

# Multi-objective Particle Swarm Optimization considering the diversity of the inferior solutions

メタデータ	言語: eng 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	<a href="https://doi.org/10.24517/00007841">https://doi.org/10.24517/00007841</a>

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



# 非劣解の多様性を考慮した多目的Particle Swarm Optimization

北山哲士<sup>\*1</sup> 荒川雅生<sup>\*2</sup> 山崎光悦

## Multi-Objective Particle Swarm Optimization Considering the Diversity of the inferior solutions

Satoshi KITAYAMA, Masao ARAKAWA, Koetsu YAMAZAKI

Department of Human & Mechanical Systems Engineering, Kanazawa University

Kakuma-machi, Kanazawa, 920-1192, Japan

In this paper, a simple method for the multi-objective optimization problems by the Particle Swarm Optimization (PSO) is proposed. The objectives of the Multi-Objective Evolutionary Algorithms (MOEA) are summarized as follows: (1) To find the pareto optimal solutions, (2) To find the pareto optimal solutions as diverse as possible. To achieve these objectives by the PSO for the single objective problems, we propose how to define the g-best in the swarm without introducing some new parameters. That is, one particle among the non-inferior solutions is selected as the g-best to achieve the diversity among the non-inferior solutions. The relative distance in the objective space is utilized to select the g-best among the non-inferior solutions. Additionally, some particles among the non-inferior solutions are also selected as the g-best of the inferior solutions to find the pareto optimal solutions. The absolute distance in the objective space is utilized to select the g-best of the inferior solutions. We also show the geometric interpretation about the movement of particles. The validity of proposed approach is examined through typical numerical examples.

**Key Words :** Engineering Optimization, System Engineering, Multi-Objective Optimization, Particle Swarm Optimization

### 1 緒言

進化的計算法を用いた多目的最適化 (Multi-Objective Evolutionary Algorithms: MOEA) に関する研究が盛んに行われており, 特に遺伝的アルゴリズム (GA) を用いた多目的最適化問題への取り組みに関しては, 文献 (1) に詳しくまとめられている. 文献 (1) によれば, MOEAの主目的は次の2点であるとされている. (MOEA1) 探索点がパレート最適解を見つけること. (MOEA2) パレート最適解を目的関数空間に写像したパレートフロント上に, 可能な限り探索点を分散させること.

計算コストの観点から眺めると, MOEAは必ずしも効率的な方法であるとは言えないものの, 近年のコンピュータおよびネットワーク技術の進歩を考えれば, MOEAは, 一度の計算で多くのパレート最適解を見つけられるという利点があるため, 多目的最適化問題に対する有効な方法の一つであると考えられる.

近年, 進化的計算法の一つであるParticle Swarm Optimization (PSO) を用いた多目的最適化に関する研究が行われつつあり, その有効性が検討されている. そ

れらの研究の一部では, 多目的GAの考え方を継承したもののや<sup>(2,3)</sup>, 関数の感度計算をアルゴリズムに含ませるなどの方法が提案されている<sup>(4,5)</sup>. しかし, PSOは本来, 感度を利用しない最適化手法であり, またその構造上, 陰に探索方向ベクトルを持つなどの性質<sup>(6)</sup> から, アルゴリズム内に感度計算を含ませるようなことは避けるほうがよい. また, PSOは基本的には連続変数の最適化問題に適しており, 確定的な要素の強い最適化手法であることを踏まえれば<sup>(7)</sup>, 設計変数を基本的には0-1で表現し, 再生・交叉・突然変異などの確率的な操作を許容し, 探索点の選択と消滅, 新たな生成等を繰り返す最適化手法であるGAとは, その性質は大きく異なるものであり, PSOの性質を十分理解した上で, 多目的最適化問題への適用を心掛けるほうがよい.

多目的PSOのいくつかの研究では, 後の3.2節で述べるように, 単一目的の下で最適解を探索するPSOに対し, いくつかのパラメータを含ませることにより, パレート最適解およびパレートフロントを見つける方法が多く見受けられる<sup>(8-11)</sup>. しかし, パラメータを増やすことは, 意思決定者の負担を増やし, また試行錯誤的な数値実験の結果からパラメータを決定する必要がある. そのため, 単一目的の下でのPSOを直接的に多目的問題へ適用するには, 若干の修正が必要であり, 必ずしも汎用性のある方法とは言いがたい. さらにPSO

\* 原稿受付 平成??年?月 日

\*<sup>1</sup>正員, 金沢大学工学部 (〒920-1192 金沢市角間町).

\*<sup>2</sup>正員, 香川大学工学部 (〒761-0396 高松市林町2217-20).

は、探索点の更新式がベクトル演算であるため、探索点がどのように移動するかを、幾何学的に説明できるが、従来の研究報告では、このような説明が全く行われていない。すなわち、仮に何らかのパラメータを追加するのであれば、探索点の移動と追加したパラメータの意味や影響を幾何学的に説明する必要があるが、そのような報告はなされていない。

そこで本論文では、単一目的の下でのPSOを遵守する立場をとり、単一目的の下でのPSOに対して、新たなパラメータを追加することなく、多目的最適化問題へ拡張でき、かつパレートフロント上の探索点間の多様性を考慮した多目的Particle Swarm Optimization (Multi-Objective PSO: MOPSO)を提案する。また提案するMOPSOにおいて、探索点がどのように移動するかを、幾何学的に説明する。数値計算を通じて、その有効性を検討する。

## 2 Particle Swarm Optimization

PSOは、探索点 (Particle) が持つ最良の情報 (p-best) と、探索点から形成されるグループ (Swarm) の最適値 (g-best) から、過去の探索履歴を考慮して連続変数の多峰性関数の大域的最適解もしくは次善の局所的最適解を高い精度で求めることができる手法であり、集団はg-bestへ向う。PSOでは各探索点が「位置」と「速度」を持ち、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。

**2.1 位置と速度の更新**  $t$  回目の探索において、探索点  $d$  の位置  $\mathbf{x}_d^t$  と速度  $\mathbf{v}_d^t$  を用いて、 $t+1$  回目の位置  $\mathbf{x}_d^{t+1}$  と速度  $\mathbf{v}_d^{t+1}$  は、次の式を用いて更新される。

$$\mathbf{x}_d^{t+1} = \mathbf{x}_d^t + \mathbf{v}_d^{t+1} \quad (1)$$

$$\mathbf{v}_d^{t+1} = w\mathbf{v}_d^t + c_1r_1(\mathbf{p}_d^t - \mathbf{x}_d^t) + c_2r_2(\mathbf{p}_g^t - \mathbf{x}_d^t) \quad (2)$$

