタに着目し、どのような条件属性を満たすとき特定のクラスタに帰着するのかを見てみよう。まず表 2 にクラスタの番号と特徴をまとめる。

表 2　自己組織化マップ上のクラスタの特徴

クラスタ	性能改善の特徴
C2	supersonic drag
C3	transonic drag
C6	Pitching moment
C7	Bending moment

次に設計変数の概要を、表 3・図 7 にまとめる。DV1~6 が翼の平面形を決める変数で、DV7～26 はキャンバーを、DV27~33 は翼のねじりを、DV34~72 は翼の厚み分布を与える曲線のパラメータである。

表 3　翼平面形の設計変数

設計変数	内容
DV1	スパン長（内翼,b_{in}）
DV2	スパン長（外翼,b_{out}）
DV3	スイープ角（内翼,α_{root}）
DV4	スイープ角（外翼,α_{kink}）
DV5	コード長（ルート,C_{root}）
DV6	コード長（キンク,C_{kink}）

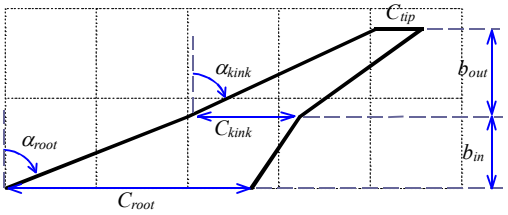

図 7　翼平面形の定義

表 4 に、生成されたルールを当てはまるデータ数順に 20 位まで並べる。上 2 つを見ると、DV2・DV3 または DV2・DV4 の組み合わせがともに{大}なら、スパン長が長く後退角も大きいので、{C2} すなわち超音速性能に優れた翼となることが示されている。

一方、遷音速抵抗を改善する {C3} となるには、DV3・DV4 のスイープ角や DV5・DV6 のコード長がともに{小}で、後退角が浅くアスペクト比の大きい翼となり、さらに翼厚を決定する特定の変数と関係があることが示されている。また、曲げモーメントを改善する {C7} となるには、DV2 のスパン長が{小}で DV5・DV6 のコード長が{大}の、アスペクト比の小さい翼となることが示されている。

これらのルールは、翼の平面形については解釈が簡単であ

り、既知の空気力学的知見と合致するもので、妥当なルールが得られていると評価できる。一方、その他の変数については解釈が難しい。属性として、曲線を指定する設計変数そのものをとるのではなく、技術者の分かりやすい属性を取り上げてデータマイニングをする方が、よりよいマイニングを行えるかもしれない。

５．まとめ

本稿では、超音速機主翼の多目的最適化を対する、ラフ集合によるデータマイニングの適用例を報告した。多目的最適化に際して、進化計算は一度の計算で多数のパレート解を生成する。パレート解は単一目的最適解に比べ、はるかに多くの情報を提供してくれる。パレート解に SOM を適用すると、パレート解の可視化のみならず、クラスタリングにより設計空間の分類・構造の解明にも役立つ。本研究では、SOM によるクラスタリングの結果を利用して、ラフ集合理論により特定のクラスタに帰属するために必要な条件(決定ルール)を生成した。

ラフ集合によって生成されたルールには、既知の設計知識と合致するものもあり、妥当な結果を得ている。しかし、機械的にルールを生成すると、大量のルールが生成され、フィルタリングやその後の解釈が課題となる。また、設計変数そのものを属性としてデータマイニングを適用しても、結果の解釈が難しく、属性の取り方にも課題があることが分かった。

しかし、いくつかのルールは確かにクラスタを特徴づける決定ルールとなっており、本データマイニング法の可能性を裏付ける成果を得た。本稿の例は、最適化法が単にブラックボックスとして問題の答えをもたらすものではなく、問題についての知識をもたらす可能性を持つことを示している。計算科学におけるシミュレーションは、最適化を通じてデータマイニングと出会うことで、自然現象の再現から知識発見に展開しているといえよう。

参考文献

[1] Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski, *Data mining methods for knowledge discovery*, Kluwer Academic, Boston, Mass., 1998.

[2] A. Øhrn, *Discernibility and Rough Sets in Medicine: Tools and Applications*, PhD thesis, Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. NTNU report 1999:133, IDI report 1999:14, ISBN 82-7984-014-1, 1999. http://www.idi.ntnu.no/~aleks/thesis/

[3] 森典彦、田中英夫、井上勝雄編、「ラフ集合と感性」、海文堂出版、東京、2004.

[4] A. Øhrn, *ROSETTA Technical Reference Manual*, Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2000. http://rosetta.lcb.uu.se/general/resources/manual.pdf

[5] Shigeru Obayashi and Daisuke Sasaki, “Visualization and Data Mining of Pareto Solutions Using Self-Organizing Map,” *Second International Conference on Evolutionary Multi-Criterion Optimization*, Faro, Portugal, LNCS 2632, Springer-Verlag Berlin Heidelberg, pp. 796-809, 2003.

表 4　データの当てはまり数が 20 位までのルール

ルール	データ数
dv2([0.80, *)) AND dv3([0.98, *)) => Cluster(C2)	12
dv2([0.80, *)) AND dv4([0.90, *)) => Cluster(C2)	10
dv1([0.30, 0.50)) AND dv4([0.90, *)) => Cluster(C2)	9
dv3([0.98, *)) AND dv56([0.54, 0.74)) => Cluster(C2)	8
dv3([0.98, *)) AND dv6([0.16, 0.25)) => Cluster(C2)	8
dv34([*, 0.15)) AND dv49([0.41, 0.59)) => Cluster(C2)	8
dv2([0.80, *)) AND dv18([*, 0.10)) => Cluster(C2)	8
dv11([0.89, *)) AND dv21([0.67, *)) => Cluster(C2)	7
dv1([0.50, 0.62)) AND dv3([0.98, *)) => Cluster(C2)	7
dv4([0.90, *)) AND dv5([0.09, 0.21)) => Cluster(C2)	7
dv2([0.80, *)) AND dv17([0.12, 0.23)) AND dv49([0.41, 0.59)) => Cluster(C2)	7
dv2([0.80, *)) AND dv63([0.36, 0.58)) => Cluster(C2)	7
dv1([0.30, 0.50)) AND dv2([0.80, *)) AND dv49([0.41, 0.59)) => Cluster(C2)	7
dv2([0.80, *)) AND dv3([0.98, *)) AND dv34([*, 0.15)) => Cluster(C2)	7
dv4([*, 0.33)) AND dv6([*, 0.16)) AND dv48([*, 0.31)) => Cluster(C3)	7
dv3([*, 0.57)) AND dv5([*, 0.05)) AND dv35([0.85, *)) => Cluster(C3)	7
dv3([*, 0.57)) AND dv6([*, 0.16)) AND dv62([0.37, 0.50)) => Cluster(C3)	7
dv1([0.62, *)) AND dv5([*, 0.05)) AND dv35([0.85, *)) => Cluster(C3)	7
dv2([*, 0.47)) AND dv41([*, 0.25)) => Cluster(C7)	7
dv5([0.21, *)) AND dv6([0.41, *)) AND dv41([*, 0.25)) => Cluster(C7)	7

dv の引数は各設計変数の区間を示し、*は上下限値を示す。左に*を含めば属性{小}、右なら{大}である。