

# 複数の PBIL 探索エージェントによる 離散最適化問題の解法に関する研究

加藤 康彦（熊本学園大学）

A Society of Population Based Incremental Learners  
for Discrete Optimization Problems  
Kumamoto Gakuen University Yasuhiko KATO

## [Abstract]

In past several decades, evolutionary algorithms have been successfully applied in many studies and a wide variety of application areas. This paper presents a new evolutionary algorithm, which is a kind of probabilistic model based genetic algorithms. We designate our proposed algorithm as a multi-agent system called “A Society of Population Based Incremental Learners (SPBIL)”. To explain it in short, intelligent agents with Population Based Incremental Learning (PBIL) move around in a model space as particles in Particle Swarm Optimization (PSO) fly around in a solution space. In concrete, agents utilize three pieces of information such as social influence, personal influence and inertia. We attempt to examine the performance of SPBIL for discrete optimization problems; Multi Knapsack Problems (MKPs) is used as a set of benchmark test problems. Results show that SPBIL exhibits better results than other algorithms in all problems. The reason SPBIL has higher performance is the ability of diversity maintenance in early search iterations judging from a diversity index proposed by R. K. Ursem.

キーワード：進化的計算、遺伝的アルゴリズム、分布推定アルゴリズム、並列アルゴリズム、粒子群最適化、Population Based Incremental Learning (PBIL)、離散最適化問題

**Keywords:** Evolutionary computation, Genetic Algorithm, Estimation of Distribution Algorithm, Parallel algorithm, Particle swarm optimization, Population Based Incremental Learning (PBIL), Discrete Optimization Problem

## 1. はじめに

進化計算は、生物学的進化のプロセス（自然淘汰など）を応用した人工モデルによる最

適化問題などの近似解を探索するメタヒューリスティクスである。その研究は、1960年代の Holland<sup>(1)</sup>、Fogel et al.<sup>(2)</sup>、Rechenberg<sup>(3)</sup> の研究に端を発し、現在でも盛んに研究が進められている。特に、非線形最適化問題、組合せ最適化問題などへの応用分野では顕著な成果をあげている。ここ数十年、この進化計算の研究分野において、遺伝オペレータ（交叉・突然変異など）を使わず、優良個体集団の遺伝子構造の確率分布の推定を行い、その分布からサンプリングしたものを次世代の個体（解候補）とする確率モデル遺伝的アルゴリズム（Probabilistic Model-Building Genetic Algorithm：PMBGA）や分布推定アルゴリズム（Estimation of Distribution Algorithm：EDA）が注目を浴びている。その中でも特に単純なモデルである Population Based Incremental Learning（PBIL）<sup>(4)</sup> は、変数間の従属関係を独立として仮定するなどの問題点はあるが、計算コストから見ても十分な性能を有していることは Baluja<sup>(4)</sup> の研究からも明らかである。

進化計算手法の1つである遺伝的アルゴリズム（Genetic Algorithm：GA）は並列化に適したアルゴリズムであり、並列化することにより最適解の探索性能が向上することが数多く報告されている<sup>(5), (6), (7)</sup>、GAの並列化手法としてよく知られた方法に島モデルがある。通常のGAが一つの集団で探索を行うのに対し、島モデルGAでは母集団を複数のサブ集団（島）に分割して探索を行う。局所解が多い問題に対しては多くのサブ集団（島）を使った探索が有効であることが知られており、Holtschulte<sup>(8)</sup> や Whitley et al.<sup>(9)</sup> らの論文では一つのサブ集団（島）あたりの個体数の規模についての詳細な分析がなされている。さらに、Skolicki and De Jong<sup>(10)(11)</sup> らは各サブ集団（島）に異なる遺伝子表現型を用いて進化の多様性を図った手法や、サブ集団間（島間）を移住する個体数や移住の仕方について有効な提案を行っている。GAの一種であるPBILも Miyagi et al.<sup>(12)</sup> や Rodrigues<sup>(13)</sup> では島モデルによるPBILの並列化により探索性能の向上を行っている。Miyagi et al.<sup>(12)</sup> ではサブ集団間（島間）の移住にツリートポロジーを利用し、ネットワーク構造による移住制限を設けている。一方、Rodrigues<sup>(13)</sup> では各サブ集団間（島間）で直接移住を行うのではなく、ブリッジとなる島を介して移住を行うマスタースレーブ方式をとっている。

本研究では、PBILの並列化を PSO（Particle Swarm Optimization）<sup>(14)</sup> の考え方に基づいたマルチエージェント型の進化計算モデルとして捉え直した新たなアルゴリズムを提案する。PSOは粒子の群れに解空間上を飛行させて最適な解を探す手法である。粒子はランダムに解空間を飛行するのではなく、優れた解の情報に基づいて飛ぶ方向を決める。一方、提案アルゴリズムでは探索エージェントどうしで優良解の共有情報をもとにして、モデル空間を飛行（移動）することで自身が保持する確率ベクトルの更新を行い、その確率ベクトルに従い、探索エージェントは、それぞれの解空間上で複数の解候補を生成し評価する。これらを繰り返すことで優良遺伝子構造の確率分布（確率ベクトル）を推定するのが本研究の提案アルゴリズムの特徴である。本研究のアイデアに近い研究として、PBILの並列化ではないが A. Moraglio et al.<sup>(15)</sup> らによる幾何学的なフレームワークを用いた PSO の研究がある。

