Trade-off analysis on the multi-objective design optimization

メタデータ	言語: jpn 出版者: 公開日: 2017-10-03 キーワード (Ja): キーワード (En): 作成者: メールアドレス: 所属:
URL	https://doi.org/10.24517/00007882

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 International License.

多目的最適設計におけるトレードオフ分析法

北山哲士[*1]　山崎光悦
荒川雅生[*2]　山川　宏

Trade-Off Analysis on the Multi-Objective Design Optimization

Satoshi KITAYAMA, Koetsu YAMAZAKI, Masao ARAKAWA, Hiroshi YAMAKAWA
Faculty of Engineering, Kanazawa University
Kakuma-machi, Kanazawa, 920-1192, Japan

The aims of the multi-objective design optimization are to find the pareto optimal solutions, and are also to analyze the trade-off between the objectives. The interactive multi-objective design optimization based on the Satisficing Trade-Off Method (STOM) is used in this paper. Especially, the trade-off matrix to analyze the trade-off between objectives quantitatively is introduced. Each element of trade-off matrix consists of the projection matrix of active constraints and the gradient of objective functions. Additionally, it is possible to determine the new aspiration level by using the trade-off matrix. To obtain the new aspiration level, the compromise point and solution are defined by using the trade-off ratio. To obtain the compromise point, the designer input the trade-off ratio between objectives and can obtain the new aspiration level. It is also possible to apply the trade-off matrix to the automatic trade-off method, which is a method to determine the new aspiration level. Through basic numerical examples, the validity is examined.

Key Words **: Engineering Optimization, Multi-Objective Optimization, Optimal Design, Trade-Off Analysis, Satisficing Trade-Off Method, Compromise Point, System Engineering**

1　緒言

単一目的関数下における最適設計の目的は，大域的最適解を求めることであり，設計変数空間で考えることが多い．一方，多目的最適設計問題では，複数の競合する目的関数が存在することが想定されるため，設計変数空間ではパレート最適解を求め，同時にパレート最適解を目的関数空間へ写像したパレート最適値から成るパレートフロントを求めること，また目的関数空間において競合する目的関数間のトレードオフ分析を行うという二点に重点が置かれる．多目的最適設計問題におけるパレート最適解の探索方法に関しては，大きく次の三つの方法に分類できると思われる．

（１）スカラー化法[(1)]：線形加重和法，制約法，*Lp*ノルム法など．

（２）対話型手法[(1)]：STEp Method（STEM），満足化トレードオフ法，基準点法など．

（３）進化的計算法[(2,3)]：多目的遺伝的アルゴリズム*(MOGA)，多目的Particle Swarm Optimization（MOPSO）などに代表される多点同時探索法．

スカラー化法の中でも，特に線形加重和法は，考え方が極めて容易であるため，実際の設計にもしばしば用いられている方法であるが，目的関数と重みの間には，明確な相関関係が存在ないこと，またいくつかの目的関数や制約条件に関する条件など[(4)]から，必ずしも多目的最適設計問題のすべてのパレート最適解を求めることはできない[(5)]．

進化的計算法によるパレート最適解の探索に関しては，近年のコンピュータの飛躍的な進歩により，盛んに研究が行われている．進化的計算法によるパレート最適解探索では，パレートフロントが非凸であったり分離していても，一度の計算で多くのパレート最適解が得られるという利点がある一方で，アルゴリズムに内在するいくつかのパラメータを適切に設定しなければならず，また，多くの計算コストを要するなどの問題点を含む．

一方，対話型手法の多くは，各目的関数の目標値である希求水準をコンピュータに直接入力し，補助的最適化問題を解くことによってパレート最適解およびパレート最適値を求め，設計者の満足のゆく解が得られるまでこの操作を繰り返すため，自然と設計者の価値判断を含んだトレードオフ分析を行っていることになり，また設計問題の構造を暗に理解できるなど，設計者とコンピュータの役割が上手く分担されており，あ

*　原稿受付　平成？？年？月　日
*[1]正員，金沢大学理工学域（〒920-1192　金沢市角間町）．
*[2]正員，香川大学工学部（〒761-0396　高松市林町2217-20）．
*[3]正員，早稲田大学理工学術院（〒169-8555　新宿区大久保3-4-1）．
E-mail: kitagon@t.kanazawa-u.ac.jp

る意味で優れている方法であると思われる．特に，中山が提唱した満足化トレードオフ法[6]（Satisficing Trade-Off Method:STOM）は，希求水準を直接調整する方法であり，非常に多くの研究や実設計への適用の報告がなされている[1),(7)]．

対話型手法の多くは，複数回，希求水準を入力し，得られたパレート最適値から，各目的関数間のトレードオフ分析を行うことが一般に行われれているが[8),(9)]，このような方法は必ずしも，定量的なトレードオフ分析法とは言い難い．ここで言う定量的なトレードオフ分析法とは，各目的関数のトレードオフ比を算出し，この結果を利用して新しい希求水準を決めることを意味しており，さらに言及すれば，妥協・満足するような希求水準を決めることである．このようなトレードオフ分析法としては，自動トレードオフ法[10),(11)]が挙げられる．自動トレードオフ法は，改善する目的関数に対して新しい希求水準を入力すれば，緩和する目的関数の新しい希求水準が自動的に求まるというものであるが，必ずしも設計者が妥協・満足するような希求水準を決める方法ではない．また後述するように，改善する目的関数と緩和する目的関数のラグランジュ乗数ベクトルの算出が必要となるが，設計者としては，ラグランジュ乗数ベクトルの数学的な意味を理解して新しい希求水準を決めるよりも，目的関数間のトレードオフ比が定量的かつ直接的に示されるほうが，新しい希求水準を決め易いと思われる．

