

粒子群最適化法のネットワーク化による 探索性能向上に関する研究

平成28年度

東京都市大学大学院工学研究科
情報工学専攻

佐々木 智志

目次

1	緒論	2
2	本論文に関わる基本事項	9
2.1	最適化問題	9
2.2	メタヒューリスティクス	14
3	本論文に関わる諸研究	21
3.1	粒子群最適化法	22
3.2	PSO の従来研究	24
3.2.1	解探索性能の向上に関する研究	24
3.2.2	理論解析に関する研究	25
3.2.3	並列計算環境への実装に関する研究	27
3.2.4	適用可能な問題の拡張に関する研究	28
3.2.5	ハードウェア化に関する研究	28
3.3	粒子間に近傍構造を有する PSO の研究	29
3.3.1	静的な近傍構造の研究	30
3.3.2	動的な近傍構造の研究	34
3.4	マルチスウォーム PSO の研究	38
3.4.1	単一の近似解を探索する研究	39
3.4.2	複数の近似解を探索する研究	40
3.5	PSO アルゴリズムの並列計算環境への実装の研究	41
3.5.1	シングルスウォーム PSO の実装に関する研究	42
3.5.2	マルチスウォーム PSO の実装に関する研究	42
4	粒子群最適化法のネットワーク化による解探索性能の向上手法	54
4.1	4 章の概要	54
4.2	粒子群最適化ネットワーク	54
4.3	PSON における粒子の理論解析	59

4.3.1	粒子の収束性の解析	59
4.3.2	安定性解析における数値実験	61
4.4	数値実験	64
4.4.1	<i>DBG</i> の設定に関する検討	64
4.4.2	解探索性能の比較実験	69
4.5	4 章の結び	75
5	多様性向上のための粒子群最適化ネットワークに対する確率的なネットワーク化手法	78
5.1	5 章の概要	78
5.2	確率的結合を有する粒子群最適化ネットワーク	80
5.3	PSON-SC と PSON の多様性についての検討	84
5.4	数値実験	87
5.5	計算時間における比較検討	98
5.6	5 章の結び	99
6	結論	102

第一章 緒論

1 緒論

工学の目的は人間社会に有用な機器、構造物あるいは各種の設備やそれを運用するシステムなどを創生することであり、そのための作業は一般に工学設計と呼ばれている。工学設計では、設計過程において考慮すべき点が多数存在し、それらを満たす候補案も多数存在する。このような状況に対して科学的な最適化手段を用いて設計を行うことが求められている。また、工学設計以外の場において、例えば、生産活動、販売活動、投資決定などのオペレーション (Operation) に対しても最適な意思決定を行うことが求められている。これらの状況においてはその選択を経験や勘のみに基づいて行うのではなく、科学的に行うことが望ましい。

オペレーションズ・リサーチ (Operations research; OR) は“最適な意思決定”を行うために生まれた学問であり、その黎明期には軍事上の問題解決に用いられていたが、今では、在庫管理、生産計画、輸送問題など様々な分野において用いられている。OR では、解決すべき問題を観察し、その問題を数理計画 (Mathematical programming) することでモデルを作成する。そして、作成したモデルの最良解を導き出し、その解が解決案となる。

数理計画は対象とする問題に対する最適な解決策を求めるための定式化の手法である。数理計画によって解決すべき問題を最適化問題 (Optimization problem) として記述し、数学的にその数式を解くことで最適な状態 (Optimal solution または Globally optimal solution; 最適解) を求めることができる。最適化問題の定式化 (モデル化) は目的関数や制約条件の種類によっていくつかの問題のクラスに分けることができる。線形計画問題、非線形計画問題 (連続最適化問題)、組合せ最適化問題 (離散最適化問題) にクラス分けできる。

このような数理計画された最適化問題を解くための手法として、モデルの勾配情報を用いて解を求める手法も様々に提案されている。最急降下法 (Steepest decent method)、ニュートン法 (Newton's method) などは勾配法 (Gradient method) の一種である。