式 (2) 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

$$c_1 + c_2 \leq 4 \quad (3)$$

となるよう、 $c_1$  と  $c_2$  は決められる必要があるが、 $c_1 = c_2 = 2$  が一般には用いられる。また  $w$  は慣性項と呼ばれるパラメータである。 $\mathbf{p}_d^t$  は、探索点  $d$  が  $t$  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、 $\mathbf{p}_g^t$  は  $t$  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 (1), (2) 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 (2) 中の  $\mathbf{p}_g^t$  を、 $t$  回目までの探索で目的関数値を最良にした探索点  $\mathbf{p}_g$  で置き換えたモデルは、最良値保存型モデルと呼ばれる。

**2.2 基本アルゴリズム** 単一の目的関数を最小化 (もしくは最大化) するPSO (g-bestモデル) の基本ア

ルゴリズムは、一般に次のようになる。

(STEP1) 探索点数, 最大探索回数  $t_{\max}$  を決定。また側面制約条件を設定。

(STEP2) 各探索点に対して, ランダムに初期位置  $\mathbf{x}_d^t$  と初期速度  $\mathbf{v}_d^t$  を決定。  $t=1$  とする。

(STEP3) 各探索点に対して, 関数値を計算。

(STEP4)  $\mathbf{p}_d^t$  と  $\mathbf{p}_g^t$  を求める。

(STEP5) 各探索点の速度と位置を式 (1), (2) に従い更新。また慣性項  $w$  を以下の式に従い更新。

$$w = w_{\max} - (w_{\max} - w_{\min}) / t_{\max} \times t \quad (4)$$

一般には  $w_{\max} = 0.9$ ,  $w_{\min} = 0.4$  が用いられており、慣性項を線形的に減少させることにより、探索点の多様性と集中化をコントロールしている<sup>(12)</sup>。

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

最良値保存モデルの場合は式 (1) 中の  $\mathbf{p}_g^t$  を  $\mathbf{p}_g$  で置き換えればよい。

## 3 多目的最適化問題とMOPSO

**3.1 多目的最適化問題**  $k$  個の目的関数を最小化する多目的最適化問題は次のように定式化される。

$$\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \rightarrow \min \quad (5)$$

$$x_i^L \leq x_i \leq x_i^U \quad i=1, 2, \dots, n \quad (6)$$

$$g_j(\mathbf{x}) \leq 0 \quad j=1, 2, \dots, m \quad (7)$$

$f_i(\mathbf{x})$  は  $i$  番目の目的関数、 $\mathbf{x}$  は設計変数ベクトルであり、 $n$  は設計変数の数を表す。また  $x_i^L$  と  $x_i^U$  は  $i$  番目の設計変数に直接的に課される側面制約条件であり、 $g_j(\mathbf{x})$  は挙動制約条件、 $m$  はその数を表す。

多目的最適化問題では、目的関数間にトレードオフ関係が存在する時は、複数の目的関数を同時に最小化する解は一般には存在せず、パレート最適解の概念が導入される。多点同時探索型多目的最適化問題では、探索点間の優劣関係が存在し、 $i$  番目の探索点の設計変数ベクトルを  $\mathbf{x}_i$  とし、 $j$  番目の探索点の設計変数ベクトルを  $\mathbf{x}_j$  とすると、(ただし  $i \neq j$ )

$$f_{io}(\mathbf{x}_i) \leq f_{io}(\mathbf{x}_j) \quad \forall i, o=1, 2, \dots, k \quad (8)$$

が成立する場合、 $\mathbf{x}_i$  は  $\mathbf{x}_j$  に対して、優越するといい、 $\mathbf{x}_i$  を非劣解 (non-inferior solution),  $\mathbf{x}_j$  を劣解 (inferior solution) と呼ぶ。また、ある探索点  $\mathbf{x}^{prt}$  に対して、優越する探索点の実行可能領域内にない場合、 $\mathbf{x}^{prt}$  はパレート最適解と呼ばれる。

**3.2 MOPSO** 本節では代表的なMOPSOについて、その概略と問題点について、簡単にまとめる。詳細は各文献を参照されたい。

**(a) Huらの方法**<sup>(8)</sup> : 2目的関数に対して、提案されている。ある特定の探索点に注目し、

(STEP1) 1番目の目的関数を固定する。

(STEP2) 目的関数空間において、固定された目的関数の近傍にある  $m$  個の探索点を選ぶ。(文献(6)では、 $m=2$  としている。)

(STEP3)  $m$  個の探索点が、2番目の目的関数の最小値を探索する。

(STEP4) 次に、2番目の目的関数を固定する。

(STEP5) (STEP2)と同じ。

(STEP6)  $m$  個の探索点が、1番目の目的関数の最小値を探索する。

この操作をすべての探索点に対して、適用することにより、パレートフロントを見つける方法である。その概略を図1に示す。

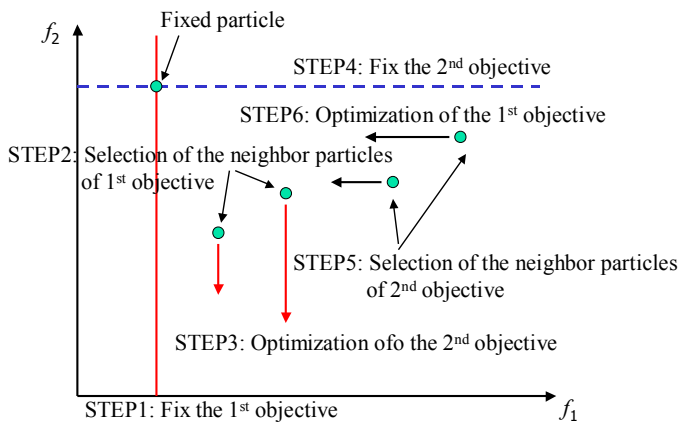


Fig.1 Concept of Hu's MOPSO

この方法は、計算コストが莫大であり、また特定の目的関数を固定しているため、複数の目的関数を同時に評価していない。さらにパラメータ  $m$  に関しては、 $m=2$  しか試されておらず、探索点間の分散メカニズムが導入されていない。また固定された目的関数は、制約条件にはなっていないものの、実質的には  $\epsilon$  制約法を用いたPSOと捉えることもできる。<sup>(9)</sup>

