進化的計算手法による多目的最適化

同志社大工 ○廣安 知之　同志社大工 三木 光範　同志社大院 渡邉 真也

Evolutionary Alogorithms for Multiobjective Optimization

○ Tomoyuki HIROYASU,Doshisha University, tomo@is.doshisha.ac.jp
Mitsunori MIKI,Doshisha University, mmiki@mail.doshisha.ac.jp
Shinya WATANEBE,Graduate School of Engineering, Doshisha University

Abstract: A problem to find design variables that make a value of the objective function maximize or minimize is called an optimization problem. Among optimization problems, problems that have not only one objective function but several functions are called multi-objective optimization problems(MOPs). The MOPs are found in many real world problems. Usually, there is a trade-off relationship between objective functions. Therefore, it is difficult to find the solution that minimize/maximize all objective functions. One of the goals of MOPs is to find Pareto-optimal solutions. These days, many researchers applay evolutionary algorithms to MOPs. Since evolutionary algorithms are multi point searching methods, they are very suited to find Pareto-optimal solutions. In this paper, the basic concept of evolutionary algorithms for MOPs is explained. The important algorithms are also summarized and the problems of evolutional algorithms are discussed.

1　はじめに

制約条件下で目的関数を最大もしくは最小にするような設計変数を決定する問題は, 最適化問題と呼ばれる．スケジューリング問題や構造最適化問題, タンパク質構造同定問題などさまざまな問題が最適化問題である．一方, 複数の目的関数を対象とする問題は多目的最適化問題と呼ばれる．実問題では，単一目的問題の方がまれであり，多くの場合, 多目的問題となる．

多目的最適化問題における意志決定方法は種々考えられるが，その一つにパレート解集合を得ることがあげられる．多目的最適化問題における各々の目的は，同時にそれらの目的値を最小化もしくは最大化することは難しく，通常，一つの目的値を最小化すると他の目的値は増大する，いわゆるトレードオフの関係にある．設計問題などにおいて，特に設計の上流部では，これらの目的の関係が明確でないために，どの程度，ある目的の変更が他の目的へ影響を及ぼすか不明である．そのため，パレート解集合を得ることにより目的の関係が明確になることは非常に有意である．

近年，進化的計算手法，特に遺伝的アルゴリズムにより，多目的最適化問題を解決する手法に関する研究が活発に行われており, これらの手法は"Evolutionary Multi-Objective Optimization (EMO)" と呼ばれている．文献 [1, 2, 3, 4] にはこれらの研究のサーベイ論文・解説がまとめられている．Coello による進化的計算手法による多目的最適化の関連の論文がまとめられた web サイト [5] も存在する．進化的計算手法の国際会議でも EMO のセッションが会議内に設けられ [6, 7]，また, 単独でも EMO に関連する国際会議が開催されている [8]．

本稿では，その進化的計算手法による多目的最適化の概要について, 特に, 遺伝的アルゴリズムを中心に述べる．

2　多目的最適化問題

最適化問題において複数の目的関数を持つような問題は特に多目的最適化問題 (Multi-Objective Optimization Problems: MOPs) と呼ばれる．多目的最適化問題は以下のように定式化できる．すなわち次式で表せる制約条件

$$g_i(x) \leq 0 \qquad (1, 2, \ldots, m) \tag{1}$$

を満足し，複数の目的関数 $f_i(x)$ が

$$min[f_1(x), f_2(x), \ldots, f_n(x)] \tag{2}$$

となるような設計変数 $X \in \mathcal{F}$ を求める問題.

ここで $\mathcal{F}$ は制約条件 (1) を満たす X の集合で，可能領域と呼ばれる．

一般に目的関数の間にはトレードオフの関係があるために一意の解が求まらないのが通常である．そこで多目的最適化問題においては次に示すパレート最適解集合 (Pareto-optimum solutions) の概念が使用される [9]．

1. **優越：** x^1，$x^2 \in \mathcal{F}$ に対してすべての目的関数に対して $f_i(x^1) \leq f_i(x^2)$ が成り立ち，かつ，いくつかの目的関数に対して $f_i(x^1) < f_i(x^2)$ が成り立つとき x^1 は x^2 に優越するという．