そこで，本論文では，対話型手法を活用した定量的なトレードオフ分析法を含む多目的最適設計法を示す．対話型手法としては，満足化トレードオフ法を用いる．また，ラグランジュ乗数ベクトルの算出を必要としないトレードオフ分析法として，文献(12)があるが，数学的な考察が全く行われていない．そこで，文献(12)のトレードオフ分析法に対して，数学的な検討を行い，その妥当性を示す．またこの方法を用いれば，目的関数間のトレードオフ関係はマトリックス表記でき，さらにマトリックスの非対角項を用いることにより，自動トレードオフ法への適用が可能であることを示す．さらに，提示する定量的なトレードオフ分析法から得られる知見を基に，定量的に妥協点・妥協解を定義し，それらを得るような希求水準の算出法を新たに提案する．第2章において，多目的最適設計や満足化トレードオフ法に関する簡単な記述をし，第3章において，本論文で提示する定量的なトレードオフ分析法，妥協点の定義や妥協点を得るような希求水準の算出法について記述する．数値計算例を通じ，本論文で提示する多目的最適設計法の有効性を検討する．

2　多目的最適設計

2.1　定式化　一般に，k 個の目的関数を同時に最小化する多目的最適設計問題は，次のように定式化される．

$$(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x})) \to \min \tag{1}$$

$$x_i^L \le x_i \le x_i^U \quad i = 1, 2, \cdots, n \tag{2}$$

$$g_j(\boldsymbol{x}) \le 0 \quad j = 1, 2, \cdots, m \tag{3}$$

ここで $f_i(\boldsymbol{x})$ は i 番目の目的関数，$\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T$ は設計変数ベクトル，n は設計変数の数を表す．また x_i^L と x_i^U は i 番目の設計変数ベクトルに直接的に課される側面制約条件であり，$g_j(\boldsymbol{x})$ は挙動制約条件，m はその数を表す．いくつかの目的関数を最大化する場合等は，最大化する目的関数にマイナス記号を付ければ，最小化問題へ帰着される．

2.2　パレート最適解　多目的最適設計問題では，いくつかの目的関数間にトレードオフの関係があるため，すべての目的関数を同時に最適にする解（完全最適解）は一般には存在せず，ある特定の目的関数を改善するためには，少なくとも他の一つ以上の目的関数を犠牲にしなければならない解の集合（パレート最適解：$\boldsymbol{x}^p$）が導入される．また，パレート最適解を目的関数空間へ写像した集合をパレートフロントとよび，パレート最適解の中の点に対応するパレートフロント上の各点をパレート最適値 $(f_1^p, f_2^p, \cdots, f_k^p)$ と呼ぶ．

2.3　理想点および最悪点　理想点とは，各目的関数を単独で最小化したときの目的関数値で，f_i^I（$i = 1, 2, \cdots, k$）と表記する．一方，最悪点とは，パレートフロントの上限値のことであり，f_i^N（$i = 1, 2, \cdots, k$）と表記する．最悪点はペイオフ行列[13)]を用いて算出することも可能である．

2.4　満足化トレードオフ法　満足化トレードオフ法（Satisficing Trade-Off Method）は，STEMとほぼ同じ考え方であるが，その最大の特徴は，STEMが満足した目的関数に対する緩和量を調整するのに対し，STOMでは希求水準を直接調整するというものである．また，各目的関数の重みは，STEMでは理想点と最悪点を設定すると固定されていたのに対し，STOMでは希求水準に応じて変化するという特徴もある．STOMの流れを以下に示す．

（STEP1）各目的関数の理想点 f_i^I および最悪点 f_i^N（$i = 1, 2, \cdots, k$）を決める．なお，理想点および最悪点は，設計者が任意に決めても良い．

（STEP2）各目的関数の希求水準 f_i^A（$i = 1, 2, \cdots, k$）を決め，次式で重みを計算する．

$$w_i = 1 / (f_i^A - f_i^I) \quad i = 1, 2, \cdots, k \tag{4}$$

(STEP3)重み付きTchebyshevノルム問題(min-max問題)を設定.

$$\min \max_{i=1,2,\cdots,k} \{w_i(f_i(\boldsymbol{x})-f_i^I)\} \tag{5}$$

$$x_i^L \le x_i \le x_i^U \quad i=1,2,\cdots,n \tag{6}$$

$$g_j(\boldsymbol{x}) \le 0 \quad j=1,2,\cdots,m \tag{7}$$

これを次の等価な補助最適化問題へ変換し，最適解を求める.

$$\min \quad z \tag{8}$$

$$w_i(f_i(\boldsymbol{x})-f_i^I) \le z \quad i=1,2,\cdots,k \tag{9}$$

$$x_i^L \le x_i \le x_i^U \quad i=1,2,\cdots,n \tag{10}$$

$$g_j(\boldsymbol{x}) \le 0 \quad j=1,2,\cdots,m \tag{11}$$

(STEP4)STEP3で得られた解における目的関数値に満足であれば，終了．そうでなければ，STEP2へ戻る.

3　トレードオフ分析法

本章では，はじめにトレードオフ比に関する簡単な記述をし，その後，本論文で提示するトレードオフ分析法について述べる．また，このトレードオフ分析法を活用した自動トレードオフ法への適用，さらには，妥協点を得るような希求水準の算出方法について記述する.