(b) **Parsopoulosらの方法 (VEPSO)** : ベクトル評価型GA (VEGA)と同様に、目的関数の数だけ、サブ集団を作り、サブ集団が特定の目的関数のみを評価する。 $g$ -bestは集団の数(すなわち目的関数の数)だけ存在する。探索点間のランダムなシャッフルにより、探索点間の分散を達成している。問題点とし、VEPSOではVEGAと同様に、すべての探索点が複数の目的関数を同時に評価できないこと、各目的関数の最小値に探索点が集中することがあること、また、(探索点数) / (目的関数の数) が整数値でなければならないことなどが挙げられる。<sup>(10)</sup>

(c) **Fieldsendらの方法** : 非劣解もしくはパレート最適解となる探索点が、目的関数空間において、ある矩形の領域を作成する。そしてその矩形領域内にある劣

解となる探索点は、非劣解もしくはパレート最適解となる探索点を  $g$ -bestとして、移動する。(図2)そのため、 $g$ -bestは複数存在する。アーカイブサイズが決められていないため、多くの探索点がパレートフロントへ到達できるが、パレートフロント上にある探索点間の分散メカニズムはない。

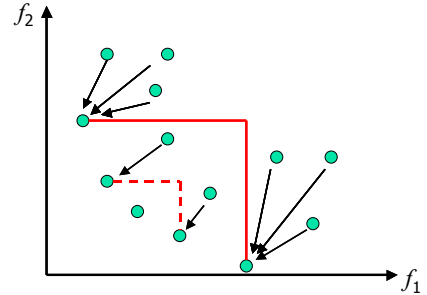


Fig.2 Concept of Fieldsend's MOPSO

(d) **Mostaghimらの方法** <sup>(11)</sup> : sigma法と称する方法を用いており、目的関数空間において、原点と非劣解もしくはパレート最適解となる探索点を直線で結ぶ。そして、図3に示すように、劣解となる探索点は、目的関数空間の近傍に存在する非劣解もしくはパレート最適解となる探索点を  $g$ -bestとして、移動する。SPEA等と同様にアーカイブサイズが決められており、クラスタリングにより、パレートフロント上にある探索点間の分散を達成している。

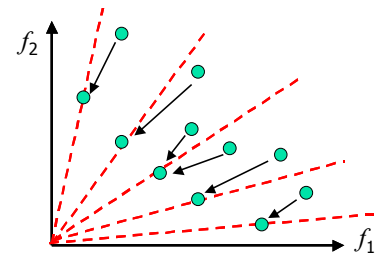


Fig.3 Concept of Mostaghim's MOPSO

### 3.3 従来のMOPSOのまとめ

3.2節での検討から、従来のMOPSOの問題点をまとめると、次のようになる。

(D1) 探索点集団に対し、 $g$ -bestを1つだけにすれば、 $g$ -bestが頻繁に更新される可能性があり、結果的に収束が悪化する。

(D2) 3.2節の(a)や(b)のような方法では、複数の目的を同時に評価できないという問題がある。

(D3) 文献(11)のようにアーカイブサイズを決めれば、パレートフロント上に集まる探索点数が限られ、アーカイブサイズを越えた場合、クラスタリングなどの操作が必要となり、この際、クラスタの数の決定が問題となる。

また、上記(a)~(d)のMOPSOに対する疑問点は、次

のようになる。

(CD1) 第2章で記述したPSO以外のパラメータを導入しなければいけない<sup>(8), (11)</sup>。

(CD2) 第2章で記述したPSOとは異なり、初期位置や初期速度をゼロとしているものもある<sup>(2), (11)</sup>。

(CD3) 慣性項を一定としている。文献(12), (15)で述べられているように、慣性項は探索点間の多様性と集中化をコントロールするパラメータであるが、これを無視している。

上記の理由より、単一目的と多目的の場合で、異なるPSOを適用しなければならず、結果的にPSOの汎用性を妨げていると考えられる。

ただし、これらのMOPSOにおいて、積極的に活用すべき事柄は次のようにまとめられる。

(A1) MOPSOの性能向上の重要な要素の一つは、g-bestの選び方とその数である<sup>(10), (11)</sup>。そこで、パレートフロント上にp-bestがない探索点に対しては、パレートフロント上にある探索点をg-bestとして、複数のg-bestを許容する。つまり、パレートフロント上にp-bestがない探索点に対しては、探索点毎にg-bestを持たせばよい。これにより、多くの探索点がパレートフロントへ到達できる可能性があり、また収束が改善される可能性がある。

(A2) アーカイブサイズを決めないこと<sup>(10)</sup>。クラスタリング等の複雑な操作が必要ない。

## 4 提案するMOPSO

第2章で記述した単一目的の下でのPSOを遵守する形のMOPSOを開発する。本論文で提案するMOPSOでは、第3章の(A1), (A2)を活用しており、さらに従来の研究にはない、非劣解となる探索点に対し、目的関数空間における相対距離を用いた明確な分散メカニズムが導入されているところに特徴がある。はじめにアルゴリズムを図4に示し、以下、その説明をする。

**4.1 パレートフロントの生成(STEP1)** パレートフロントは、p-bestの中で、非劣解となる探索点が生成する。例えば図5に示すように5つの探索点のp-bestが、目的関数空間において位置する場合、パレートフロントを形成するp-bestは、探索点1, 3, 4となる。同図の探索点1, 3, 4を結ぶ破線は、パレートフロントを示す。

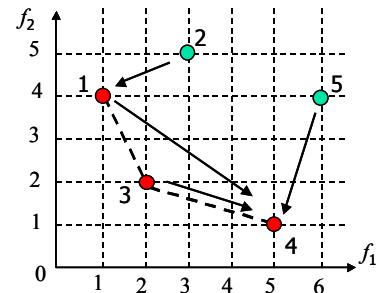


Fig.5 Pareto front by the non-inferior solutions

**4.2 パレートフロント上のg-best(STEP2)** パレートフロントを構成するp-bestが  $p$  個あり、それらを  $p_{d,i}^{PF}$  ( $i=1,2,\dots,p$ ) と表記する。このとき、 $p_{d,i}^{PF}$  間の