2. **パレート最適解：** x^0 に優越する $x^1 \in \mathcal{F}$ が存在しない場合，x^0 は Pareto 最適解であるという．

実問題の設計問題は多目的最適化問題となる場合が多く，通常それらの目的関数はトレードオフの関係にある．それらトレードオフの関係は不明であり，この関係を明らかにすることで，設計はより容易になるものと考えられる．よって，多目的最適化問題においてはこのパレート最適解集合を把握することが一つの目標となる場合がある．

3　GA による多目的最適化

3.1　遺伝的アルゴリズム

遺伝的アルゴリズム (Genetic Algorithm: GA) は生物の遺伝と進化を模擬した確率的多点探索手法の一つである．ここでは，単一目的の最適化問題に対する GA の適用について概要を説明する．

探索点は個体と呼ばれ，コーディングと呼ばれる何らかの操作によりビット列に変換される．これらのビット列は遺伝子と呼ばれる．GA では複数の個体を利用して探索が行われる．これら複数個体全体を母集団と呼ぶ．

GA ではいくつかの遺伝的操作により探索が行われる．複数の個体から次探索点を生成する「交叉」，単一個体から次探索点を生成する「突然変異」，母集団から確率的に次母集団を生成する「選択」である．これらの，一連の操作は「世代」と呼ばれ，世代を繰り返すことによって，母集団は最適解に収束していく．

GA には下記のような特徴を有する．

- 傾斜法のような感度を利用した最適化手法ではないため，連続問題だけでなく離散問題にも適用可能である．

- 種々の最適化問題への適用が比較的容易である．

- 多点探索手法であるため，大局的最適解が探索可能である．

- 設計変数の値はビット列へ変換され，そのビット列で探索が行われるため，設計変数空間では多峰性の問題であっても，探索空間では, 局所解の少ない問題へ変換される可能性がある．

- 多点探索手法の一つであり，解を探索するためには，繰り返しを多く必要とするため，計算コストが高い．

- 得られた解に対して，大局的および局所的最適解の保証がない．

3.2　遺伝的アルゴリズムによる多目的最適化

多目的最適化の目標はいくつか考えられるが，ここではパレート解を得ることを目的とする．GA の多目的問題への適用は容易である．

個体を複数生成し母集団を形成する．それらの母集団に対して，交叉，突然変異，選択の遺伝的操作を繰り返し適用して，最終的にパレート解を得る．GA の多目的最適化問題への適用において，単一目的最適化問題と唯一異なる点は，選択の方法であろう．選択の方法はパレート最適性を陽に扱う方法と陰に扱う方法と 2 種類に分類される．パレートフロントの解を陽に取り扱う方法では, 選択において，なんらかの方法でパレート解に近い個体に高い適合度値を与え，選択される確率を高めるのである．このようにすることで，個体は世代を繰り返すごとに，真のパレート解に近付いていく．

その選択の際に各個体に与える適合度を決定する際に重要な手法がランキングとシェアリングである．

・ランキング ランキングはパレート最適の概念を基に Goldberg が開発した手法である [10)]．この手法では他に優越した個体をランク 1 とする．ランク 1 の個体を取り除いた中で, 他に優越した個体をランク 2 とする．以下, 同様にして個体のランク付けを行う．選択はランクを基にしたランキング法, ランクの逆数を適合度値として行うルーレット法などが存在する．

一方,Fonseca と Fleming も上記とは異なるランキング手法を提案している [11)]．そこでは, 個体のランキングは優越される個体の数に 1 増加させたものとしている．すなわち, ランク 5 の個体は優越された 4 個体が存在することとなる．

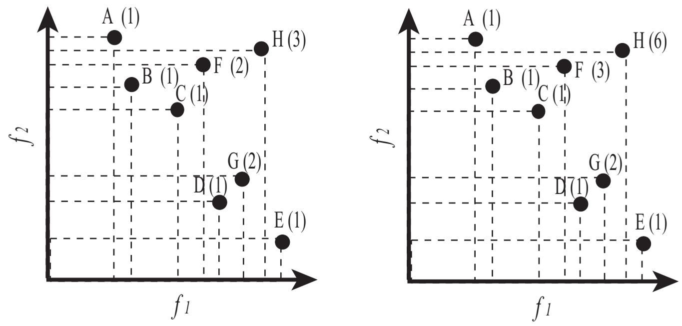
それぞれのランキング手法の 2 目的の場合の概念図を Fig. 1 に示す．

・シェアリング 多目的最適化問題に対し GA を適用する際の重要な点は，解をパレート最適解集合上により広範囲にかつ均等に分布するような解を求めることである．その一つの有効的な方法として Horn らの Goldberg らによって提案されたシェアリングの適用がある [12)]．以下に，シェアリングの概念を適合度に反映させる手法について述べる．