しかし、世の中の多くの問題においては数理計画が困難なことがある。このような問題では目的関数の勾配情報や景観が単峰性か多峰性か変数分離可能か不可能かなどの情報が事前に明示されず、設計変数 (Design variables) \mathbf{x} に対する目的関数値 $f(\mathbf{x})$ しか知ることができない問題を“Black-box optimization” (ブラックボックス最適化問題) と呼び、現実の多くの問題がこのクラスに該当する。このような問題に対しては勾配法の使用が困難である。実世界のシステム最適化問題においてはブラックボックス最適化問題であり、システム内部は未知であり目的関数を定式化もしくはモデル化できない場合がある。この場合でも、システムを評価する評価量が存在することから、この評価量を目的関数値とすることで最適化手法を適用することができる。そのため、最適性の保証はなくても目的関数値がどれくらい良い解であるかを評価し、実行可能解の探索を行うことを第一義のヒューリスティクス解法 (Heuristics; 発見的手法) が利用される。

ヒューリスティクスとは、「許容性や最適性の保証はないが、実用的な計算時間内に良好な近似解

あるいは最適解を発見する手法」のことである。ただし、ヒューリスティクスは特定の問題にしか適用できないことが多い。

ヒューリスティクスによって近似解を効率的に求めることができるが、その近似解に対して修正を加えることで、さらに最適解へ近づけようとする戦略をメタヒューリスティクス (Metaheuristics) と呼ぶ。よって、メタヒューリスティクスはヒューリスティクスよりも広いクラスの問題に適用可能な手法である。メタヒューリスティクスにおいて共通する基本的な操作として (1) 過去の探索履歴を利用し新しい解を生成する, (2) 生成した解を評価し次の解探索に必要な情報をフィードバックする, があり, これらの操作の反復により解空間の探索を行う。

多くのメタヒューリスティクスは「近傍」の概念を利用している。「近傍」とは, 「ある解を与えたとき, その解に小さな摂動を加えて得られる解の集合」であり, 「良い解同士は似通った構造を持つ」という POP (Proximate optimality principle) の概念に基づいている。多くのメタヒューリスティクスは「近傍」を基礎として構築されている。

メタヒューリスティクスは様々な手法が提案されており, 遺伝的アルゴリズム (Genetic algorithm; GA), 粒子群最適化法 (Particle swarm optimization; PSO), 差分進化法 (Differential evolution; DE) などが良く知られた手法である。

本論文では, 1995 年に J. Kennedy と R. C. Eberhart らによって提案された PSO に注目した。PSO は解候補である探索個体 (Particle; 粒子) を解空間に多数配置し, 各粒子は他粒子との相互作用により解空間を並列的に探索することで高精度な近似解を発見することが可能な群知能 (Swarm intelligence; SI) の一手法であり, 粒子の振る舞いが確率的なダイナミクスに基づくアルゴリズムである。他のメタヒューリスティクスと比較したときの PSO の特徴として, (1) 複雑な処理を必要としない簡素なアルゴリズム, (2) 制御パラメータ数が比較的少ない, (3) 発見した良解への高い収束性能, (4) 高い並列計算性, がある。

PSO は様々な他のメタヒューリスティクスと比較して優れた解探索性能を有することが示されており, その計算時間も短いことが明らかにされている。そのため, PSO は様々な工学的な分野における最適化問題に適用され, その有効性が示されている強力なメタヒューリスティクスの一つである。

PSO に関する研究は図 1.1 に示すように 5 つに分類できる。

- (1) 解探索性能の向上
- (2) 理論解析
- (3) 並列計算環境への実装
- (4) 適用可能な問題の拡張
- (5) ハードウェア化

(1) に属する研究は既存の PSO の解探索性能を向上させ、様々な最適化問題に対して高精度な近似解を求めることを目的としている研究である。PSO の粒子間の近傍構造 (*i.e.*, 粒子間の情報共有の構造) にグラフ理論 (Graph theory) などのネットワーク構造を導入する手法、粒子群の母集団を複数の小規模なサブ集団に分割し、解空間を複数のサブ集団で並列的に探索するマルチスウォーム (Multi-swarm) 手法、PSO のアルゴリズム自体に新しい処理の導入や粒子更新式を改良する手法、そして、既存の他のメタヒューリスティクス (*e.g.*, 遺伝的アルゴリズムなど) の良い性質を PSO に組み込む手法、がある。

(2) に属する研究は PSO の粒子の動作を解析し、PSO の解探索性能を制御するパラメータと粒子が収束する条件との関係を明らかにすることを目的としている研究である。解析を容易にするために PSO の持つ確率要素を排除した決定論型の PSO (Deterministic PSO; D-PSO) を用いてパラメータによる粒子軌道の収束条件などの理論解析が行われており、PSO のパラメータ選択方法の指針を与えている。