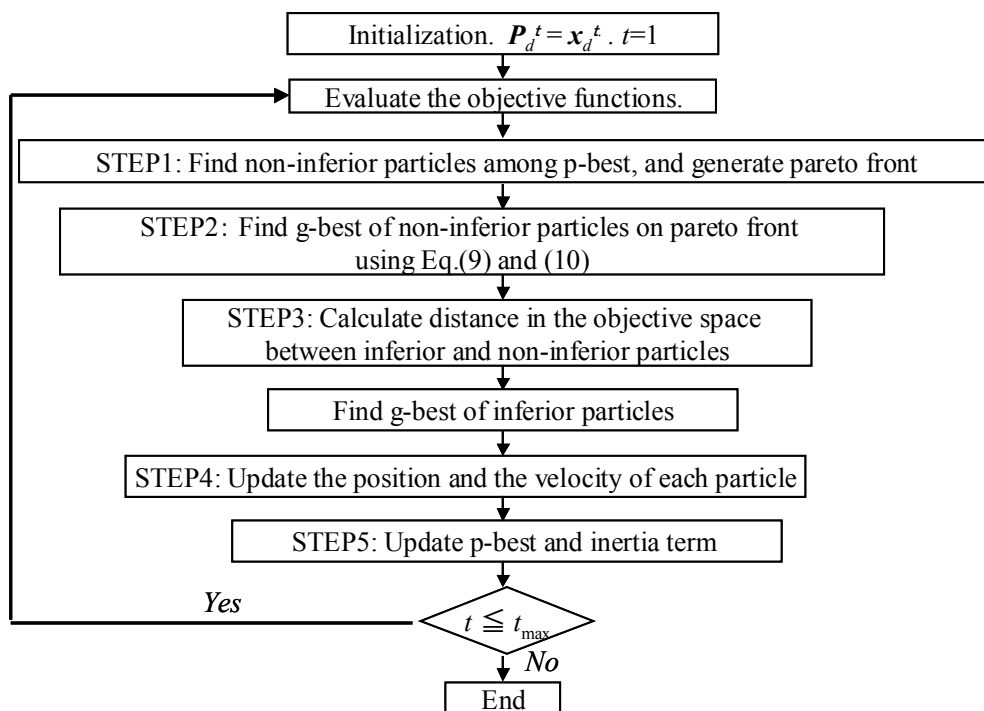


Fig.4 The algorithm of proposed MOPSO

目的関数空間でのばらつきを評価するため、相対距離を以下の式で測定する。

$$d_i = \min_{j=p \wedge i \neq j} \left( \sum_{io=1}^k |f_{io}(\mathbf{p}_{d,i}^{PF}) - f_{io}(\mathbf{p}_{d,j}^{PF})| \right) \quad i=1,2,\dots,p \quad (9)$$

そして、式 (9) が最大となる探索点

$$\mathbf{p}_g^{PF} = \max \{d_i\} \quad i=1,2,\dots,p \quad (10)$$

を  $\mathbf{p}_{d,i}^{PF}$  の g-best とする。すなわち、 $\mathbf{p}_{d,i}^{PF}$  の g-best は1つである。図5の例では、 $\mathbf{p}_{d,i}^{PF}$  は探索点1, 3, 4であり、式 (9), (10) から、探索点4が探索点1, 3の g-best となる。式 (9) は探索点間のばらつきを評価する一つの指標である、spacing を求める際に用いられる式であり、本論文では、パレートフロントを構成する探索点の中で最もばらついていると判断された p-best を、 $\mathbf{p}_{d,i}^{PF}$  の g-best とした。

**4.3 パレートフロントに p-best がいない探索点の g-best (STEP3)** パレートフロント上に p-best がいない探索点の g-best は、3.3節 (A1) に従い、次のように決めることにした。すなわち、パレートフロント上に p-best がいない探索点が  $q$  個あり、それらを  $\mathbf{x}_j^{NP}$  ( $j=1,2,\dots,q$ ) とする。  $j$  番目の探索点に対し、 $\mathbf{p}_{d,i}^{PF}$  ( $i=1,2,\dots,p$ ) との目的関数空間での距離

$$D_{j,i} = \sum_{io=1}^k \sqrt{(f_{io}(\mathbf{x}_j^{NP}) - f_{io}(\mathbf{p}_{d,i}^{PF}))^2} \quad i=1,2,\dots,p \quad (11)$$

を計算し、式 (11) が最小となる  $\mathbf{p}_{d,i}^{PF}$

$$\mathbf{p}_{g,j} = \min \{D_{j,i}\} \quad (12)$$

を  $j$  番目の探索点の g-best  $\mathbf{p}_{g,j}$  とする。つまり、 $\mathbf{p}_{g,j}$  は、目的関数空間において、最近隣のパレートフロント上の探索点となる。

**4.4 位置と速度の更新 (STEP4)** はじめに p-best がパレートフロント上にある探索点について考える。位置に関しては式 (1) を用い、速度に関しては、次式を用いる。

$$\mathbf{v}_d^{t+1} = w\mathbf{v}_d^t + c_1 r_1 (\mathbf{p}_{d,i}^{PF} - \mathbf{x}_d^t) + c_2 r_2 (\mathbf{p}_g^{PF} - \mathbf{x}_d^t) \quad (13)$$

一方、p-best がパレートフロント上にない探索点について、位置に関しては式 (1) を用い、速度に関しては、次式を用いる。

$$\mathbf{v}_d^{t+1} = w\mathbf{v}_d^t + c_1 r_1 (\mathbf{p}_{d,i}^{PF} - \mathbf{x}_d^t) + c_2 r_2 (\mathbf{p}_{g,j} - \mathbf{x}_d^t) \quad (14)$$

$$j=1,2,\dots,q$$

**4.5 p-best の更新 (STEP5)** p-best の更新は、図6左に示すように

$$f_{io}(\mathbf{x}_d^{t+1}) \leq f_{io}(\mathbf{p}_d^t) \quad \forall io=1,2,\dots,k$$

$$\rightarrow \mathbf{p}_d^{t+1} = \mathbf{x}_d^{t+1} \quad (15)$$

のときに p-best を更新する。これは図6右に示すような場合、いわゆる、もぐらたたき現象が起こり、結果的には探索点のパレートフロントへの収束に影響を及ぼ

すためである。慣性項の更新は、式 (4) を用いる。

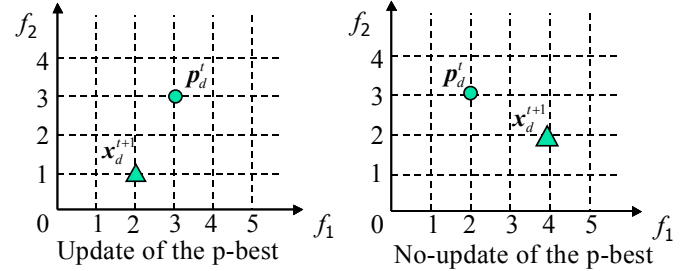


Fig.6 Update scheme of the p-best

**4.6 幾何学的な解釈** 単一目的の PSO は、式 (1), (2) で表されるように、単純なベクトル演算で探索点を更新するため、幾何学的な解釈が可能であり、多目的 PSO を開発するのであれば、幾何学的な解釈は必要であり、また重要である。