3.1　トレードオフ比　式(1),(2)から成る事実上の無制約多目的最適設計問題を考える．このとき，ラグランジュ関数は

$$L(\boldsymbol{x},\boldsymbol{\mu}) = \sum_{i=1}^{k} \mu_i f_i(\boldsymbol{x}) \tag{12}$$

である．ここで $\boldsymbol{\mu}=(\mu_1,\mu_2,\cdots,\mu_k)^T$ はラグランジュ乗数ベクトルである．式(12)の設計変数に対する停留条件より，

$$\sum_{i=1}^{k} \mu_i \nabla f_i(\boldsymbol{x}) = 0 \tag{13}$$

が得られる．ここで $\nabla f_i(\boldsymbol{x})$ の一次独立性，ラグランジュ関数の二次の最適性および相補性が成立すると仮定すると，式(13)は微小変動量 Δf_i を用いて，次のように表現することができる[11].

$$\sum_{i=1}^{k} \mu_i \Delta f_i = 0 \tag{14}$$

ここでトレードオフ関係にある i 番目と j 番目の目的関数を考え，他の目的関数は固定する．このとき，i 番目の目的関数が Δf_i だけ増加（減少）したとき，j 番目の目的関数は Δf_j だけ減少（増加）するため，

$$\Delta f_j / \Delta f_i = \mu_i / \mu_j \tag{15}$$

となる．式(15)は一般にトレードオフ比と呼ばれる．また，重み付きTchebyshevノルムを用いた場合のトレードオフ比は

$$\Delta f_j / \Delta f_i = \mu_i w_i / \mu_j w_j \tag{16}$$

となる．式(15),(16)からわかるようにトレードオフ比の算出には，競合する目的関数間のラグランジュ乗数ベクトルの算出が必要となる.

3.2　トレードオフ行列　ある設計点 $\boldsymbol{x}$ において，i 番目の目的関数 $f_i(\boldsymbol{x})$ と j 番目の目的関数 $f_j(\boldsymbol{x})$ の間にはトレードオフの関係があるものとする.

はじめに無制約多目的最適設計問題について考える．設計変数空間において $\Delta\boldsymbol{x}$ の微小変動を与えたとき，$f_i(\boldsymbol{x})$ の微小変動量 Δf_i は，近似的に

$$\Delta f_i \simeq \nabla^T f_i(\boldsymbol{x}) \Delta \boldsymbol{x} \tag{17}$$

で与えられる．また，トレードオフ比を表す量 $t_{i,j}$

$$t_{i,j} = -\Delta f_j / \Delta f_i \tag{18}$$

を導入する．式(18)右辺において，マイナスは，$f_i(\boldsymbol{x})$ と $f_j(\boldsymbol{x})$ がトレードオフの関係にあることを意味している．ここで，式(17)の両辺に左側から $\nabla f_i(\boldsymbol{x})$ を掛け，$\Delta\boldsymbol{x}$ について解けば，

$$\begin{aligned} t_{i,j} &= -\Delta f_j / \Delta f_i \\ &\simeq -\nabla^T f_j(\boldsymbol{x})[\nabla f_i(\boldsymbol{x}) \nabla^T f_i(\boldsymbol{x})]^{-1} \nabla f_i(\boldsymbol{x}) \end{aligned} \tag{19}$$

を得る．このように $f_i(\boldsymbol{x})$ と $f_j(\boldsymbol{x})$ のトレードオフ比 $t_{i,j}$ は，式(19)より近似的に求めることができる.

次に有制約多目的最適設計問題の場合を考える．ここで，ある設計点 $\boldsymbol{x}$ において q 個の制約条件がアクティブであるとする．またアクティブな制約条件を $g_i^a(\boldsymbol{x})$（$i=1,2,\cdots,q$）と表記する．アクティブな制約条件の感度から成る射影行列を $\boldsymbol{P}$ とすれば，

$$\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{N}(\boldsymbol{N}^T\boldsymbol{N})^{-1}\boldsymbol{N}^T \tag{20}$$

で与えられる．ただし，

$$\boldsymbol{N} = [\nabla g_1^a(\boldsymbol{x}), \nabla g_2^a(\boldsymbol{x}), \cdots, \nabla g_q^a(\boldsymbol{x})] \tag{21}$$

$$\nabla g_i^a(\boldsymbol{x}) = (\partial g_i^a / \partial x_1, \partial g_i^a / \partial x_2, \cdots, \partial g_i^a / \partial x_n)^T \tag{22}$$

である．設計変数空間において，$\Delta\boldsymbol{x}$ の微小変動を与えたとき，$f_i(\boldsymbol{x})$ の微小変動量 Δf_i は，近似的に

$$\Delta f_i \simeq \nabla^T F_i(\boldsymbol{x}) \Delta \boldsymbol{x} \tag{23}$$

で与えられる．ただし，

$$\nabla F_i(\boldsymbol{x}) = \boldsymbol{P} \nabla f_i(\boldsymbol{x}) \tag{24}$$

である．射影行列 $\boldsymbol{P}$ を考えることにより，制約条件のある場合のトレードオフ比を求める問題を，事実上無制約多目的最適設計問題におけるトレードオフ比を求める問題へ変換する．式(23)の両辺に左側から $\nabla F_i(\boldsymbol{x})$ を掛け，$\Delta\boldsymbol{x}$ について解けば，

$$\begin{aligned} t_{i,j} &= -\Delta f_j / \Delta f_i \\ &\simeq -\nabla^T F_j(\boldsymbol{x})[\nabla F_i(\boldsymbol{x}) \nabla^T F_i(\boldsymbol{x})]^{-1} \nabla F_i(\boldsymbol{x}) \end{aligned} \tag{25}$$

を得る．制約条件がある場合の $f_i(\boldsymbol{x})$ と $f_j(\boldsymbol{x})$ のトレードオフ比 $t_{i,j}$ は，式(25)より近似的に求めることができる．さらに $t_{i,j}$ を行列形式で表わせば次のようになる.

$$\boldsymbol{T} = \begin{bmatrix} -1 & t_{1,2} & \cdots & t_{k,1} \\ t_{2,1} & -1 & \cdots & t_{k,2} \\ \vdots & \vdots & \ddots & \vdots \\ t_{k,1} & t_{k,2} & \cdots & -1 \end{bmatrix} \tag{26}$$

上式$\boldsymbol{T}$をトレードオフ行列と呼ぶことにする．なお，式(19)，(25)の逆行列の計算において，$[\nabla f_i(\boldsymbol{x})\nabla^T f_i(\boldsymbol{x})]$や$[\nabla F_i(\boldsymbol{x})\nabla^T F_i(\boldsymbol{x})]$は特異となることがあるので，逆行列の計算には，一般化逆行列を用いる．トレードオフ行列は，次の性質を持つ．