(3) に属する研究はマルチコア、PC クラスタや GPU (Graphics processing unit) などの並列計算環境を用いて PSO のアルゴリズムを並列化し、その計算量を分散させることでより短い計算時間で近似解を発見することを目的としている研究である。PSO では、粒子ごとに計算が独立しているため、並列計算環境への実装は容易であり、多数の粒子を並列的に計算することで計算時間の低減が図られる。

(4) に属する研究は解空間が動的に変化する動的最適化問題、組合せ最適化問題などの離散最適化問題や目的関数が複数存在する最適化問題である多目的最適化問題などに対して PSO を適用し、問題に対する近似解を発見することを目的とした研究である。標準的な PSO は解空間が変化しない静的な問題であって設計変数が連続値をとる最適化問題を解くために設計されている。よって、上記のような最適化問題を解くためには PSO アルゴリズムの改良が必要となる。

(5) に属する研究は PSO アルゴリズムを組み込みアプリケーションに実装し、最適化を行うことや粒子計算を高速に行うための専用プロセッサの開発など PSO アルゴリズムをハードウェア上に実装することを目的としている研究である。高速に粒子演算が可能な専用プロセッサを多数用いて並列計算させることで (3) に属する研究と同様に計算時間の低減ができる。

本論文では、(1) に属する PSO アルゴリズムの解探索性能を向上させる手法の研究を (3) に属する並列計算環境への実装を考慮して行うものである。また、(2) に属する粒子軌道の安定性についても議論する。PSO の解探索性能を向上させる手法は多数提案されているが、これらの多くは並列計算環境への実装を考慮しておらず、並列計算環境への実装の研究についても標準的な PSO を用いた手法が多い。このため、解探索性能を向上させ、かつ計算時間の短縮を考慮する PSO アルゴリズムの研究は有意義である。

PSO アルゴリズムの解探索性能の向上を図ることを目的とした研究を行うに際して、PSO の解探索性能の低下要因を正確に把握することが不可欠である。標準的な PSO の粒子は全粒子の探索し

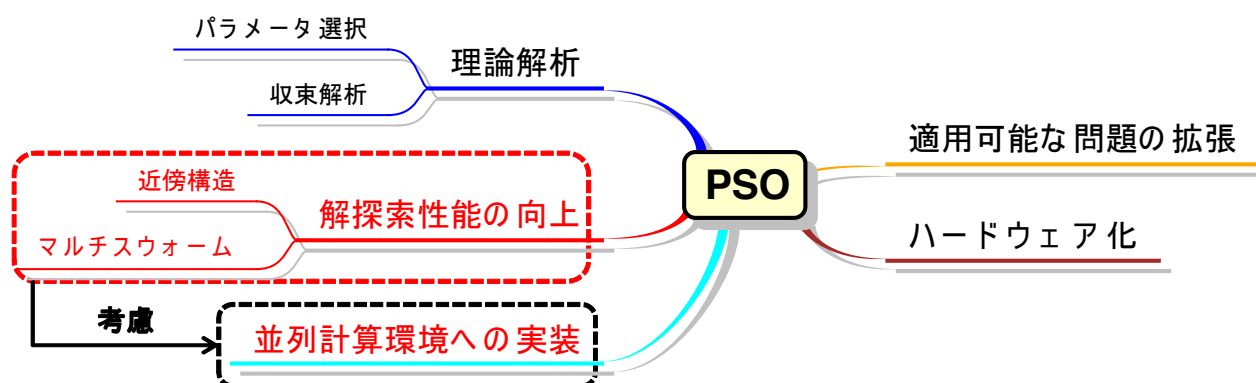


図 1.1: PSO の研究の分類と位置付け

てきた過程の最良解の情報 ($gbest$) を共有する全結合型の近傍構造 (Fully connected topology) を有し、粒子群は $gbest$ の方向へ探索を行う。この特性により PSO は発見した解に対する高い収束性を有するが、解の多様性 (Diversity) が失われやすい要因となっている。本論文における多様性は解収束に至る (全ての粒子が同じ位置に至る) までの時間 (イタレーション) の長さを意味する。多様性が高い最適化手法の場合は解収束まで時間を要するが、逆に多様性が低い最適化手法の場合は解収束が早いことを表す。このため、標準的な PSO は多峰性問題に対して十分な探索を行う以前に局所解へ初期収束 (Premature convergence) してしまう。したがって、PSO アルゴリズムの多様性を高めて解探索性能を向上させることで初期収束も回避できる。多様性を向上させる手法では、粒子群が長時間に亘って解探索を行うことで解探索空間を広げることができ、最適解への収束可能性が増加するものと考えられる。