本論文の構成は以下のとおりである。第2節では本研究で新たに提案する進化計算アルゴリズムと関係性が深い並列PBILとPSOの2つのアルゴリズムについて解説する。第3節では提案アルゴリズムについて述べる。第4節では、離散最適化問題（Discrete Optimization Problem）を用いた性能比較実験の結果を示す。最後に第5節で結論を述べ本論文のむすびとする。

## 2. 関連アルゴリズムの概説

### 2. 1 並列 PBIL

1994 年に Baluja<sup>(4)</sup> により開発された PBIL は、交叉・突然変異のかわりに確率ベクトルを用いることから、EDA または PMBGA に分類される。標準的な PBIL の並列化においては、生物が島ごとに独自の進化を遂げていることを模倣して母集団を複数のサブ集団（島）に分割し、それぞれが独自の探索（進化）を行うことで探索（進化）の多様化を図っている。

以下では標準的な並列 PBIL アルゴリズムの流れを簡単に述べる。

Step1：確率ベクトルの初期化

各サブ集団  $i$  の確率ベクトル  $P^i=(p_1^i, p_2^i, \dots, p_{nbit}^i)$  をすべて 0.5 に初期化する。ここで、確率ベクトルの成分数は  $nbit$ （個体の遺伝子長）とする。ただし、各個体の遺伝子長は 1 変数あたり何ビットで表現するかによって決まる。ここで目的関数の次元数（変数の数）を  $D$  とすると遺伝子長は、 $nbit = (D \times (1 \text{ 変数あたりのビット数}))$  となる。

Step2：確率ベクトルに従い次世代の個体群を生成

確率ベクトルの各成分は、各成分が 1 を発生する確率を表しているので、例えば、それに従って個体を発生させると、 $nbit$  ビットの (10100...1) のような個体を得ることができる。このようにして複数の個体を発生させる。

Step3：各個体の評価

個体ごとに対象の最適化問題の目的関数の値を計算して、その値を適合度とする。（適合度はスケール処理される場合もある）

Step4：優良個体の選択

世代  $t$  におけるサブ集団  $i$  の個体群の中で最良の適合度を持つ優良個体を  $B^i(t)=(b_1^i(t), b_2^i(t), \dots, b_{nbit}^i(t))$  として保持する。そして、世代  $t$  の全サブ集団の中で最良の適合度を持つ優良個体を  $B^G(t)=(b_1^G(t), b_2^G(t), \dots, b_{nbit}^G(t))$  として保持する。

Step5：確率ベクトルの更新

選択された優良個体に基づいてそれぞれのサブ集団の確率ベクトルを式(1)で更新する。

$$p_j^i(t+1)=(1-LR)p_j^i(t)+LR(c_1b_j^i(t)+c_2b_j^G(t)) \quad \text{式(1)}$$

$LR \in [0,1]$  は学習率 (Learning Rate) を表す。 $c_1$  と  $c_2$  はそれぞれ  $B^i(t)$ 、 $B^G(t)$  に対する重み係数である。

Step6：突然変異

突然変異率  $MR$  で確率ベクトル  $P$  の各成分の値を式(2)で変更する。

$$p_j^i(t-1) \leftarrow (1-MR)p_j^i(t-1)+MR \cdot rand \quad \text{式(2)}$$

$MR \in [0,1]$  (Mutation Rate) は突然変異による変化の度合いを表す。 $rand \in [0,1]$  は一様乱数とする。

Step7：Step2 ～ Step6 の繰り返し

Step2 ～ Step6 の処理を終了条件が満たされるまで繰り返す。

## 2. 2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) は1995年に Eberhart and Kennedy<sup>(14)</sup> によって提案された鳥や魚の群れなどの行動を模倣した最適化アルゴリズムである。群れ (swarm) は多次元空間において位置  $s$  と速度  $v$  を持つ粒子 (particle) 群としてモデル化される。各粒子は個としての最良解と群れ全体の最良解の2つを利用して探索点を更新しながら、最適な位置 (最適解) を見つけ出す。

以下で PSO アルゴリズムの流れを簡単に述べる。

Step1: 各粒子の位置と速度を初期化

粒子  $i$  の速度と位置をそれぞれ  $v^i$ , 位置  $s^i$  とする。

Step2: 各粒子  $i$  の現在位置から適合度を計算

適合度関数に現在の位置  $s^i$  を代入して適合度を計算する。

Step3: 各粒子  $i$  の最良解と群れ全体の最良解の更新

各粒子  $i$  が現在保持している世代  $t$  の最良解  $Pb^i=(pb_1^i, pb_2^i, \dots, pb_D^i)$  と世代  $t+1$  で新たに発見した優良解を比較して、世代  $t$  の最良解よりも優良な個体であれば、その個体を世代  $t+1$  の最良解  $Pb^i=(pb_1^i, pb_2^i, \dots, pb_D^i)$  として更新する。同様の方法で、群れの全体の最良解  $Gb=(gb_1, gb_2, \dots, gb_D)$  も更新する。

Step4: 各粒子  $i$  の速度と位置を更新

世代  $t$  における各粒子  $i$  の過去の最良解  $Pb^i(t)$  と全体の過去の最良解  $Gb(t)$  を用いて粒子  $i$  の速度  $v^i$  と位置  $s^i$  を式(3)から式(5)を使い更新する。

$$w(t)=w_{\max}-\frac{w_{\max}-w_{\min}}{iter_{\max}} \times t \quad \text{式(3)}$$

$$s_j^i(t+1)=s_j^i(t)+v_j^i(t+1) \quad \text{式(4)}$$

$$v_j^i(t+1)=w(t)v_j^i(t)+c_1r_1(pb_j^i(t)-s_j^i(t))+c_2r_2(gb_j(t)-s_j^i(t)) \quad \text{式(5)}$$