（T1）非対角項の値が競合する目的関数間のトレードオフ比を与えている．

（T2）ある設計点$\boldsymbol{x}$において，非対角項の値が正であれば，目的関数間にトレードオフの関係があることがわかる．一方，非対角項の値が負のときは，目的関数間にトレードオフの関係はなく，そのような場合，多目的最適設計では目的関数と制約条件の入れ替えが可能であることから[11]，どちらかの目的関数を制約条件にしたり，目的関数を削減するなど，はじめに設定した設計問題を簡略化できる可能性を持っている．すなわち，目的関数間の競合・従属関係を容易に把握できる．上記のことを例を用いて説明する．

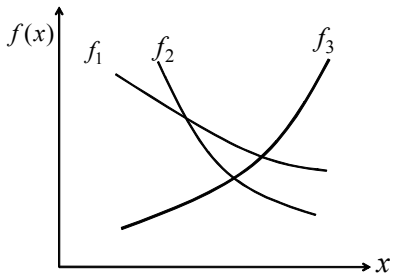

Fig.1 An illustrative example

例えば図1に示すような三つの目的関数f_1，f_2，f_3があるとする．このとき，f_1とf_2の間にはトレードオフの関係がなく，f_1とf_3およびf_2とf_3の間にはトレードオフの関係が存在する．このとき，トレードオフ行列のf_1とf_2に対応する成分(1行2列および2行1列)は負となるので，トレードオフの関係がないことがわかる．すなわち，従来のトレードオフ分析では，希求水準を逐次代入することにより，目的関数間のトレードオフ関係を把握していたのに対し，トレードオフ行列を用いれば，目的関数のトレードオフ関係を瞬時に把握できるという利点を含んでいる．この結果より，f_1もしくはf_2を制約条件にするなどして，目的関数を絞り，実行可能領域を込むことが可能となる．このことは，はじめに設定した複数存在する目的関数を削減することにより，設計問題をより簡素化できる可能性を含んでいる．

（T3）トレードオフ行列の非対角項成分が近似的にトレードオフ比を表していることから，設計者はトレードオフ行列の非対角項成分の値を考慮し，目的関数間のバランスを考慮した希求水準を決めることも可能である．

3．3　自動トレードオフ法への適用　改善する目的関数の成分を$I^<$，緩和する目的関数の成分を$I^>$とする．また，改善する目的関数の希求水準を$f_i^{A,new}$（$i \in I^<$）とし，緩和する目的関数の希求水準を$f_j^{A,new}$（$j \in I^>$）と表記する．各目的関数のパレート最適値f_i^P（$i=1,2,\cdots,k$）からの微小変動量をΔf_iと表記すれば，式(14)より，

$$\sum_{i\in I^<} \mu_i \Delta f_i + \sum_{j\in I^>} \mu_j \Delta f_j = 0 \tag{27}$$

が成立する．ここで，

$$\Delta f_i = f_i^{A,new} - f_i^P < 0 \quad i \in I^< \tag{28}$$

$$\Delta f_j = f_j^{A,new} - f_j^P > 0 \quad j \in I^> \tag{29}$$

である．自動トレードオフ法は，改善する目的関数の希求水準$f_i^{A,new}$（$i \in I^<$）を入力すれば，緩和する目的関数の希求水準$f_j^{A,new}$（$j \in I^>$）の候補値が自動的に求まるというものである．緩和する目的関数の数をNとするとき，$f_j^{A,new}$（$j \in I^>$）は，

$$f_j^{A,new} = f_j^P - \frac{\sum_{i\in I_I} \mu_i (f_i^{A,new} - f_i^P)}{N\mu_j} \tag{30}$$

として，自動的に決まる．重み付きTchebyshevノルムを用いた場合は，

$$f_j^{A,new} = f_j^P - \frac{\sum_{i\in I_I} \mu_i w_i (f_i^{A,new} - f_i^P)}{N\mu_j w_j} \tag{31}$$

となる．式(30)，(31)からわかるように，自動トレードオフ法では，目的関数に対するラグランジュ乗数ベクトルが必要となるが，式(26)を用いれば，ラグランジュ乗数ベクトルを算出することなく，緩和する目的関数の希求水準$f_j^{A,new}$をトレードオフ行列の非対角項の成分を用いて，容易に算出することができる．

3．4　妥協点の算出法　文献(16)によれば，妥協点及び妥協解は妥協計画法(Compromise Programming:CP)を解くことによって得られるとされている．しかし，CPによる解（妥協解）は，複数存在する目的関数をスカラー化法によって単一目的に変換することによって得られる解であり，競合する目的関数間のトレードオフ比を考慮して得られる解ではない．そこで，本研究では3．2節の(T3)で記述した通り，トレードオフ行列の非対角項成分に着目し，目的関数間の妥協点を得るような希求水準の算出する．そこで本論文では，妥協点および妥協解を定量的に次のように定義する．

【　妥協点および妥協解の定義　】

設計者の望む目的関数間のトレードオフ比を$a_{i,j}$とする．このとき，トレードオフ行列の非対角項の成分が，$a_{i,j}$となるような設計変数を妥協解と呼び，そのときの目的関数値を妥協点とと定義する．

自動トレードオフ法は，必ずしも設計者が望む目的関数間のトレードオフ比を得るような希求水準を算出するものではない．そこで，上で定義した妥協点および妥協解を得るような希求水準の決定法を考える．