図7に示すように、2つの探索点の p-best がパレートフロント上にあるとする。このとき、 $t$  回目の探索における位置をそれぞれ  $\mathbf{x}_1^t$ ,  $\mathbf{x}_2^t$  とすれば、それらは g-best  $\mathbf{p}_g^{PF}$  を共有している。また  $\mathbf{x}_1^t$ ,  $\mathbf{x}_2^t$  の p-best をそれぞれ  $\mathbf{p}_1^{PF}$  と  $\mathbf{p}_2^{PF}$  と表記すると、式 (1) は

$$\mathbf{v}_i^t = \mathbf{x}_i^t - \mathbf{x}_i^{t-1} \quad i=1,2 \quad (16)$$

となり、式 (13) の右辺のベクトル項を抽出すると、

$$\mathbf{p}_i^{PF} - \mathbf{x}_i^t \quad (\text{p-bestに関する項}) \quad i=1,2 \quad (17)$$

$$\mathbf{p}_g^{PF} - \mathbf{x}_i^t \quad (\text{g-bestに関する項}) \quad i=1,2 \quad (18)$$

となる。よって、各ベクトルは図7に示すような向きに生成される。

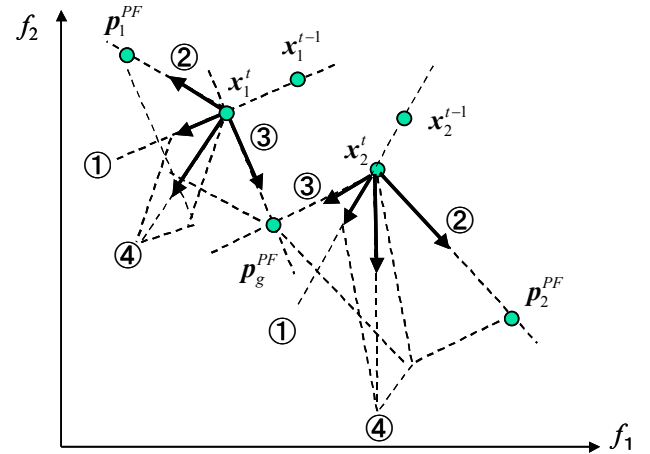


Fig.7 Geometrical interpretation of the particle on the pareto front

$\mathbf{v}_i^t = (\mathbf{x}_i^t - \mathbf{x}_i^{t-1})$ ,  $i=1,2$  : 図7中①の向き。

$(\mathbf{p}_i^{PF} - \mathbf{x}_i^t)$ ,  $i=1,2$  : 図7中②の向き。

$(\mathbf{p}_g^{PF} - \mathbf{x}_i^t)$ ,  $i=1,2$  : 図7中③の向き。

最終的にこれらの線形和で、 $t+1$  回目の探索点の移動方向が決まる。 $\mathbf{x}_1^t$ ,  $\mathbf{x}_2^t$  は図7中④の向きにベクトルが生成されるため、これらの点は大域的なパレートフロントを探索することが予想される。

次に図8に示すように、p-best がパレートフロント上にない探索点について考える。 $t$  回目の探索における

探索点の位置を  $\mathbf{x}_3^t$ , その点のp-bestを  $\mathbf{p}_3^t$  とする. また, p-bestがパレートフロント上にない探索点のg-bestは, 最近隣のパレートフロント上の探索点となり, それを  $\mathbf{p}_{g,3}$  とし, 図8中の▲で表す. この時, 速度ベクトルは,

$$\mathbf{v}_3^t = \mathbf{x}_3^t - \mathbf{x}_3^{t-1} \quad (19)$$

となり, 式 (14) 中のベクトル項を抽出すると, 次のようになる.

$$\mathbf{p}_3^t - \mathbf{x}_3^t \quad (20)$$

$$\mathbf{p}_{g,3} - \mathbf{x}_3^t \quad (21)$$

よって, p-bestがパレートフロント上にない探索点の各ベクトルは次のような向きに生成される.

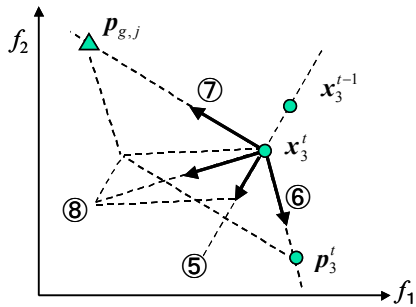


Fig.8 Geometrical interpretation of the inferior solutions

$\mathbf{v}_3^t = (\mathbf{x}_3^t - \mathbf{x}_3^{t-1})$ : 図8中⑤の向き.

$(\mathbf{p}_3^t - \mathbf{x}_3^t)$ : 図8中⑥の向き.

$(\mathbf{p}_{g,3} - \mathbf{x}_3^t)$ : 図8中⑦の向き.

これらの線形和により,  $t+1$  回目の探索点の移動方向が決まるため,  $\mathbf{x}_3^t$  は図8中⑧の向きへ移動する. これより, p-bestがパレートフロント上にない探索点は, パレートフロントへ到達する方向へ移動することが予想される.

探索点は, 図7, 8に示すように目的関数空間を移動するため, いくつかの探索点が探索初期段階に一部のパレートフロントに集まってしまった場合は, 結果的に, 多くの探索点がパレートフロントに到達する可能性があるものの, 一部に密集してしまい, 必ずしも探索点がパレートフロント上に一様に分散するとは限らない. これは図7, 8に示すように, 探索点が探索点間の内分点を常に取りるようにするように移動し, 外分点を取るようには移動しないためである.

## 5 数値計算例

数値計算例を通じて, 本論文で提案するMOPSOの有効性を検討する.