まず，各個体についてその個体の近傍の混み具合をあらかじめ与えられた関数に従ってニッチ数として計算する．ここではニッチ数 m_{x_i} を

$$m_{x_i} = \sum_{j=1}^{N} s(d(x_i, x_j)) \tag{3}$$

と定義しておく．$d(x_i, x_j)$ は個体 i, j 間の距離で，その定義としては幾つかの方法が提案されている．シェアリングの適用に関し，表現空間で行うことを提案している



(a) Goldberg's ranking　(b) Fonseca and Fleming's ranking

Fig. 1　Ranking

もの[13]，目的関数空間で行うことを提案しているもの[11]がある．

また，$s(d)$ はシェアリング関数と呼ばれ，例えば近傍を定めるパラメータ (シェアリング半径)$\sigma > 0$ を与え次式を用いる．

$$s(d) = \max\left(1 - \frac{d}{\sigma}, 0\right) \tag{4}$$

このようにして算出したニッチ数 m_{x_i} でその個体の適合度 $g(i)$ を割り，それを新たな適合度 $g_s(i)$ とする：

$$g_s(i) = \frac{g(i)}{m_{x_i}} = \frac{g(i)}{\sum_j s(d(x_i, x_j))} \tag{5}$$

上式により再計算された適合度は，個体間の集中度合いも考慮に入れているため，この適合度を用いた選択を行うことにより個体が均一に分散された状態で次世代に受け継がれるものと考えられる．Fig. 2 に 2 次元空間におけるシェアリングの概念図を示す．

3.3　GA の多目的最適化の探索メカニズム

3.3.1　探索の特徴

単一目的の最適化問題において，GA が探索する上で重要な点は，

- 大局的最適解を求めるために，探索初期では多様性を維持し，設計可能領域全体を探索する．

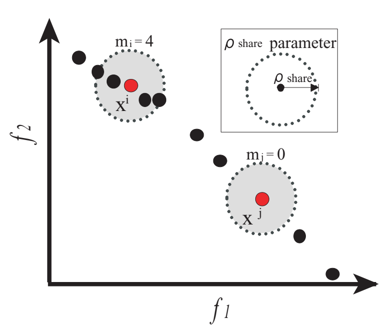


Fig. 2　Sharing

- 余分な計算量を削減するため，探索終期では，最適解に収束させ，局所探索を行う．

ことにある．

一方で，多目的最適化問題において，GA が探索する上で重要な点は，パレート解が解集合であるために，探索初期および終期ともに，解の多様性の維持，および局所探索が必要となる．

そのため，多目的最適化問題では，

- 多様性の維持
- 局所探索

に優れた手法が，より良いパレート解を発見することが可能となる．代表的な手法については後述する．

3.3.2　良好なパレート解

GA による多目的最適化の場合，有限の個体数によって，パレート解集合を表現する．そのため，良好なパレート解とは，以下のような解であると言える．

- 得られた解が真のパレート解であること．
- 得られた解は 1 点に集中するのではなく，できるだけ分散していること．
- できるだけ広い範囲に解が存在すること．

そのため，良好なパレート解を求めるためには，GA の手法は以下のような特徴を有することが必要となる．

- ある世代での個体集合のうち，ランク 1 のものをパレート最適個体と呼ぶ．そのパレート最適個体ができるだけ早い世代で真のパレート解に近付くこと．
- 各目的の最適値が確実に求められていること．

3.4　代表的進化的多目的最適化手法

本節では代表的な手法についてその概要を述べる．

3.4.1　重み付け法

古くから多目的最適化において, 各目的関数 f_i に重み w_i を設定して, 荷重和 $\sum_k w_k f_k$ をとることで単一問題に変換する手法は広く行われていた．この手法を GA に拡張した手法はいくつか見られ, その一つでは重み w_k を個体ごとに変化させることにより, パレート解集合を求める [?]．ただし, 可能領域が目的関数空間の中で非凸である場合に, 一部のパレート解が求められないという問題が存在する．