トレードオフ行列の非対角項の成分の中でも特に，$t_{i,j}$（$j>i$，$i=1,2,\cdots,k-1$，$j=i+1$）の成分を対象とする．このとき，トレードオフ行列の非対角項成分を所望の値$a_{i,j}$に近づければ，妥協点に近づくため，以下の問題を解けば，妥協点を得るための希求水準が得られると考えた．

$$(t_{i,j}-a_{i,j})^2 \to \min$$
$$j>i \quad i=1,2,\cdots,k-1 \quad j=i+1 \tag{32}$$
$$g_j(\boldsymbol{x}) \le 0 \quad j=1,2,\cdots,m \tag{33}$$

ここで式(33)は，実行可能領域内の妥協解を得るための制約条件であるが，対話型手法では，新しい希求水準を得た後，必ずmin-max問題を解いてパレート最適解およびパレート最適値を求めるため，無視しても差し支えない．

式(32)は，二目的の場合は単一目的関数の最小化問題となり，三目的以上となると，多目的関数の最小化問題となる．そのため，三目的以上の場合は，補助変数zを導入し，次のmin-max問題を解けばよい．

$$z \to \min \tag{34}$$
$$(t_{i,j}-a_{i,j})^2 \le z$$
$$j>i \quad i=1,2,\cdots,k-1 \quad j=i+1 \tag{35}$$
$$g_j(\boldsymbol{x}) \le 0 \quad j=1,2,\cdots,m \tag{36}$$

式(34)～(36)の最適解を$\boldsymbol{x}^c$と表記すれば，$f_i(\boldsymbol{x}^c)$（$i=1,2,\cdots,k$）が妥協点を得る希求水準となる．

しかし，競合する目的関数が数多く存在する場合，所望のトレードオフ比$a_{i,j}$をすべて設定することは現実的ではない．そこで本論文では，少なくとも二つの目的関数間の妥協点を得るような希求水準を見つけることを目的とした次のような方法を用いる．

(a) トレードオフ行列の非対角項成分を見て，設計者が満足でいない目的関数間に対し，所望のトレードオフ比$a_{i,j}$を決める．

(b) 設計者が満足でいない目的関数間に対応するトレードオフ行列の成分に対し，

$$|t_{i,j}-a_{i,j}| \to \max$$
$$j>i \quad i=1,2,\cdots,k-1 \quad j=i+1 \tag{37}$$

となる非対角項の成分を一つ見つけ，それを$t^*_{i,j}$とする．

(c) 下記の最適化問題を構築する．

$$D(\boldsymbol{x})=(t^*_{i,j}-a_{i,j})^2 \to \min \tag{38}$$

この問題の最適解$\boldsymbol{x}^c$を見つけ，$f^*_i(\boldsymbol{x}^c)$と$f^*_j(\boldsymbol{x}^c)$を新しい希求水準とする．そして，2.4節STEP3のmin-max問題を解き，妥協解および妥協点を求める．

3.5 提示する多目的最適設計法の流れ 本論文では，対話型手法の一つであるSTOMを基調とした多目的最適設計法である．特にトレードオフ行列を活用し，所望の二つの目的関数間のトレードオフ比を設定し，妥協点を得るような希求水準を求めるところに特徴がある．自動トレードオフ法が，必ずしも所望のトレードオフ比を得る方法でないことをに対し，本論文で提示する希求水準の設定法では，設計者がパレート最適値およびトレードオフ比を見て，所望のトレードオフ比を入力することで，新しい希求水準を求め，妥協解および妥協点を得ようとする方法である．本論文で提示する多目的最適設計法は，いくつかのバリエーションが考えられるが，最も基本的なものとしては，2.4節で記述したSTOMの流れの中で，特にSTEP4を以下のように変更したものである．

(STEP4-1) トレードオフ行列を作成．

(STEP4-2) 自動トレードオフ法を用いて新しい希求水準を決める場合は，改善する目的関数に対する希求水準を入力する．緩和する目的関数の希求水準は，トレードオフ行列を利用して，式(32)から得られる．そしてSTEP2へ戻る．自動トレードオフ法を用いない場合は，STEP4-3へ．

(STEP4-3) 満足でいない目的関数間に対し所望のトレードオフ比$a_{i,j}$を決め，式(37),(38)を用いて，新しい希求水準を決める．そしてSTEP2へ戻り，妥協解および妥協点を求める．

その他の方法としては，3.2節（T2）で記述した通り，任意の設計点$\boldsymbol{x}$が与えられたとき，トレードオフ行列の非対角項が負となるものがある場合は，どちらかの目的関数を制約条件とし，トレードオフ行列の非対角項が正値となるまで，目的関数を減らし，トレードオフの関係があることを明確にしてから，所望のトレードオフ比$a_{i,j}$を決め，式(37),(38)から新しい希求水準を求め，妥協解および妥協点を求める方法が考えられる．また，特定の目的関数に対しは常に希求水準を入れ，一方で他の目的関数間のトレードオフ比を考慮して，それらの目的関数に対してはトレードオフ比を代入する等の方法も考えられる．

4 数値計算例

本論文で提示する多目的最適設計法の有効性を，基礎的な数値計算例を通じて検討する．特に，トレードオフ行列および妥協点を得る希求水準の算出法に特徴があるため，それらの点を重点的に取り扱う．min-max問題および式(38)の最適解探索には，文献(14)を参考