ただし、 $w$  は慣性係数、 $c_1$  と  $c_2$  はそれぞれ  $Pb^i(t)$ 、 $Gb(t)$  に対する加速係数、 $r_1$  と  $r_2$  は  $[0,1]$  の範囲の値をとる一様乱数とする。

Step5: Step2 ~ Step4 の処理を終了条件が満たされるまで繰り返す。

以上でアルゴリズムの流れを簡単に述べたが、以下では、Step4 の式について少し詳しく説明する。式(5)の左辺第一項は一世代前の速度による慣性 (*inertia*) の影響を表している。また、式(5)の左辺第二項は各粒子の最良解からの影響 (*personal influence*)、さらに、第三項は全粒子の最良解 (*social influence*) を表している。後者の2項は  $c_1$  と  $c_2$ ,  $r_1$  と  $r_2$  により重みづけられ、探索の集中と多様化のバランスに影響を与えている。

次に、アルゴリズムの動作をイメージできるように、探索次元が3のケースでの粒子の位置の更新の様子を Fig.1 に示す。まず、世代  $t$  における粒子  $i$  の位置  $S^i(t)$  を始点として  $Pb^i(t)$  を通る半直線と、同じく  $S^i(t)$  を始点として  $Gb(t)$  を通る半直線を引く。 $c_1$  と  $c_2$  によって決まるそれぞれの半直線上の点と  $S^i(t)$  を結ぶ2つの線分、そしてそれらの対辺から成る平行四辺形<sup>i</sup>を描くことができる。乱数  $r_1$  と  $r_2$  によって、この平行四辺形の内点として  $A^i(t)$  が決定される。この  $A^i(t)$  と粒子の速度  $V^i(t)$  に慣性パラメータを掛けた  $wV^i(t)$  の和として、 $t+1$  世代での速度  $V^i(t+1)$  が求まる。 $S^i(t)$  が  $V^i(t+1)$  で移動した位置が  $t+1$  世代での粒子  $i$  の位置  $S^i(t+1)$  となる。

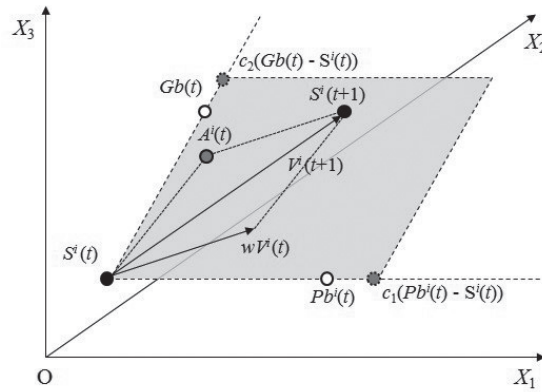


Fig.1 : Update the velocity and position of a particle.

### 3. A Society of Population Based Incremental Learners (SPBIL)

本節では、PBIL の並列化を PSO の考え方に基づいたマルチエージェント型の進化計算モデルとして捉えなおした新たなアルゴリズムを提案する。本提案アルゴリズムの探索エージェントの基本動作は次に示す 2 つである。

- ①各エージェントは PBIL のアルゴリズムを用いて、各確率ベクトルに従い解候補を複数発生させる。さらに全ての解候補の評価をして暫定的な最良解を記憶して保持する。
- ②各エージェントは自身の保持している最良解と全エージェント内で共有している最良解をもとに一辺の長さが 1 のハイパーキューブのモデル空間上を移動することでそれぞれの確率ベクトルを更新する。(モデル空間上のエージェントの位置は確率ベクトルが示す座標である)

これら 2 つの基本動作の繰り返しにより、エージェントは優良解を生成する確率分布（確率ベクトル）を推定することで最適解を見つけ出すところに本提案アルゴリズムの特徴がある。以下では各エージェントが確率ベクトルを更新するイメージを捉え易くするために、探索次元が 3 の場合の動作を図示した Fig.3 を参照しながら、各エージェントの確率ベクトルの更新方法を詳しく説明する。

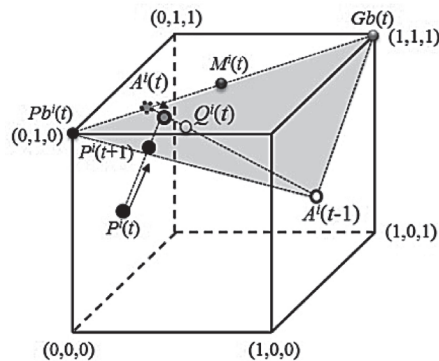


Fig. 2 : Conceptual diagram of the algorithm

i この平行四辺形の辺の長さは、 $c_1$  と  $c_2$  によっては、Fig.1 と異なり辺  $S^i(t)Pb^i(t)$  や辺  $S^i(t)Gb(t)$  よりも短くなることがある。



・第1世代 ( $t=1$ ) :

エージェント  $i$  は、自身の保持している確率ベクトル  $P^i=(0.5,0.5,...,0.5)$  を用いて、解候補を複数発生させて解候補群を得る。さらに各エージェントはそれぞれが生成した解候補群のすべての解候補を評価する。エージェント  $i$  は、自身が過去に探索した最良解と全エージェント内における過去の最良解を、それぞれ、 $Pb^i(t)=(pb_1^i(t),pb_2^i(t),...,pb_{nbit}^i(t))$ 、 $Gb(t)=(gb_1(t),gb_2(t),...,gb_{nbit}(t))$  として記憶し保持する。エージェント  $i$  の最良解  $Pb^i(t)$  からの影響が PSO における個々の情報 (*personal influence*)、全エージェント内での最良解  $Gb(t)$  からの影響が共通知報 (*social influence*) からの影響に相当する。