3.4.2　Schaffer の VEGA

パレート最適性を陽に扱わない手法の代表が Schaffer の Vector Evaluated Algorithms(VEGA)[15] である．この手法では, 選択の際に, 個体を目的関数の数に分割し, それぞれにおいて単一の目的関数値で選択するという手法である．

GA による多目的最適化手法の初期の手法であり, 並列選択であるために, 並列化効率が高いという利点がある半面, パレート解集合すべてが求められるのではなく, 一つの目的関数の値が極端に良い解に集中するという欠点が指摘されている．

しかしながら, 良好な解の要素として, それぞれの目的関数の最適値が求められていることが要求されるため, いずれの手法においても VEGA の機構を組み込むことは非常に有意義である．

3.4.3　Goldberg の手法

パレート最適性を陽に扱う手法は Goldeberg により提案された [10]．個体のランキング方法はすでに述べた通りである．

3.4.4　Fonseca と Fleming の MOGA

Fonseca と Fleming は先に述べたランキング法により Multiple Objective Genetic Algorithm を開発した [11]．この手法では選択圧が必要以上に高まる場合があるので, それを回避するためにシェアリングを利用している．

3.4.5　Srinvas と Deb の NSGA

Srinvas と Deb の Non-dominated Sorting Genetic Algorithm[13] では Goldberg のランキング方法が使用され, 設計変数空間においてシェアリングを用いている．さらに, 選択の際には, 確率を考慮した線形選択手法が採用されている．

3.4.6　Horn と Nafpliotis の NPGA

Horn らはトーナメント選択による手法の提案を行っている [12]．この手法では, 2 個の個体と, 基準になる複数の比較集団個体を選出する．2 個の個体は比較集団個体に対して, ランキングとシェアリングにより, 適合度が決定される．それにより, 適合度の高い個体が次世代に保存される．

本手法は, 個体全体に対してランキングなどが行われないため, 高速に解が求められる可能性がある半面, 比較集団個体のサイズの決定が困難であるという欠点が存在する．

3.4.7　玉置らのパレート保存戦略

玉置らは,VEGA の欠点を補うために, 以下のような手法を適用している [16]．すなわち,VEGA により次個体を決定する中で, 個体集合中のパレート最適個体すべてを時世代の個体集合に残す．個体集合のサイズ以上のパレート最適個体が存在する場合には, それらをシェアリングにより個体を選択する．

パレートフロントをできるだけ高速に真のパレート解集合に近付けるためには, パレート最適個体の保存は非常に重要な操作である．かつ, 本手法では, シェアリングによる解の多様性の維持と VEGA により各目的の最適値が求められる機構を有していると言える．

3.4.8　小林らの非パレート淘汰戦略

パレート最適個体の保存をさらに押し進めたのが小林らの非パレート淘汰戦略である [17]．本手法では, その世代で得られたパレート最適個体全てを保存し, 他は淘汰する．複数の同一の個体は選択しない．

本手法は非常に強力な手法である半面, 個体数が莫大に増加する可能性があり, 選択の途中でパレート最適個体が小数のみであるような問題においては, 精度の良い解が得られない可能性がある．

3.4.9　森らの熱力学遺伝的アルゴリズム

多目的最適化問題において解の多様性の維持は極めて重要である．解の集中を抑え多様性を維持しながら探索を行う手法の一つにシェアリングがある．シェアリングはすでに述べた通り, シエアリング半径の値が非常に重要である．しかしながら, この値は問題に依存し, 決定するのが非常に困難である．

森らは個体のばらつきをエントロピーで評価する TDGA を提案している [18]．

4　GA による多目的最適化の問題点

これまでに説明した GA による多目的最適化手法は非常に強力な手法である半面，次のような問題が存在する．

- **個体数** 先に述べた通り, より良いパレート解を得るためには，多様性の維持が不可欠である．多様性を維持するための最も簡単な手法は多くの個体を利用することである．しかしながら，多くの個体の利用は計算量の増大につながる．逆に個体数が少ないと，単一目的の場合と同様に精度の良い解が得られないだけでなく，パレート解集合の概観が把握できなくなる．