とし，制約条件をペナルティ関数として取扱い，事実上の無制約最適化問題へ変換し，PSOにより最適解を求めた．

4．1　所望のトレードオフ比を複数持つ問題　次のような二目的最小化問題を考える[15]．

$$f_1(\boldsymbol{x}) = \exp(-x_1) + 1.4\exp(-x_1^2) \to \min \tag{39}$$

$$f_2(\boldsymbol{x}) = \exp(x_1) + 1.4\exp(-x_1^2) \to \min \tag{40}$$

$$-2 \le \boldsymbol{x} \le 2 \tag{41}$$

この問題のパレートフロントを図2に示す．

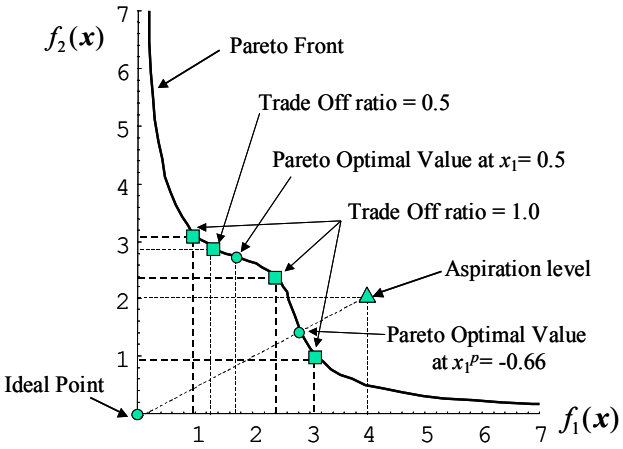

Fig.2 Pareto front and some points on the trade-off ratio

ここで，基礎的な検討として，以下の二つの場合を考える．

Case1：ある設計点が与えられた場合の方法．

Case2：STOMによる方法．

【　ある設計点が与えられた場合　】

各目的関数の理想点を，$f_1^I = 0$，$f_2^I = 0$ とする．トレードオフ行列の非対角項成分 $t_{1,2}$ は，

$$t_{1,2} = \frac{\exp(x_1) - 2.8x_1\exp(-x_1^2)}{\exp(x_1) + 2.8x_1\exp(-x_1^2)} \tag{42}$$

となる．ここで，例えば $x_1 = 0.5$ とすると，$f_1 = 1.697$，$f_2 = 2.739$ となり，トレードオフ行列 $\boldsymbol{T}$ は，

$$\boldsymbol{T} = \begin{bmatrix} -1 & 0.329 \\ 3.039 & -1 \end{bmatrix} \tag{43}$$

となる．トレードオフ行列の非対角項成分が正であるため，$x_1 = 0.5$ において，目的関数間にトレードオフの関係があることがわかる．ここで，設計者が目的関数間のバランスが取れていないと考え，トレードオフ比を1にするような希求水準を求めることを考える．このとき，式(38)は

$$D(\boldsymbol{x}) = (t_{1,2} - 1)^2 \to \min \tag{44}$$

$$-2 \le \boldsymbol{x} \le 2 \tag{45}$$

となる．式(44)の様子を図3に示す．

図3より，式(44)を満足する最適解は，図3中●で示すように，三つある．式(44)の最適解が三つあるとい

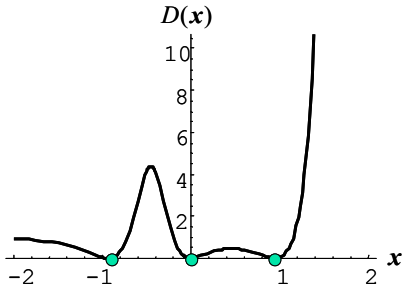

Fig.3 The behavior of $D(\boldsymbol{x})$

うことは，トレードオフ比を1とするような設計点が三つあるということを意味している．PSOで最適解の探索を行った結果，$x_1^c = 0$ が得られ，そのときの目的関数値はそれぞれ，

$$f_1(x_1^c) = f_2(x_1^c) = 2.4 \tag{46}$$

となる．式(46)が妥協点を得るための希求水準となる．この点を希求水準とし，min-max問題を解けば，パレート最適解およびパレート最適値として，

$$x_1^p = 0,\quad f_1^p = f_2^p = 2.4 \tag{47}$$

が得られる．ここで，x_1^p が妥協解，f_1^p と f_2^p が妥協点となる．また，$x_1^p = 0$ におけるトレードオフ行列は，

$$\boldsymbol{T} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \tag{48}$$

となり，所望のトレードオフ比が得られていることがわかる．なお，式(44)の最小値を取る設計変数は，$x_1^c = 0$ のほかに，$x_1^c = \pm 0.941$ であり，これらの点においても，トレードオフ比は1となる．

【　STOMによる方法　】

各目的関数の理想点は同じとし，希求水準を次のように設定する（図2中▲印）．

$$f_1^A = 4,\quad f_2^A = 2 \tag{49}$$

このときのパレート最適解 x_1^p およびパレート最適値 f_1^p，f_2^p は

$$x_1^p = -0.66,\quad f_1^p = 2.841,\quad f_2^p = 1.421 \tag{50}$$

となり，トレードオフ比は次のようになる．

$$\boldsymbol{T} = \begin{bmatrix} -1 & 2.309 \\ 0.433 & -1 \end{bmatrix} \tag{51}$$