**5.1 無制約多目的最適化問題** 以下に示す2目的の最小化問題を取り扱う.

$$f_1(\mathbf{x}) = x_1 \rightarrow \min \quad (22)$$

$$f_2(\mathbf{x}) = g(\mathbf{x}) \times h(\mathbf{x}) \rightarrow \min \quad (23)$$

Table 1 Functions of  $g(\mathbf{x})$  and  $h(\mathbf{x})$

Problem	$g(\mathbf{x})$	$h(\mathbf{x})$
ZDT1	$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$	$h(\mathbf{x}) = 1 - \sqrt{f_1(\mathbf{x})/g(\mathbf{x})}$
ZDT2	$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$	$h(\mathbf{x}) = 1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2$
ZDT3	$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$	$h(\mathbf{x}) = 1 - \sqrt{f_1(\mathbf{x})/g(\mathbf{x})} - (f_1(\mathbf{x})/g(\mathbf{x})) \sin(10\pi f_1(\mathbf{x}))$

$$0 \leq x_i \leq 1 \quad i=1,2,\dots,n \quad (24)$$

ここで  $g(\mathbf{x})$  と  $h(\mathbf{x})$  は表1に示す通りである. 本節で扱う問題のパレート最適解はすべて,

$$0 \leq x_i^{opt} \leq 1 \quad (25)$$

$$x_i^{opt} = 0 \quad (26)$$

である. 設計変数の数を, すべて  $n=10$  とし, 探索点数を100, 最大探索回数を500とした. 上記の問題に共通している点は,  $f_1(\mathbf{x})$  からわかるように, 探索点をパレートフロント上に幅広く分散させるメカニズムがアルゴリズム内に含まれているかどうかを検査できる点である. またパレートフロントの形状が連続かつ凸のもの (ZDT1), 連続かつ非凸のもの (ZDT2), 不連続 (ZDT3) のものである. パレートフロントについては, 文献 (1) を参照されたい.

本論文で提案するMOPSOを乱数の種を変更させ, 20回試行したとき, 最良と考えられる結果を図9(a)~(c)に, 最悪と考えられる結果を図10(a)~(c)に, それぞれ示す. 各図の中の●は, すべての探索点を表しており, 図中の数値  $pa$  はパレートフロントへ到達した探索点数を表している. 文献 (1) 等では, パレートフロント上の探索点のみが表示されており, 他の探索点が目的関数空間のどのような位置に存在するかが示されていないため, あえてすべての探索点を表示している. なお, 最良および最悪と判断した理由として, 定量的にはパレートフロントへ到達した探索点の数, 定性的にはパレートフロントに到達した探索点が幅広く分散しているかを判断基準とした. 探索点が一部に密集するような場合は, 緒言で述べた (MOEA2) に反するため, 最悪と判断した.

**5.2 有制約多目的最適化問題** 文献 (1) で用いられている以下の2変数2目的からなる有制約多目的最適化問題を扱う.

**BNH:**

$$f_1(\mathbf{x}) = 4x_1^2 + 4x_2^2 \rightarrow \min \quad (27)$$

$$f_2(\mathbf{x}) = (x_1 - 5)^2 + (x_2 - 5)^2 \rightarrow \min \quad (28)$$

$$g_1(\mathbf{x}) = (x_1 - 5)^2 + x_2^2 - 25 \leq 0 \quad (29)$$

$$g_2(\mathbf{x}) = -(x_1 - 8)^2 - (x_2 + 3)^2 + 7.7 \leq 0 \quad (30)$$

$$0 \leq x_1 \leq 5, \quad 0 \leq x_2 \leq 3 \quad (31)$$

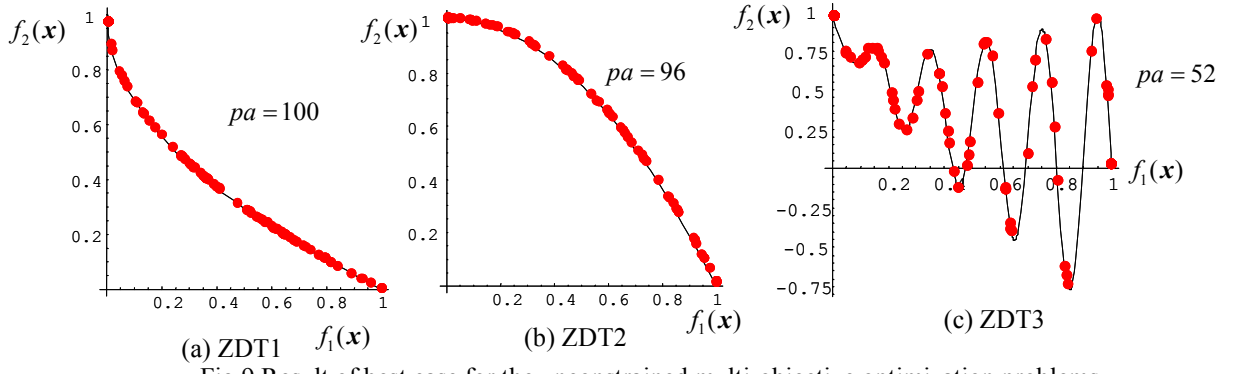


Fig.9 Result of best case for the unconstrained multi-objective optimization problems

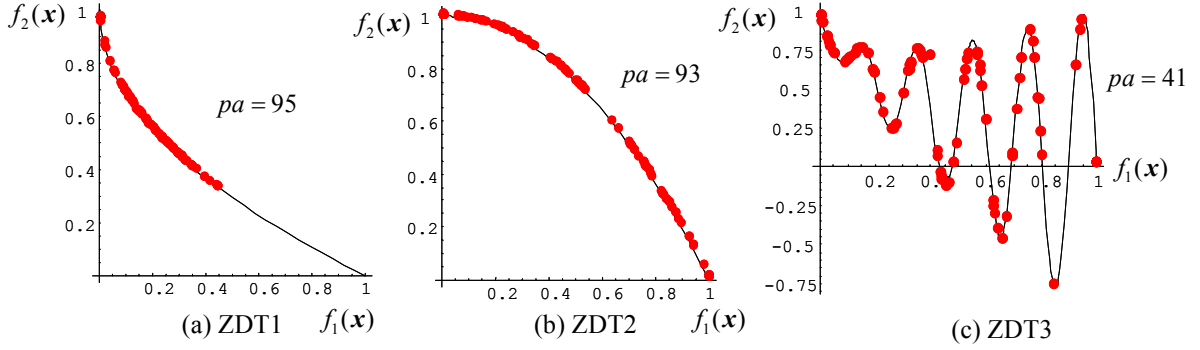


Fig.10 Result of worst case for the unconstrained multi-objective optimization problems