本論文と関係が深い関連研究についての詳細な説明は 3 章で行うため、本章では、次の関連研究について簡単に説明する。

- 粒子間の近傍構造に関する手法
- マルチスウォーム (Multi-swarm) 手法

「粒子間の近傍構造に関する手法」における代表的なものとして、グラフ構造の概念を粒子間の近傍構造に導入した手法がある。各粒子は近傍構造で定められた近傍粒子とのみ情報共有を行う。近傍の粒子数は次数 (Degree) により決定され、次数が小さいほど粒子は独立して探索を行うようになり、PSO アルゴリズムの多様性は高くなる。一方、次数が大きいほど粒子は多くの他粒子の影響を受けて最良解へ集まるようになるため多様性は低くなる。

「マルチスウォーム手法」は粒子群の母集団を複数の小規模なサブ集団に分割し、サブ集団同士が相互作用しながら解空間の探索を行う手法である。各サブ集団は独立して解空間を探索するため、アルゴリズムの多様性を高く維持しながら探索を行うことができる。また、複数のサブ集団を用いるため、単一の近似解だけでなく、複数の近似解を並列的に求める手法も多数提案されている。

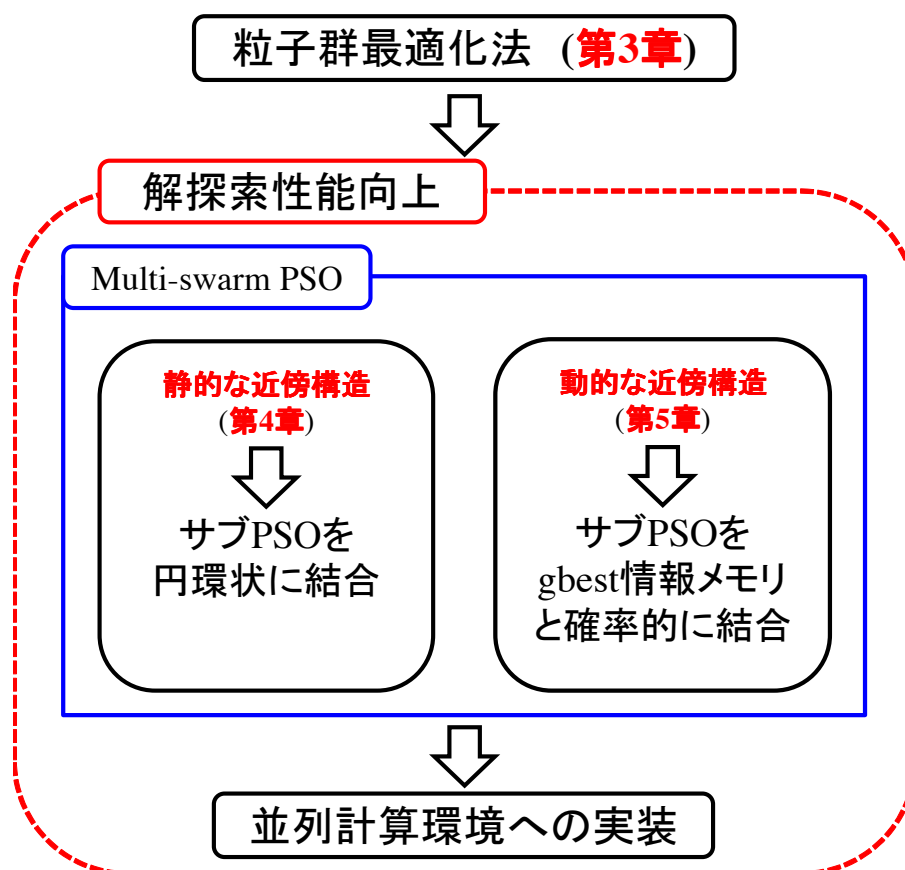


図 1.2: 本論文の構成

近年の技術の発達により、取り扱えるデータ量やシステムの規模は大きくなってきており、データの分類や計算流体力学 (Computational fluid dynamics; CFD) のシミュレーションなどにおける最適化の計算量は大きくなってきている。よって、このようなブラックボックス最適化問題に対する最適化は PSO を用いたとしても膨大な計算時間を要する。また、このような大規模な最適化問題においては、小数の粒子を用いて高精度な近似解を発見することが難しく、多数の粒子を用いることが要求されている。並列計算環境を用いて PSO の多数の粒子群の計算を並列分散的に計算することで計算時間を低減することと並列計算環境への実装に適した PSO アルゴリズムの解探索性能の向上を図る手法は大規模・複雑な最適化問題において意味深い。