ここで，設計者がトレードオフ比を0．5とするような妥協解および妥協点を得たいとする．妥協解および妥協点を得るための希求水準は式(38)より，

$$f_1^A = 1.328,\quad f_2^A = 2.884 \tag{52}$$

となる．2．4節STEP3を解くことにより，妥協解，妥協点およびトレードオフ行列はそれぞれ次のようになる．

$$x_1^p = 0.716,\quad f_1^p = 1.328,\quad f_2^p = 2.884 \tag{53}$$

$$\boldsymbol{T}=\begin{bmatrix}-1 & 0.5\\ 2.0 & -1\end{bmatrix} \tag{54}$$

4.2　有制約多目的最適化問題　次の有制約多目的最適化問題を考える．

$$f_1(\boldsymbol{x})=x_1\rightarrow\min \tag{55}$$

$$f_2(\boldsymbol{x})=x_2\rightarrow\min \tag{56}$$

$$f_3(\boldsymbol{x})=x_3\rightarrow\min \tag{57}$$

$$0\le\boldsymbol{x}\le 1 \tag{58}$$

$$g_1(\boldsymbol{x})=(x_1-1)^2+(x_2-1)^2+(x_3-1)^2-1\le 0 \tag{59}$$

この問題は，線形加重和法で重みを変更しても，設計者の望む通りのパレート最適値が得られない問題であることが知られている[11]．

理想点を $(f_1^I,f_2^I,f_3^I)=(0,0,0)$ とし，所望のトレードオフ比を $a_{1,2}=a_{1,3}=a_{2,3}=1$ と指定した場合，対話過程おける新しい希求水準，パレート最適値，トレードオフ行列を表1 に示す．なお，すべての目的関数間のトレードオフ比を1にするということは，すべての目的関数間で釣り合いが取れるようなパレート最適値を求めるものとしており，トレードオフ行列の非対角項成分が同じになった場合，釣り合いが取れていると解釈した．事実，この問題では，トレードオフ行列の非対角項成分が同じになるパレート最適値は，$f_1^p=f_2^p=f_3^p=0.423$ であり，そのときのトレードオフ行列の非対角項成分はすべて0.5となる．表1より，トレードオフ行列の非対角項成分は厳密には，0.5とはならなかったが，すべての目的関数間に十分釣り合いが取れていると考えられるパレート最適値が得られていることがわかる．また，対話過程において，2行3列が常に選ばれているが，この成分値も対話を繰り返すうちに，0.5へ近づき，結果としては，すべての目的関数間のトレードオフ比をほぼ同じ値に近づけていることがわかる．

5　結言

本論文では，定量的なトレードオフ分析法を含む対話型多目的最適設計法を提案した．定量的なトレードオフ分析法では，目的関数間のトレードオフ比が，近似的に目的関数とアクティブな制約条件の感度で表されること示し，これらを行列形式で表現できることを示した．またトレードオフ行列の各成分が意味することについて，考察を行い，自動トレードオフ法への適用が可能であること，さらには，妥協点および妥協解を定量的に定義し，それらを得るようなトレードオフ比を考慮した希求水準の算出法を示した．簡単な数値計算例を通じ，提示した方法の有効性を検討した．

本論文で提示した手法は，多目的計画問題一般に適用可能な汎用性ある方法である．また「妥協」という言葉を定量的に定義し，妥協点おおび妥協解を得るような希求水準の算出方法に大きな特徴がある．さらに，トレードオフ行列の性質の中でも，特に3.2節(T2)の内容は，実務レベルの多目的問題に対し，極めて有効な方法であると考えている．

本研究を遂行するにあたり，適切なご助言を頂いた杉本博之先生（北海学園大学），中山弘隆先生（甲南大学）には，特に感謝したい．

Table 1 Aspiration level using trade-off matrix and pareto optimal value

Iteration	Obj.	Aspiration Level	Pareto optimal value	Trade-off matrix		
1st iteration	$f_1(\boldsymbol{x})$	0.200	0.240	-1.000	0.917	0.706
	$f_2(\boldsymbol{x})$	0.400	0.479	0.536	-1.000	0.282
	$f_3(\boldsymbol{x})$	0.500	0.599	0.360	0.246	-1.000
2nd iteration	$f_1(\boldsymbol{x})$	0.240	0.283	-1.000	0.719	0.719
	$f_2(\boldsymbol{x})$	0.425	0.502	0.469	-1.000	0.326
	$f_3(\boldsymbol{x})$	0.425	0.502	0.469	0.326	-1.000
3rd iteration	$f_1(\boldsymbol{x})$	0.283	0.374	-1.000	0.563	0.563
	$f_2(\boldsymbol{x})$	0.336	0.444	0.496	-1.000	0.441
	$f_3(\boldsymbol{x})$	0.336	0.444	0.496	0.441	-1.000
4th iteration	$f_1(\boldsymbol{x})$	0.374	0.385	-1.000	0.546	0.546
	$f_2(\boldsymbol{x})$	0.425	0.437	0.498	-1.000	0.456
	$f_3(\boldsymbol{x})$	0.425	0.437	0.498	0.456	-1.000
5th iteration	$f_1(\boldsymbol{x})$	0.385	0.390	-1.000	0.539	0.539
	$f_2(\boldsymbol{x})$	0.429	0.435	0.498	-1.000	0.462
	$f_3(\boldsymbol{x})$	0.429	0.435	0.498	0.462	-1.000
6th iteration	$f_1(\boldsymbol{x})$	0.390	0.421	-1.000	0.502	0.502
	$f_2(\boldsymbol{x})$	0.392	0.423	0.500	-1.000	0.498
	$f_3(\boldsymbol{x})$	0.392	0.423	0.500	0.498	-1.000

参考文献

(1) Miettinen, K.M., Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, (1998).

(2) Deb, K., Multi-Objective Optimization using Evolutionary Algorithms, Wiley, (2001).

(3) Engelbrecht, A.P., Fundamentals of Computational Swarm Intelligence, Wiley, (2005).

(4) Sakawa, M., Optimization of Nonlinear Systems, Morikita Publishers, (1996), pp.114-141.

(5) Das, I., Dennis, J.S., A Closer Look at Drarbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems, Structural Optimization, Vol.14, (1997), pp.63-69.

(6) Nakayama, H., Proposal of Satisficing Trade-Off Method for Multiobjective Programming, Journal of the Society of Instrument and Control Engineers, Vol.20, No.1, (1984), pp.29-35. (in Japanese).

(7) Kamenoi, T., Sugimoto, H., Nakayama, H., A Study on Improved Satisficing Trade-Off Method for Structural Design, Journal of Japan Society of Civil Engineers, No.441. I-18, (1992), pp.117-126.(in Japanese)

(8) Yang, J.B., Chen, C., Zhang, Z.J., The Interactive Step Trade-Off Method (ISTM) for Multiobjective Optimization, IEEE Trans. on Systems, Man, and Cybernetics, Vol.20, No.3, (1990), pp.688-695.

(9) Diaz, A., Interactive Solution to Multiobjective Optimization Problems, International J. for Numerical Methods in Engineering, Vol.24, (1987), pp.1865-1877.