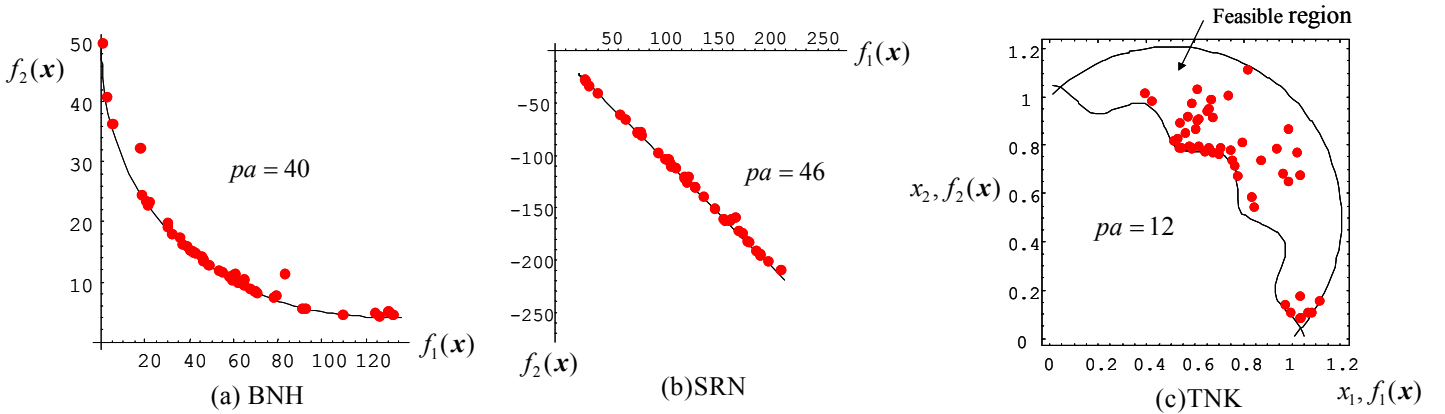


Fig.11 Result of best case for the constrained multi-objective optimization problems

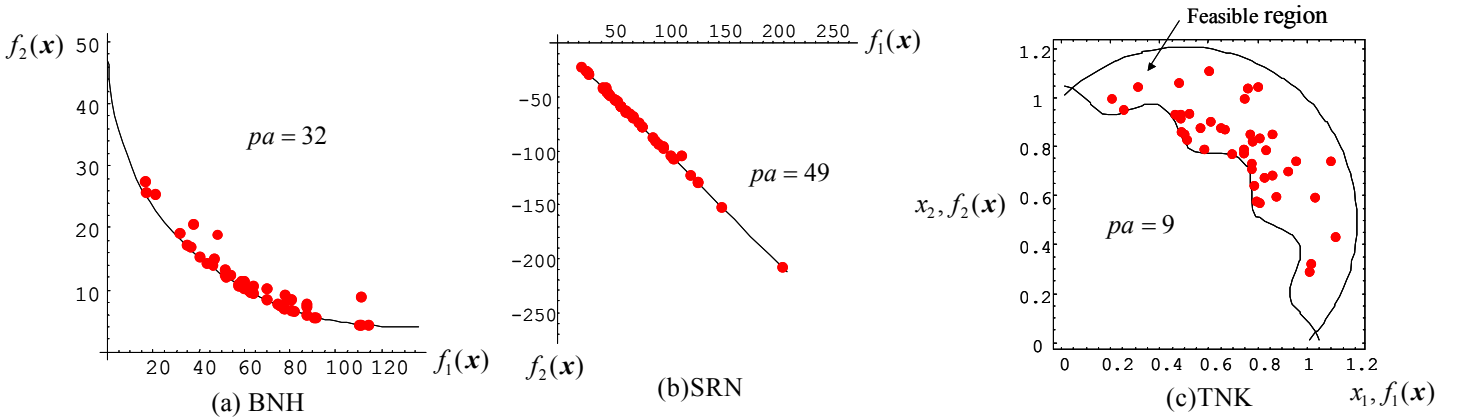


Fig.12 Result of worst case for the constrained multi-objective optimization problems

**SRN:**

$$f_1(\mathbf{x}) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2 \rightarrow \min \quad (32)$$

$$f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2 \rightarrow \min \quad (33)$$

$$g_1(\mathbf{x}) = x_1^2 + x_2^2 - 225 \leq 0 \quad (34)$$

$$g_2(\mathbf{x}) = x_1 - 3x_2 + 10 \leq 0 \quad (35)$$

$$-20 \leq x_1, x_2 \leq 20 \quad (36)$$

**TNK:**

$$f_1(\mathbf{x}) = x_1 \rightarrow \min \quad (37)$$

$$f_2(\mathbf{x}) = x_2 \rightarrow \min \quad (38)$$

$$g_1(\mathbf{x}) = -x_1^2 - x_2^2 + 1 + 0.1 \cos[16 \tan^{-1}(\frac{x_1}{x_2})] \leq 0 \quad (39)$$



$$g_2(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.5 \leq 0 \quad (40)$$

$$0 \leq x_1, x_2 \leq \pi \quad (41)$$

上記3つの問題において、パレートフロントの形状はそれぞれ、下に凸 (BNH)、直線的 (SRN) であり、また非凸かつ分離 (TNK) している。また挙動制約条件に関しては、文献(7)で提案した方法を用い、挙動制約条件をペナルティ関数として取扱い、事実上の無制約多目的最適化問題に変換した。探索点数を50、最大探索回数を200と設定し、側面制約条件内に探索点をランダムに発生させ、20回試行した時、最良と考えられる結果を図11(a)~(c)に、最悪と考えられる結果を図12(a)~(c)に、それぞれ示す。同図には、5.1節と同様に、すべての探索点を●で表示してある。各図の中の赤丸は、すべての探索点を表しており、図中の数値  $pa$  はパレートフロントへ到達した探索点数を表している。また、最良もしくは最悪と判断とした理由としては、探索点が幅広くパレートフロントへ分散しているかを定性的に判断した。