- **シェアリング半径** 多様性の維持を行うためのもう一つのメカニズムがシェアリングである．シェアリングは非常に強力な操作であるが，そのパラメータであるシェアリング半径を決定することは非常に難しい．一般にシェアリング半径の値が解に非常に大きく影響する．

  筆者らはシェアリング半径を以下のような手順で決定している．

  1. 目的空間上で最もユークリッド距離 d の大きな２個体を選出する．
  2. シェアリングレンジ R を決定する．シェアリングレンジ R とは，上記で選んだ 2 個体間にいくつの個体を存在させるかの目標値である．2 個体間に 50 個体を存在させる場合には $R = 50$ となる．
  3. シェアリング半径 σ を次式で決定する．
     $\sigma = d/R^(1/(n-1))$
     ここで n は目的関数の数である．

- **終了条件** 終了条件は解の精度に与える影響の高いパラメータである．多くの研究では世代数を終了条件としている．

  しかしながら，解探索に必要な世代数は問題に依存するのみならず，設定した終了世代が適当な世代数であるかど うかは解が求まった後でなければ判定できず，実用的と言えない．そこで筆者らはパレートフロントを構成するパレート最適個体がこれ以上進展しないと判断された状態の際に終了としている．その手順は以下の通りである．

  1. ランク１の解を保存する．その集合を A とする．
  2. 次世代で得られたランク１の解を B とする．
  3. 集合 A と B を合わせた集合を C とする．集合 C の個体のランキングを再度計算する．
  4. C のうち，上記の操作で得られたランク１の個体と集合 A とを比較する．その際にある一定以上の数の A の個体が含まれていた場合，パレートフロントは前進していないとみなし，シミュレーションは終了する．

- **評価方法** 各手法により得られたパレート解の評価方法は非常に大きな問題である．それは，パレート解は集合であるために，精度だけでなく，より広い範囲での解の存在や，解の分散が大きいことなど，様々な要素が必要となるからである．

  代表的な評価方法としては次のようなものがある．

  – **ランク１の数** 探索終盤では個体のすべてがランク１になるわけではなく，いくつがランク１になる．そのため，ランク１である個体の数は一つの指標となる．

    また，いくつかの手法を比較する場合，それぞれで得られた解のうちランク１を抽出し，複数手法で得られた解のランク１のものと合計する．その集合で，ランキングを行い，その手法の解のランク１のものの数の割合を指標とする．その割合が高ければ，その手法で得られた解が他の手法での解よりも優越していることを示している．

  – **誤差 (error)** 真のパレート解が既知の場合，各パレート最適個体と真のパレート解とのユークリッド距離の平均は誤差とみなせる．誤差が小さいときパレート最適個体が真のパレート解集合に近いことを示している．ただし，この評価基準は真のパレート解が既知の場合でなければ使用できない．

  – **被覆率 (cover rate)** パレート解を探索する場合，たとえ誤差が 0.0 であったとしても１点に集中していては良い解集合とは言えない．そのため，解のばらつきを示す指標が必要となる．その指標が被覆率である．

    まず，Fig.3 に示すように（図では２目的の場合），各目的関数の最大値および最小値を検索し，その間をあらかじめ決めておいた分割数で分割する．それぞれの分割された領域の中に解が存在する場合は 1，存在しない場合には 0 とする．これらの数値を合計し，領域の数で除したものを被覆率とする．よってこの被覆率が 1 に近い方がすべての領域に解が存在していることになり，解が集中することなく全体に行きわたっていることがわかる．

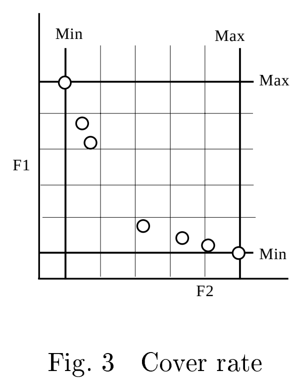


Fig. 3　Cover rate

  – **計算時間もしくは目的関数の計算回数** パレート最適個体の誤差に差が生じない場合には，で

きるだけ短い計算時間もしくは目的関数の値の計算回数ができるだけ少なく解を求めるモデルを評価する．よってこれら計算時間と目的関数の計算回数を指標の一つとして考える．