本論文では、PSO の解探索性能の向上を並列計算環境への実装を考慮して行うものとする。なお、本論文の提案手法ではそれらの係数が探索の状況によらず不変の固定係数型の手法である。図 1.2 に示すように、本論文はマルチスウォーム PSO に対して分割したサブ集団間に近傍構造によるネットワーク化を導入した手法により構成される。各章の内容について簡単に言及する。

2 章では、最適化問題およびメタヒューリスティクスの分類とその紹介を行う。

3 章では、標準的な PSO および、PSO の解探索性能の向上と PSO を並列計算環境に実装する従来研究をその分類を含め総括する。

4章では、並列計算環境への実装を考慮したマルチスウォーム PSO に固定的な近傍構造を導入した手法 (PSO Networks; PSON) を提案する。提案手法では、複数のサブ PSO (サブ集団) が並列的に解空間を探索することができる。各サブ PSO は情報交換することで局所解からの脱出や良好な解探索を実現できる。また、提案手法は並列計算環境に実装することが可能な構造であり、そのことによって計算時間が低減されることが期待される。すなわち、1つのサブ PSO を1つの計算機(プロセッサ)に割り当て、サブ PSO 間の情報交換をプロセッサ間通信により行うモデルである。提案するアルゴリズムでは、標準的な PSO の更新式に加えて近傍サブ PSO 間の良解情報も考慮しており、この計算項の追加による粒子の収束性の解析を行い、収束条件を明らかにする。また、サブ PSO 間の近傍構造は次数により決定するため、次数は解探索性能に大きく影響を及ぼすことが考えられる。このため、次数と解空間の景観 (Landscape) の関係性を数値実験により明らかにし、解探索性能の有効性を明らかにするために様々な近傍構造を有する PSO 手法との比較実験を行う。

5章では、4章で提案した PSON の解探索性能を向上させる手法を提案する。PSON は多峰性問題において、その難易度によっては解探索性能が不十分である場合がある。これは、PSON のネットワーク構造が固定的であることでネットワークのダイナミクスが制限されているためである。そこで、PSON の多様性を向上させ、多峰性問題に対して良好な近似解を探索できるネットワーク構造を提案する。5章の提案手法 (PSON with Stochastic Connection; PSON-SC) では、サブ PSO は最良解情報を保持するメモリ (gbest 情報メモリ) と確率的に通信し、最良解情報の更新や参照ができる動的な近傍構造を導入する。gbest 情報メモリと通信しない場合は、サブ PSO は独立して解空間の探索を行うため、gbest 情報メモリと通信する確率が小さいほど多様性が向上する。また、動的に近傍構造が変化するため、PSON よりも大域的に解空間を探索することができる。2つの提案手法における多様性を調査し、PSON-SC の多様性が PSON よりも高いことを数値実験により明らかにする。さらに、PSON-SC の有効性を確認するために PSON を含む様々な近傍構造を持つ PSO 手法との解探索性能の比較実験を行う。

6章では、本論文の結論について述べる。

第二章 本論文に関わる基本事項

2 本論文に関わる基本事項

本論文では、並列計算環境への実装を考慮した PSO の解探索性能の向上手法の提案を行う。まず、最適化問題について述べ、本論文で対象とする最適化問題の分類と位置付けを明らかにする。次に、メタヒューリスティクスの分類について述べ、PSO の位置付けを明らかにする。

2.1 最適化問題

工学の目的は人間社会に有用な機器、構造物あるいは各種の設備やそれを運用するシステムなどを創生することであり、そのための作業は一般に工学設計と呼ばれている [2.2]。工学設計では、設計過程において考慮しなければならない点が多く、それらを満たす候補案も多い。このような状況において、その選択を経験や勘に頼るのではなく、科学的に行うことが必要である。このような科学的な最適化の手段を用いることを前提とした設計方法を最適設計方法と呼ぶ [2.2]。最適設計は従来の設計の効率化ばかりでなく、質の向上や設計・生産システムの統合化などの様々な面で新たな脚光を浴びている [2.2]。