## 6 考察

本章では、パレートフロントが連続である場合と不連続である場合について、それぞれ考察する。

**6.1 連続の場合** パレートフロントが連続の場合、提案したMOPSOでは、多くの探索点がパレートフロントへ到達している。しかし、同時にPSOの探索能力の高さ故に、図10のZDT1では、探索点が一部に密集してしまう場合もある。また図10のZDT2では、パレートフロントのおおまかな形状は把握できるものの、探索点がない部分も存在する。これは、4.6節で述べた通り、探索点の移動が探索点間の内分点を取るよう移動し、外分点を取るよう移動しないためである。また有制約多目的最適化問題において、BNHとSRNでは、いくつかの探索点はパレートフロントに到達していなかったり、一部に探索点が集中しているものの、比較的多くの探索点がパレートフロントに到達しており、パレートフロントの大きな形状が把握できていると思われる。

**6.2 不連続の場合** 一方、パレートフロントが不連続の場合、パレートフロントへ到達した探索点の数が、連続の場合と比較すると激減している。特に有制約最適化問題におけるTNKでは、実行可能領域が島状となっているため、探索点の実行可能領域を見つけるまでに多くの探索を要してしまったため、パレートフロントを見つけることが困難であった。またパレートフロントへ到達しなかった原因の一つとして、提案するMOPSOでは、目的関数空間におけるパレートフ

ロント上の探索点に対して、多様性 (ばらつき) を評価しており、設計変数空間における探索点の多様性 (ばらつき) までは評価していないことが考えられる。つまり、パレートフロントが連続であれば、パレートフロントへ到達した探索点は、パレートフロント上を動くことが考えられるが、不連続である場合、パレートフロントをジャンプしなければならない、本論文で提案した方法では、パレートフロント上の探索点のジャンプが困難であったことを示す結果となった。目的関数空間と設計変数空間の両方において、多様性を維持するようなアーカイブメカニズムにより、パレートフロントが分離している問題でも、比較的良好な結果が得られることが報告されている<sup>(14)</sup>。ことを考えれば、設計変数空間における探索点間の多様性を考慮することも有効な一つの方法であると考ええる。

**6.3 今後の検討課題** 6.2節および6.3節より、今後の検討課題をまとめると次のようになる。

(1) いくつかの探索点が、探索初期段階でパレートフロントへ到達した場合、パレートフロント上に探索点を幅広く分散させるための分散メカニズムの考案。

(2) 目的関数空間と設計変数空間の両方において、探索点のパレートフロントへの収束を考慮しつつ、多様性を維持するような分散メカニズムの導入。

(3) 本論文は、従来のMOPSOの研究成果から得られた知見を基に、単一目的関数の下でのPSOを遵守する形で多目的PSOのアルゴリズムを開発しており、他のMOPSOのとの優位性や効率性を検討した研究ではない。これらの点に関しては、今後の検討課題としたい。

## 7 結言

本論文では、単一目的関数の下でのPSOを遵守する立場で、非劣解もしくはパレート最適解となる探索点に対して多様性をもたせつつ、劣解となる探索点に対してはパレート最適解となるようなMOPSOを提案した。提案したMOPSOは、従来の方法のような特別なパラメータを導入することのない汎用性のある方法であると考ええる。また、探索点の移動に関して、幾何学的な解釈を与えた。

ただし、本論文で考えた多様性の維持に関しては、目的関数空間のみで考えているので、設計変数空間での多様性維持のメカニズムの検討が必要である。

本研究を行うにあたり、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)、山川宏先生 (早稲田大学) に感謝したい。

## 参考文献

- (1) Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*, (2001), Wiley..
- (2) Coello, C., et.al., Handling Multiple Objectives with Particle Swarm Optimization, *IEEE Trans. on Evolutionary Computation*, Vol.8, No.3, (2004), pp.256-279.
- (3) Ho, S.L., et.al., A Particle Swarm Optimization-Based Method for Multiobjective Design Optimization, *IEEE Trans. on Magnetics*, Vol.41, No.5, (2005), pp.1756-1759.
- (4) Izui, K., et. al., Hybrid Swarm Optimization Techniques Incorporating Design Sensitivities, *Trans. of the Japan Society of Mechanical Engineers, Series C*, Vol.72, No.719, (2006), pp.2264-2271 (in Japanese)
- (5) Baumgartner, U., et.al., Pareto Optimality and Particle Swarm Optimization, *IEEE Trans. on Magnetics*, Vol.40, No.2, (2004), pp.1172-1175.
- (6) Kitayama, S., et.al., Penalty Function Approach for the Mixed Discrete Non-Linear Problems by Particle Swarm Optimization, *Structural and Multidisciplinary Optimization*, Vol.32, No.3, (2006), pp.191-202.
- (7) Kitayama, S., et.al., 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.
- (8) Hu, X., and Eberhart, R., Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization, *IEEE Proceedings, World Congress on Computational Intelligence*, (2002), pp.1677-1681.
- (9) Parsopoulos, K.E., and Vrahatis, M.N., Particle Swarm Optimization Method in Multiobjective Problems, *Proceedings of the 2002 ACM Symposium on Applied Computing*, (2002), pp.603-607.
- (10) Fieldsend, J., Multi-Objective Particle Swarm Optimization Methods, *Technical Report No.419, Dept. of Computer Science, University of Exeter*, (2004).
- (11) Mostaghim, S., and Teich, J., Strategies for Finding Good Local Guides in Multi-objective Particle Swarm Optimization(MOPSO), *IEEE 2003 Swarm Intelligence Symposium*, (2003), pp.26-33.
- (12) Yasuda, K., and Ishigame, A., Nonlinear Programming Algorithm - from the point of view of practical applications-, *System, Control, and Information Engineers*, Vol.50, No.9, (2006), pp.1-7. ( in Japanese)
- (13) Kitamura, S., et.al., Modified Multi-objective Particle Swarm Optimization Method and its Application to Energy Management System for Factories, *Trans. IEE Japan*, Vol.125-C, No.1, (2005), pp.21-28.
- (14) Kim, M., et.al., The Dual-Archive Scheme which maintains the diversity of the solutions in the objective space and decision variable space, *The Information Processing Society of Japan, Trans. on Mathematical Modeling and its Applications (TOM)*, Vol. 46, No.SIG 17, pp.102-113, (in Japanese).
- (15) Aiyoshi, E., Yasuda, K., ed., *Metaheuristics and Applications*, pp.69-90, (2007), The Institute of Electrical Engineers of Japan. ( in Japanese)