- **テスト関数** 各手法はいくつものテスト関数に適用されてきたが，これまでに，一般的なテスト関数と呼べるものは存在しない．それぞれのテスト関数はそれぞれに特徴があり，そのテスト関数の特徴に適した手法が良好な結果を示す．
- **表示方法** 先に述べた通り，目的関数の関係を知るために，パレート解集合を求めるのであるが，目的関数の数は 3 以下であれば，同時に表示できる．しかしながら，それ以上になると，同時には表示ができなくなり，なんらかのパレート解を表示する手法が必要となる．
- **並列化モデル** 進化的計算手法は多点による繰り返し計算を多く必要とするため, 計算コストが非常に高い. それを解決する手法の一つが, 進化的計算手法の並列化である. 進化的計算手法は, アルゴリズム自体に並列性を内在するため, 並列化のモデルを検討する必要がある.

  筆者らは GA による多目的最適化の並列化として, 島モデル [19] およびマスタースレーブモデル [20] の提案を行い, 検討している.

5 おわりに

本稿では, 多目的最適化および遺伝的アルゴリズムを中心とした進化的手法による多目的最適化問題の概要および主要な手法の解説, 進化的手法の抱える問題点に触れた.

現在, コンピュータのパワーはますます増大するため, 多目的最適化問題を実用的に解決する必要性が増大するものと考えられる. その際に, 本稿で述べた進化的手法による多目的最適化問題の解決は非常に強力な手法となるであろう. 一方で, 現時点では多くの問題を抱えているため, 今後の研究によりその解決が望まれる.

謝辞

本研究は文部省からの補助を受けた同志社大学の学術フロンティア研究プロジェクト「知能情報科学とその応用」における研究の一環として行った．

参考文献

[1] C. M. Fonseca and P. J. Fleming. *Evolutionary Computation*, 3(1), 1-16 (1995).

[2] H. Tamaki, H. Kita and S. Kobayashi. *Proc. of the 1996 IEEEE Int. Conf. on Evolutionary Computation*, 517-522, (1996).

[3] J. Horn. *Handbook of Evolutionary Conputation. Section F1.9*, oxford Univ. Press., (1997)

[4] C. A. C. Coello. *Proc. of the 1999 Congress of Evolutionary Computation*,3-13 , (1999)

[5] C. A. C. Coello. *http://www.lania.mx/ ccoello/EMOO/.*

[6] *Proc. of the 1999 Congress of Evolutionary Computation*, (1999).

[7] *Proc. of the 2000 Congress of Evolutionary Computation*, (2000).

[8] *http://www.tik.ee.ethz.ch/emo/*

[9] Ben-Tai, A. *Proceedings of Economics and Mathematical Systems*,1-11,(1980).

[10] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, (1989).

[11] C. M. Fonseca and P. J. Fleming. *Proc. of 5th Int. Conf. on Genetic Algorithms*, 416-423, (1993).

[12] J. Horn, N, Nafpliotis and D.E. Goldberg. *Proc. of 1st IEEE Int. Conf. on Evolutionary Computation*, 82-87, (1994).

[13] N. Srinvas and K. Deb. *Evolutionary Computation*,Vol.2, No.3, 221-148, (1994).

[14] 村田, 石渕, 田中. **計測自動制御学会論文集**,Vol.31, NO.5, 583-590, (1997).

[15] J. D. Schaffer. *Proc. of 1st Int. Conf. on Genetic Algorithms and Their Applications*, 93-100, (1985).

[16] 玉置, 森, 荒木. **計測自動制御学会論文集**, Vol. 31, No. 8, 1185–1192,(1995).

[17] 高田, 山村, 小林. **第 *23* 回知能システムシンポジウム** , (1996).

[18] H. Kita, Y. Yabumoto. N. Mori and Y. Nishikawa. *Proc. Parallel Problem Solving from Nature IV*, 504-512, (1997).

[19] 廣安, 三木, 渡邉. **数理モデル化と応用**, Vol.41, No.SIG7, 79-89,(2000).

[20] Watanabe, Hiroyasu, Miki. *Proc. of GECCO 2001(***投稿予定***)*, (2001).