ここで Fig.3 の点  $Pb^i(t)$  と点  $Gb(t)$  を結ぶ線分の中点を  $M^i(t)=(m_1^i(t),m_2^i(t),...,m_{nbit}^i(t))$  と表記すると、その中点の座標の各成分は式(6)で求められる。

$$m_j^i(t)=\frac{pb_j^i(t)+Gb_j(t)}{2} \quad \text{式(6)}$$

次に線分  $Pb^i(t)M^i(t)$  をランダムに  $r_2 : r_1$  に内分した点を  $A^i(t)=(a_1^i(t),a_2^i(t),...,a_{nbit}^i(t))$  と表記すると、その内分点の座標の各成分は式(7)で求められる。

$$a_j^i(t)=\frac{r_1 pb_j^i(t)+r_2 m_j^i(t)}{r_1+r_2} \quad \text{式(7)}$$

この  $A^i(t)$  を教師ベクトルとし、通常の PBIL と同様の学習を式(8)で行い、 $P^i$  を更新する。(エージェント  $i$  は確率ベクトル  $P^i(t)$  の示す座標から確率ベクトル  $P^i(t+1)$  の示す座標に移動する)

$$P_j^i(t+1)=(1.0-LR)p_j^i(t)+LR \cdot a_j^i(t) \quad \text{式(8)}$$

・第2世代以降 ( $t \geq 2$ ) :

エージェント  $i$  は、更新された確率ベクトルを用いて複数の解候補を発生させる。すべての解候補を評価して、 $Pb^i(t)$  と  $Gb(t)$  を更新する。ここで新たな点  $A^i(t)$  を線分  $Pb^i(t)M^i(t)$  上に第1世代と同様にランダムに決める。新たに決めた  $A^i(t)$  と1世代前の  $A^i(t-1)$  により作られた線分を 1:3 に内分<sup>ii</sup>する点を  $Q^i(t)$  とすると、その座標の成分は式(9)で表すことができる。

$$q_j^i(t)=\frac{3a_j^i(t)+a_j^i(t-1)}{4} \quad \text{式(9)}$$

さらに点  $A^i(t)$  と点  $Q^i(t)$  の間をランダムに  $r_2 : r_1$  に内分した点を最新の  $A^i(t)$  として更新する。その内分点の座標の各成分は式(10)で求められる。

$$a_j^i(t) \leftarrow \frac{r_1 a_j^i(t)+r_2 q_j^i(t)}{r_1+r_2} \quad \text{式(10)}$$

ii 内分の割合は経験的に設定した。

この最新の  $A^i(t)$  を用いて式(8)で確率ベクトルを更新し、エージェント  $i$  はモデル空間上の点  $P^i(t+1)$  に移動する。このように第2世代以降は、PSOと同様に慣性 (*inertia*)<sup>iii</sup>、個々の情報 (*personal influence*)、共有情報 (*social influence*) の3つの影響を受けることになる。

#### 4. 実験の概要と結果

本節では、離散最適化問題による性能比較実験の概要と結果について述べる。以下では、まず離散最適化問題として採用したマルチナップザック問題について説明して、つぎに実験に用いたパラメータの設定の詳細と実験の結果について順に述べる。

##### 4. 1 マルチナップザック問題

マルチナップザック問題は離散最適化問題の1つである。例えば、プロジェクト候補が  $D$  個あり、各プロジェクトは  $E$  種類の資源からいくつかの資源を消費して実行されるものとする。ただし、各資源の使用できる上限の量はあらかじめ決まっているものとする。また各プロジェクトにより得られる利益と消費する資源量は異なるものとする。このような制約の下で、総利益が最大となるプロジェクトの組合せを選択することがマルチナップザック問題の目的である。以下で、マルチナップザック問題を定式化する。

今、プロジェクト  $i$  によって得られる利益を  $u_i$ 、消費する資源  $j$  の量を  $e_i^j$  とする。また、消費できる各資源の総量には制約  $C^j$  がものとする。プロジェクト  $i$  の採択状態を状態変数  $x_i$  で表わし、プロジェクトが採択されれば場合に  $x_i=1$ 、非採択ならば  $x_i=0$  をとるものとする。採択されたプロジェクトから得られる利益の総和を  $F$  とすると、式(11)として定式化できる。

$$\begin{aligned} \max F &= \sum_{i=1}^D u_i x_i \\ \text{subject to } \sum_{i=1}^D e_i^j x_i &\leq C^j (\text{for all } j) \end{aligned} \quad \text{式(11)}$$

マルチナップザック問題を提案アルゴリズムによって解く場合、制約条件を満たさない解が生成されることがある。この場合、制約条件を満たさない解の評価値は0として処理することが一般的であるが、そうすることで有益な情報を切り捨てる可能性がある。そこで、本研究では制約条件を満たさない解にペナルティを課す方法を採用することにする。下式(12)のように  $y^j$  を制約  $C^j$  を満たさない場合のペナルティとする。

$$y^j = \begin{cases} \frac{u_{\text{avg}}}{e_{\text{avg}}^j} \left( \sum_{i=1}^D e_i^j x_i - C^j \right) + u_{\text{avg}} & \text{if } \sum_{i=1}^D e_i^j x_i > C^j \\ 0 & \text{otherwise} \end{cases} \quad \text{式(12)}$$