生産活動、販売活動、投資決定などのオペレーション (Operation) に対しての意思決定に対しても、経験や勘のみに基づいて行うのではなく、長期的な観点から合理的に決定することが望ましく、問題に対して最適な意思決定を行うことができるように問題を定式化することが必要である [2.4]。

オペレーションに対して数学的なアプローチによって意思決定を支援する分野がオペレーションズ・リサーチ (Operations research; OR) である。オペレーションズ・リサーチをフォーマルに定義すると「合理的な意思決定を行うためにモデルや手法の開発を研究し、人や組織の行動・現象を数量的に把握し分析する学問」である [2.4]。OR はその黎明期において軍事上の問題解決のために用いられていたが、今では、在庫管理、生産計画、輸送問題など様々な分野において“最適な意思決定”を行うために用いられている。

OR における科学的意思決定方法を次に示す [2.4]。

Step 1 : 問題の確認

問題となっている現象を観察し、問題を明確化する。

Step 2 : 問題の設定

問題解決のために利用可能な決定手法・戦略の全体を明確化する。

Step 3 : 問題の数学的定式化

問題の基本的現象を全体的に理解してモデルを作成する。

Step 4 : 解の決定

モデルが問題の本質を捉えているかを検討し、論理的に導き出せる最良解を発見する。

Step 5 : 解の実施

モデルから得られた解が実行可能であるかや操作の方法を検討し、解決案を実施する。

このように問題を数理計画 (Mathematical programming) により定式化し、その数式を解くことで最適な状態 (Optimal solution; 最適解) を求めることができる。数理計画は対象とする問題に対する最適な解決策を求めるための定式化の手法であり、解決すべき問題を最適化問題 (Optimization

problem) として記述し、数学的に厳密なアルゴリズムを用いて解くアプローチをとる [2.1]. すなわち、最適化問題は OR の一部と考えることができる. 数理計画法が扱う代表的な数理モデルが線形計画法 (Linear programming; LP) や動的計画法 (Dynamic programming; DP) であり、広い分野で用いられている.

数理計画、最適化理論 (Optimization theory) は、理論だけでなく現実の問題を解くための手法が提案されてきた [2.3]. 1947 年に G. B. Dantzig によって線形計画問題に対する解法としてシンプレックス法を提案されてから、応用数学の一つの独立した分野として認識されるようになった. 最適化問題は、線形計画問題から非線形計画問題 (連続最適化問題)、整数計画問題、離散最適化問題 (組合せ最適化問題) と拡大し、数学の他分野との関連を深めつつ理論的にも発展し続けている.

最適化問題とは「与えられた制約条件 (Constraints) のもとで、目的関数 (Objective function) が最小 (最大) となるような設計変数 (Design variables) の値を見つける」問題である [2.1]. 最適化問題は一般的に次のように定義される [2.2],[2.5],[2.9].

$$\begin{aligned} & \underset{\mathbf{x} \in \mathcal{R}}{\text{minimize}} \ f(\mathbf{x}) \\ & \text{subject to } \mathbf{x} \in \mathcal{F} \end{aligned} \quad (2.1)$$

f を目的関数, \mathcal{F} を実行可能領域 (Feasible region), $\mathbf{x} = (x_1, x_2, \dots, x_n)$ を設計変数と呼ぶ. n は設計変数の総数を表す. 目的関数 $f(\mathbf{x})$ は実数値または整数値をとる関数

$$f : \mathcal{F} \rightarrow \mathcal{R}$$

である.

制約条件は

$$g_j(\mathbf{x}) \leq 0 \ (j = 1 \sim m) \quad (2.2)$$

$$h_k(\mathbf{x}) = 0 \ (k = 1 \sim m') \quad (2.3)$$

であり、式 (2.2) を不等式制約 (Inequality constraints), 式 (2.3) を等式制約 (Equality constraints) と呼ぶ. ここで、 m は不等式制約の総数、 m' は等式制約の総数を表す.