(10) Nakayama, H., Trade-Off Anaylsis Using Parametric Optimization Techniques, European J. of Operational Research, Vol.60, No.1, (1992), pp.87-98.

(11) Nakayama, H., Tanino, T., Theory and Application of Multiobjective Programming, The Society of Instrument and Control Engineers, (1994). (in Japanese).

(12) Tappeta R.V., Renaud, J.E., Interactive Multiobjective Optimization Procedure, AIAA J., Vol.37, No.7, (1999), pp.881-889.

(13) Ehorgott, M., Multicriteria Optimization (2nd Edition), Springer, (2005).

(14) Kitayama, S., Arakawa, M., Yamazak, K., Proposal of Adaptive Range Particle Swarm Optimization, *Trans. of the Japan Society of Mechanical Engineers, Series C*, Vol.73, No.725, (2007), pp.280-287. (in Japanese)

(15) Messac, A., Sundararaj, G.J., Tappeta, R.V., Renaud, J.E., Ability of Objective Functions to Generate Points on Nonconvex Pareto Frontiers, AIAA J., Vol.38, No.6, (2000), pp.1084-1091.

(16) Yu.P.L., Multiple-Criteria Decision Making(Concepts, Techniques, and Extensions), Plenum Press, (1985).

付 録

付録1　式(19)の導出を以下に示す．式(17)の両辺に左側から$\nabla f_i(\boldsymbol{x})$を掛け，$\Delta\boldsymbol{x}$について解けば，

$$\Delta\boldsymbol{x} = [\nabla f_i(\boldsymbol{x})\nabla^T f_i(\boldsymbol{x})]^{-1}\nabla f_i(\boldsymbol{x})\Delta f_i \qquad \text{(A. 1)}$$

を得る．ここでj番目の目的関数に対しても式(17)が成立するため，j番目の目的関数の微小変動量Δf_jに関しては添字をiからjに変更すればよい．ここでΔf_jに対して(A. 1)を代入すれば，

$$\Delta f_j = \nabla^T f_j(\boldsymbol{x})[\nabla f_i(\boldsymbol{x})\nabla^T f_i(\boldsymbol{x})]^{-1}\nabla f_i(\boldsymbol{x})\Delta f_i \qquad \text{(A. 2)}$$

となり，式(18)を考慮すれば，

$$t_{i,j} = -\frac{\Delta f_j}{\Delta f_i} \simeq -\nabla^T f_j(\boldsymbol{x})[\nabla f_i(\boldsymbol{x})\nabla^T f_i(\boldsymbol{x})]^{-1}\nabla f_i(\boldsymbol{x})$$

を得る．ここで，マイナス記号の意味は，i番目とj番目の目的関数間にトレードオフの関係が存在するということである．

付録2　Paritcle Swarm Optimization(PSO)は，多点同時探索型の大域的最適化法の一つである．PSOでは各探索点が「位置」と「速度」を持ち，集団で探索を行い，各探索点の位置と速度を更新しながら，最適解を探索する方法である．k回目の探索において，探索点dの位置$\boldsymbol{x}_d^k$と速度$\boldsymbol{v}_d^k$を用いて，$k+1$回目の位置$\mathbf{x}_d^{k+1}$と速度$\boldsymbol{v}_d^{k+1}$は，次の式を用いて更新される．

$$\boldsymbol{x}_d^{k+1} = \boldsymbol{x}_d^k + \boldsymbol{v}_d^{k+1} \qquad \text{(A. 3)}$$

$$\boldsymbol{v}_d^{k+1} = w\boldsymbol{v}_d^k + c_1 r_1(\boldsymbol{p}_d^k - \boldsymbol{x}_d^k) + c_2 r_2(\boldsymbol{p}_g^k - \boldsymbol{x}_d^k) \qquad \text{(A. 4)}$$

式（A. 4）において，r_1とr_2は[0, 1)の乱数である．またc_1とc_2はパラメータであり，一般には

$$c_1 + c_2 \le 4 \qquad \text{(A. 5)}$$

となるように，c_1とc_2は決められている（本論文では$c_1 = c_2 = 2$とした）．またwは慣性項と呼ばれるパラメータであり，線形的に減少する．$\boldsymbol{p}_d^k$は，探索点dがk回目までの探索において，今までで訪れた最良の解(p-best)を表す．一方，$\boldsymbol{p}_g^k$はk回目の探索における群れ全体の中での最良の解(g-best)を表す．式（A. 3），（A. 4）を用いて探索点を更新するモデルは通常，g-bestモデルと呼ばれる．また式（A. 4）中の$\boldsymbol{p}_g^k$を，k回目までの探索で目的関数値を最良にした探索点$\boldsymbol{p}_g$で置き換えたモデルは，最良値保存型モデルと呼ばれる．PSO（g-bestモデル）の基本アルゴリズムは次のようになる．

（STEP1)探索点数，最大探索回数を決める．また側面

制約条件を設定する．

(STEP2)各探索点に対して，ランダムに初期位置$\boldsymbol{x}_d^k$と初期速度$\boldsymbol{v}_d^k$を決める．$k=1$とする．

(STEP3)各探索点に対して，関数値を計算する．

(STEP4) $\boldsymbol{p}_d^k$と$\boldsymbol{p}_g^k$を求める．

(STEP5)各探索点の速度と位置を式（A.3），(A.4)に従い更新．また慣性項wを以下の式に従い更新．

$$w = w_{\max} - (w_{\max} - w_{\min})/k_{\max} \times k \qquad \text{(A.5)}$$

(STEP6)探索回数kが最大探索回数以下なら$k=k+1$としてSTEP3へ戻る．そうでなければ，探索終了．

本論文では，最良値保存型モデルを用い，式(A.5)中の$w_{\max}$と$w_{\min}$については，$w_{\max}=0.9$，$w_{\min}=0.4$とし，探索点数は20，最大探索回数は200とした．