ここで  $u_{\text{avg}}$  はプロジェクトの利益の平均、 $e_{\text{avg}}^j$  はプロジェクトの資源  $j$  の消費量の平均を表す。解の評価値  $f$  は、

$$f = F - \sum_{j=1}^E y^j \quad \text{式(13)}$$

で与えられる。

iii  $A^i(t-1)$  によって慣性 (*inertia*) の影響が生じる。

## 4. 2 実験の設定

本実験に用いたベンチマークテストは、Skorobohatyj<sup>(16)</sup> のマルチナップザック問題である。実験の設定は、試行回数 50 回、解生成数 500000、テスト関数の次元数（決定変数の数） $D = 30$  とした。また、パラメータは一集団あたりの個体数  $n = 100$ 、集団数（エージェント数） $I = 5$ 、遺伝子長  $nbit = 240$ 、 $LR = 0.05$ 、 $MP = 0.02$ 、 $MR = 0.05$ <sup>iv</sup> とした。一集団あたりの個体数は標準の PBIL と提案アルゴリズムともに  $n = 100$  とし、終了世代数をエージェントやサブ集団の数に応じて変えることで最終的な解の生成数を同数とした。また、各世代の多様性は Ursem<sup>(17)</sup> に基づいて式(14)で計算した。

$$Diversity = \frac{1}{|L| \cdot (n \cdot I)} \cdot \sum_{i=1}^{n \cdot I} \sqrt{\sum_{j=1}^D (x_j^i - \bar{x}_j)^2} \quad \text{式(14)}$$

ただし、 $|L|$  は探索空間の対角線の長さであり、各  $x_j$  の定義域によって求められる。 $x_j^i$  は個体  $i$  の要素  $j$  の値、 $\bar{x}_j$  は要素  $j$  の平均値を意味する。

## 4. 3 実験結果

実験結果を Table 1 に示す。表の値は 50 回の試行で得られた最適解の平均値である。すべてのマルチナップザック問題のベンチマークテストにおいて提案アルゴリズム (SPBIL) が他のアルゴリズムよりも優れた結果であった。また、マルチナップザック問題においては集団数（エージェント数）を多くした方が高い探索性能を有することも示された。Fig.3(a) から (e) の最適解と多様性の推移を示すグラフからもわかるようにすべての問題で提案アルゴリズムは標準的な並列 PBIL よりも大きな多様性を持つ傾向があることがわかった。さらに Fig.4 の Boxplot 図<sup>v</sup> から、提案アルゴリズムは標準的な並列 PBIL より外れ値が少ないことからアルゴリズムの安定性に優れていることが示された。

iv この設定は先行研究<sup>(4)</sup>などで推奨されているパラメータの組である。

v 本稿の Boxplot 図はボックスの上辺が第3四分位点、下辺が第1四分位点、ボックス内部の太線が第2四分位点（中央値）を表す。 $\pm 2\sigma$ の範囲に含まれないデータは外れ値として白い円で描かれる。ボックスから延びる垂線の先端はそれぞれ外れ値を除外したデータの内での最大値と最小値である。



Table 1 : Performance Comparison with MKPs

Algorithm	Number of Sub populations (Agents)	Multiple Knapsack Problem				
		Fleisher	Petersen 7	Senju, Toyoda 1	Weingartner 7	Weishih 30
PBIL		2108.100	16503.860	7752.580	1095141.980	11186.920
Parallel PBIL	5	2127.620	16534.580	7768.760	1095356.940	11190.520
PBIL	10	2132.240	16534.220	7769.040	1095390.780	11190.920
SPBIL	5	2126.880	<b>16537.000</b>	7770.120	1095387.060	<b>11191.000</b>
SPBIL	10	2135.020	<b>16537.000</b>	<b>7771.340</b>	<b>1095402.060</b>	<b>11191.000</b>

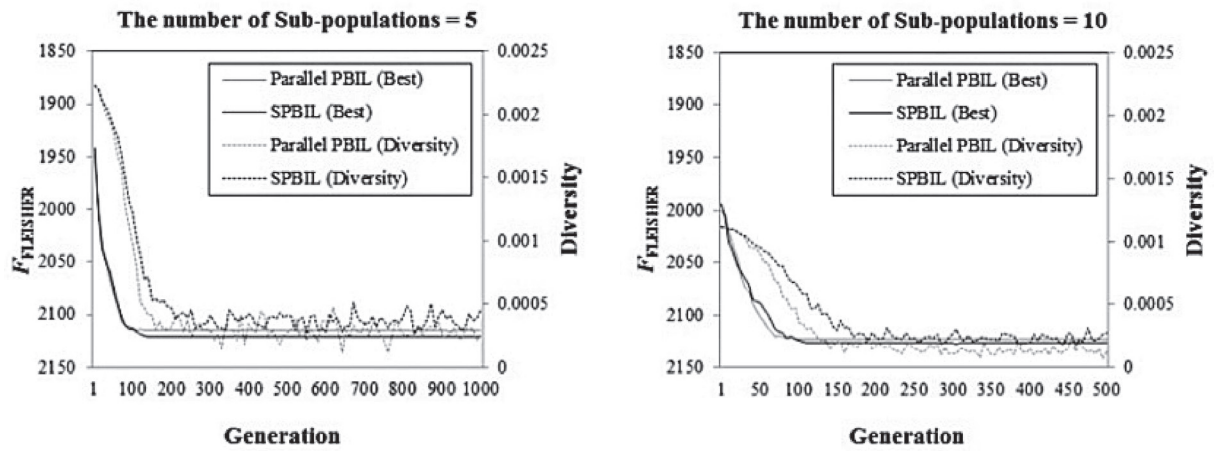


Fig. 3(a) : Transition of the best solution and diversity for MKP (FLEISHER).