式 (2.1) は目的関数の値を最小化する最小化問題であるが、目的関数の値を最大化する最大化問題を考えることもできる. つまり、 $f(\mathbf{x})$ を最小または最大にする実行可能解 (Globally optimal solution; 大域最適解 または 最適解) を見つけることが最適化問題の目的である. また、 $f(\mathbf{x})$ の実行可能な近傍の解集合においてその解よりも良好な解がない場合は、 \mathbf{x} を局所最適解 (Locally optimal solution; 局所解) と呼ぶ. $\mathcal{F} \subseteq \mathbb{R}$ (実数) を満たす場合は、式 (2.1) は連続最適化問題と呼ばれる. また、 \mathcal{F} が組合せ的な構造を持つ (i.e., $\mathcal{F} \subseteq \mathbb{Z}$ (整数)) 場合は、式 (2.1) は組合せ最適化問題または離散最適化問題と呼ばれる.

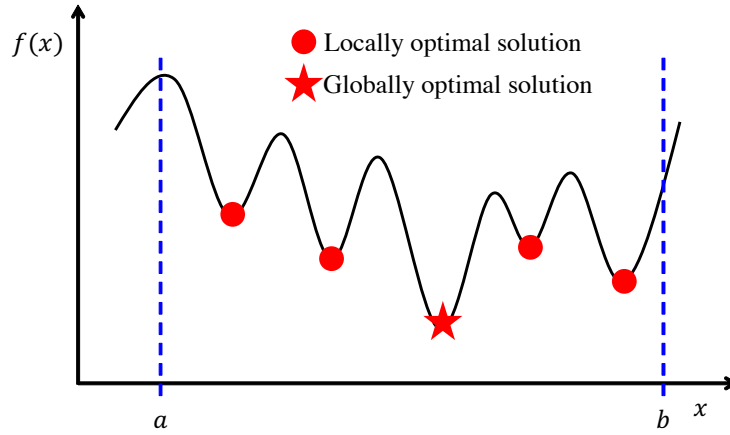
図 2.1: 1次元の目的関数 $f(x)$ の景観例

図 2.1 に最小化問題における連続的な 1 次元の目的関数 $f(x)$ の例を示す。図 2.1 において、定義域が $[a, b]$ であり、この領域の解空間においては 4 つの局所解と 1 つの最適解が存在する問題である。このように問題の評価値を図的に表現したものを景観 (Landscape) と呼び、問題の景観に応じて最適化問題の特徴が変化する。図 2.1 のように多数の峰が存在する問題を多峰性問題 (Multimodal problem)、一方、峰が 1 つだけ存在する問題を単峰性問題 (Unimodal problem) と呼ぶ。多峰性問題は解空間上に局所解が多く、得られた解が最適解ではなく、局所解のみしか探索できないことがあり、単峰性問題よりも最適解を発見することが困難な問題である。

最適化問題の分類を図 2.2 に示す。図 2.2 より、最適化問題は動的最適化問題 (Dynamic optimization problems) と静的最適化問題に分類できる。動的最適化問題は時間経過やノイズの影響などにより解空間が変化する問題であり、最大電力点追従 (Maximum power point tracking; MPPT) 問題 [2.12] などがある。一方、静的最適化問題は解空間が変化しない問題であり、巡回セールスマン問題、ナーススケジューリング問題や FNSP (Forwarding node selection problem) [2.13] などがある。これらの問題は最適化する目的関数が 1 つの単目的の最適化問題である。

最適化する目的関数が 2 つ以上の場合が多目的最適化問題である。多目的最適化問題では、複数の目的関数を最適化することが望ましいが、それらを同時に最適化できる場合は少ない。これは、複数の目的関数は互いにトレードオフの関係を有することが多いためであり、パレート最適解と呼ばれる解集合から設計者が解を選択する必要がある。例えば、飛行機の遷音速翼の設計 [2.14] では、トレードオフのある抗力 (最小化) と重量 (最小化) の多目的最適化を MOEA (Multi objective evolutionary algorithm) を用いて行っている。本論文では、静的な単目的の連続最適化問題を取り扱うものとする。

最適化問題に対して最適解を計算機を用いて求める場合は計算機のリソースと時間量 (Time complexity) を知る事が重要である [2.7],[2.9]。リソースは計算速度、主記憶装置の容量 (メモリ)、補助記憶装置の容量などである。データ数が膨大な問題においては一般的なコンピュータで用いられる