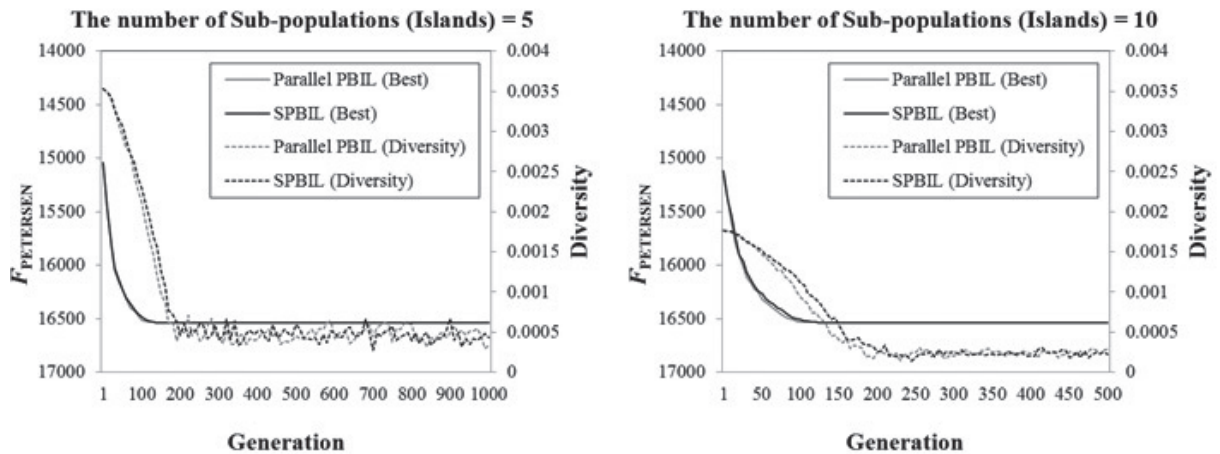


Fig. 3(b) : Transition of the best solution and diversity for MKP (PETERSEN 7).

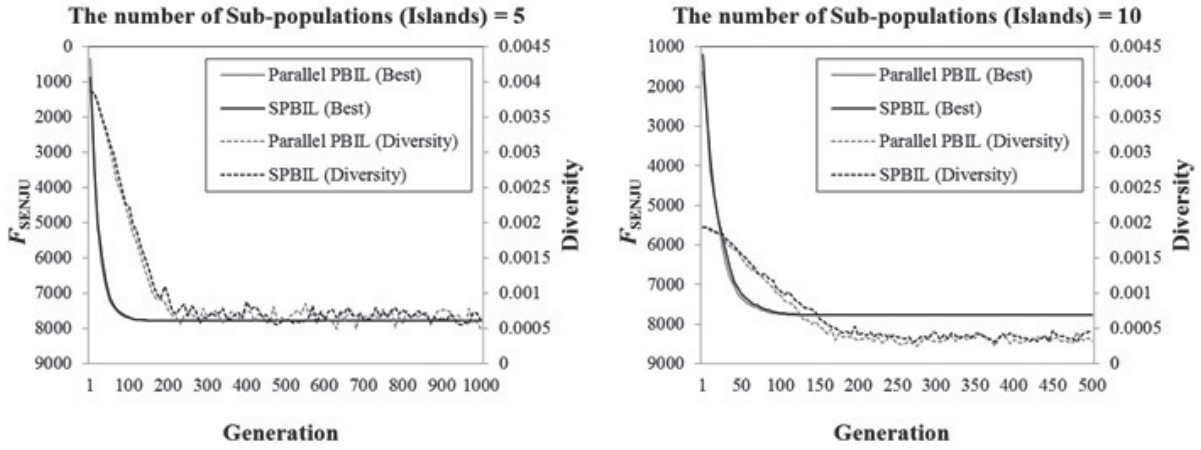


Fig. 3(c) : Transition of the best solution and diversity for MKP (SENJU, TOYODA 1).

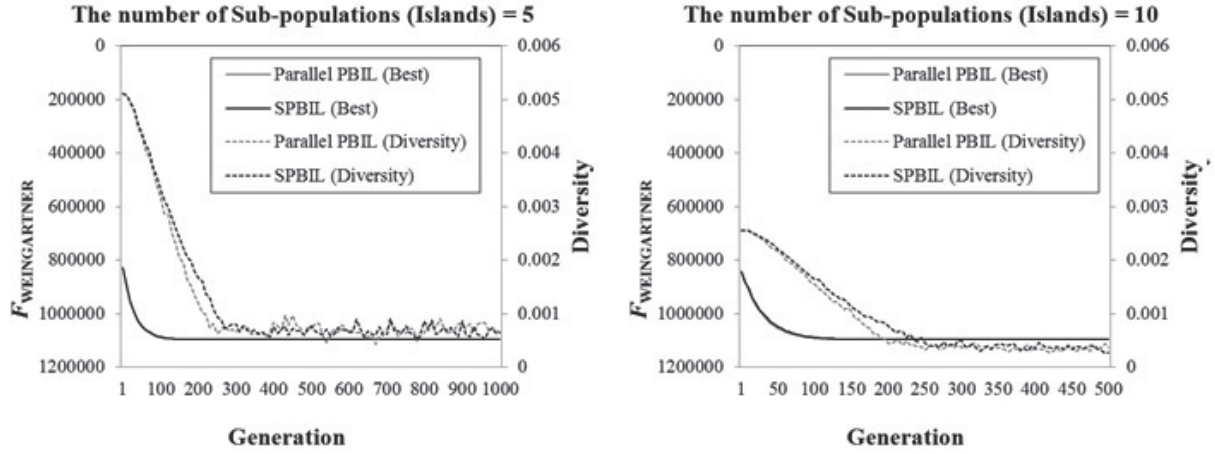


Fig. 3(d) : Transition of the best solution and diversity for MKP (WEINGARTNER 7).

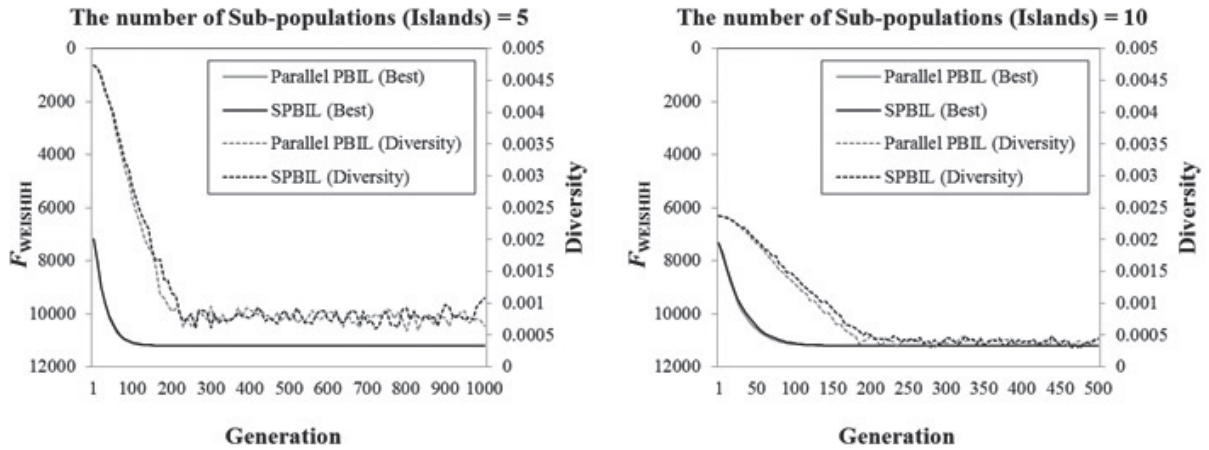


Fig. 3(e) : Transition of the best solution and diversity for MKP (WEISHIH 30).

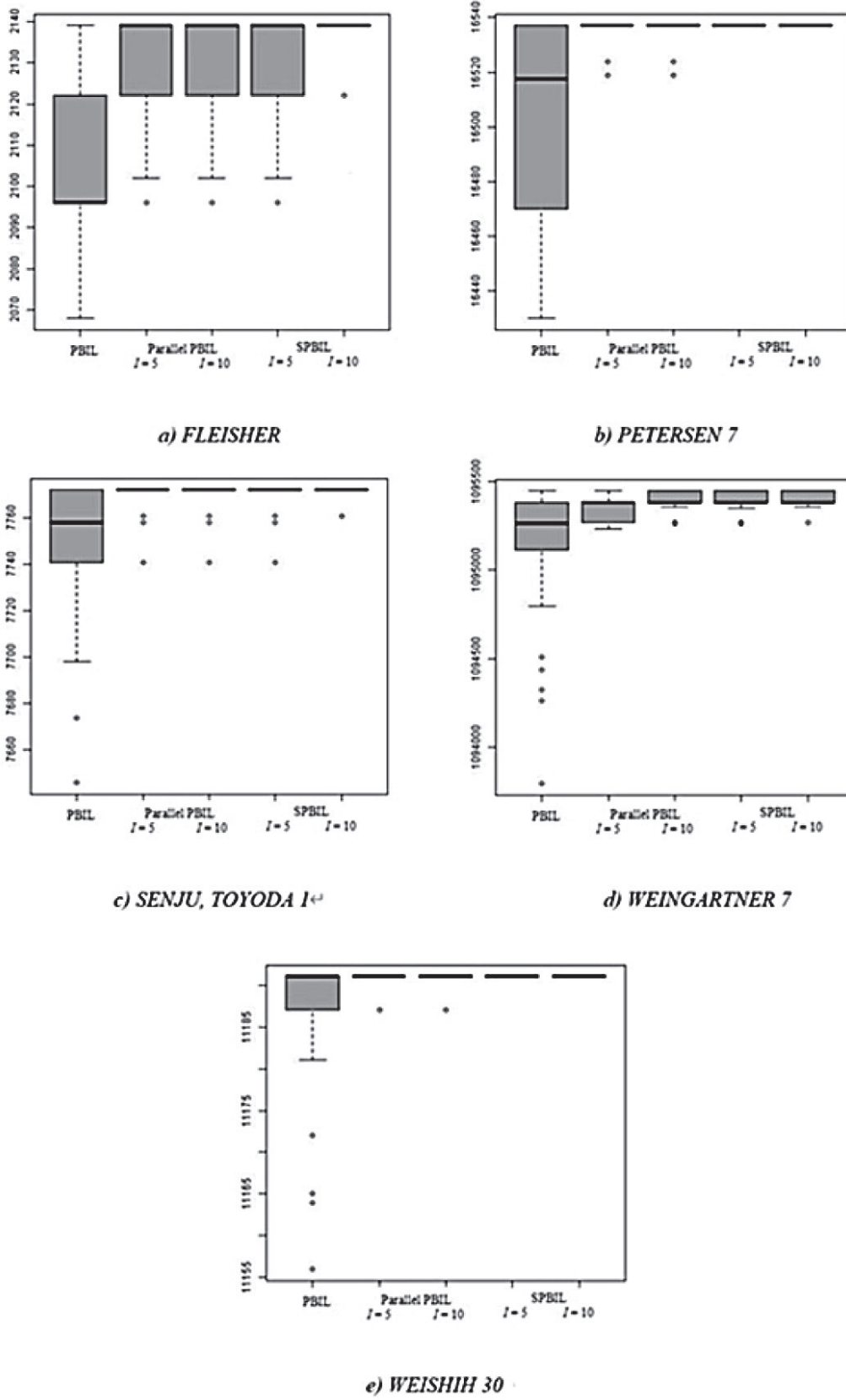


Fig.4 : Boxplot for the best-found solutions.

## 5. おわりに

本研究では複数の探索エージェントが PSO における粒子の動作を模倣して、解空間ではなくモデル空間を移動することで最適化を行う新たな進化計算アルゴリズムを提案した。さらに、提案アルゴリズムの有効性を検証するための数値実験も行った。その実験の結果、提案アルゴリズムは離散最適化問題に対し、頑健性を備えた優れた性能を有することが確認された。この理由の1つとして、提案アルゴリズムは多様性の維持により最適解探索における初期収束を防いでいると考えられる。

今後の課題としては、最適なエージェント数について明らかにすることと、モデル空間を情報幾何学的に精緻化することで理論研究を進めることが挙げられる。

## 参考文献

- (1) J. H. Holland: *Adaptation in natural and artificial systems*: An introductory analysis with applications to biology, control, and artificial intelligence, U Michigan Press (1975).
- (2) L. J. Fogel, A. J. Owens, and M. J. Walsh: *Artificial Intelligence through Simulated Evolution*, John Wiley (1966).
- (3) I. Rechenberg: *Evolutionsstrategien*, Springer Berlin Heidelberg (1978).
- (4) S. Baluja: "Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning," *Technical Report*, CMU-CS-94-163, (1994).
- (5) E. Alba, and J. M. Troya: "A Survey of Parallel Distributed Genetic Algorithms," *Journal Complexity*, Vol. 4, Issue 4, pp. 31-52, (1999).
- (6) E. Cantú-Paz: "A survey of parallel genetic algorithms," *Calculateurs paralleles, reseaux et systems repartis*, Vol. 10, No. 2, pp141- 171, (1998).
- (7) L. Wang, A. A. Maciejewski, H. J. Siegel, V. P. Roychowdhury, and B. D. Eldridge: "A Study of Five Parallel Approaches to a Genetic Algorithm for The Traveling Salesman Problem," *Intelligent Automation and Soft Computing*, Vol. 11, No. 4, pp. 217-234, (2005).
- (8) N. Holtschulte: "Optimal Population Size in Island Model Genetic Algorithms," *Proc. 8th Student Conference*, p. 94, (2012).
- (9) D. Whitley, S. Rana, and R. B. Heckendorn: "The island model genetic algorithm: On separability, population size and convergence," *Journal of Computing and Information Technology*, Vol. 7, pp.33-48, (1999).
- (10) Z. Skolicki and K. De Jong: "Improving Evolutionary Algorithms with Multi-representation Island Models," *Parallel Problem Solving from Nature-PPSN VIII*. Springer Berlin Heidelberg, pp. 420-429, (2004).
- (11) Z. Skolicki and K. De Jong: "The influence of migration sizes and intervals on island models," *Proc. the 2005 Conf. on Genetic and evolutionary computation*, GECCO' 05, pp. 1295-1302, (2005).
- (12) H. Miyagi, T. Tengan, S. Mohamed, and M. Nakamura: "Migration Effects on Tree Topology of Parallel Evolutionary Computation," *Proc. TENCON 2010 - 2010 IEEE Region 10 Conference*, pp. 1601 – 1606, (2010).
- (13) R. Rodrigues: "Distributed Population Based Incremental Learning," *Proc. Sixth WSEAS Int. Conf. on Evolutionary Computing*, Lisbon, Portugal, pp.111-116, (2005).
- (14) R. Eberhart, and J. Kennedy: "A new optimizer using particle swarm theory," *Proc. Sixth International*

*Symposium on Micro Machines and Human Science*, Nagoya, Japan, (1995).

- (15) A. Moraglio, C. Di Chio, J. Togelius, and R. Poli: “Geometric Particle Swarm Optimization,” Hindawi Publishing Corp. *Journal of Artificial Evolution and Applications*, Vol. 2008, Article ID 143624, 14 pages doi:10.1155/2008/143624
- (16) G. Skorobohatyj: “MP-Testdata AC-94 Suite of 0/1-Multiple-Knapsack Problems,” in ZUSE-INSTITUT BERLIN (ZIB), (2004). <<http://elib.zib.de/pub/Packages/mp-testdata/ip/sac94-suite/>>, accessed January 29, (2013).
- (17) R. K. Ursem: “Diversity-guided evolutionary algorithms,” *Parallel Problem Solving from Nature—PPSN VII*, pp.462-471, (2002).