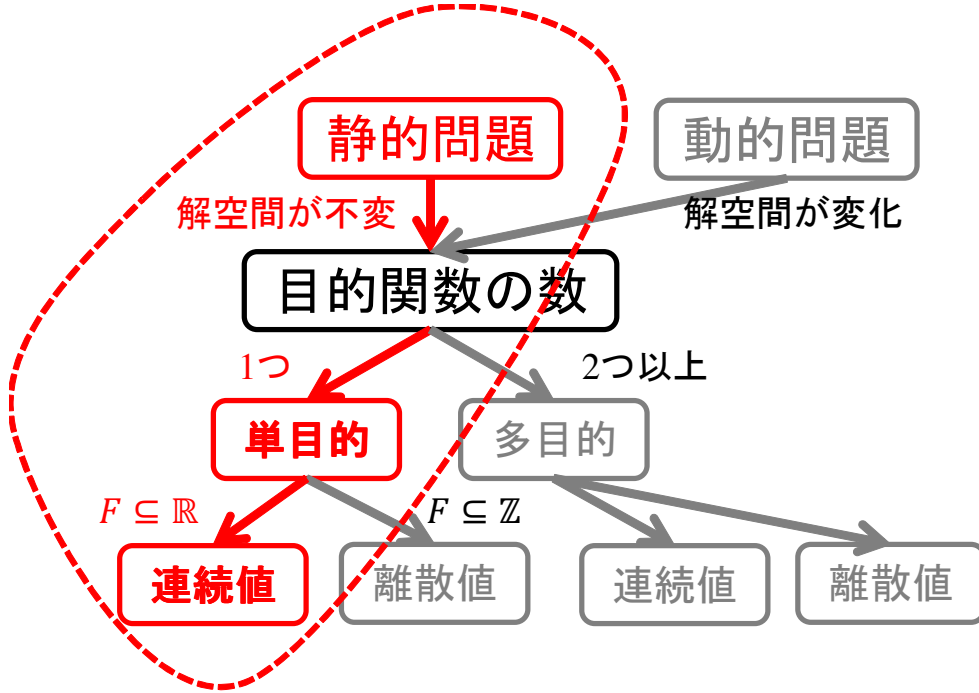


図 2.2: 最適化問題の分類

主記憶装置に全てのデータを取り込むことは困難である。一方、時間量は計算の各基本的な操作を単位とみなして、それらの実行回数を表す。これら2つの指標をまとめて計算複雑度 (Complexity) と呼ぶ。ある規模 (入力サイズ) n の組合せ最適化問題を解く場合は、アルゴリズムの計算時間が多項式オーダー (Polynomial order) である場合はそのアルゴリズムを多項式時間アルゴリズム (Polynomial time algorithm) と呼ぶ。多項式オーダーとは、問題の規模 n に対して、計算時間が

$$f(n) = \sum_{i=0}^k a_i n^i \quad (2.4)$$

であり、そのオーダーが $O(n^k)$ で表せることである。ここで、 k は定数値、 $a_i \geq 0$ を満たす値である。例えば、 n 個の数字を昇降順序に並び替えるソーティングアルゴリズムのオーダーは $O(n^2)$ である。このような問題を P (Polynomial time) 問題と呼ぶ。一方、アルゴリズムの計算時間が $O(n^{\log n})$, $O(k^n)$, $O(n^n)$ のように多項式オーダーとならないことがある。このような問題を NP (Nondeterministic polynomial time) 問題と呼ぶ。P 問題の集合を P クラス、NP 問題の集合を NP クラスと呼び、このような問題は実世界では多く存在する。NP 問題の中で最も難しい問題を NP 完全 (NP-complete) と呼び、NP 完全と比較して、それと同等以上に難しい問題を NP 困難 (NP-hard) と呼ぶ。実世界の組合せ最適化問題の多くは NP 困難であり、このような問題に対しては列挙法などの全ての解の組合せをしらみつぶしに探索する厳密解法 (Exact algorithm) では、最適解を発見するために天文学的な時間を必要とする場合も少なくない。また、連続最適化問題においても高次元 (設計変数の数が多い)

い) 問題になるに伴い解空間 (Solution space) は拡大する。したがって、厳密解法を用いて最適解を探索することは現実的には不可能である。

このように実世界には様々な最適化問題が存在するが、それらの最適化問題の解法として、線形計画法、非線形計画法、動的計画法などが提案されている。探索空間が連続で感度分析 (Sensitivity analysis) [2.1] が可能であり、かつ目的関数の景観が単峰性である場合は線形計画法や非線形計画法は非常に有効である [2.2]。しかし、一般の問題においては目的関数の景観は多峰性である場合が多く、探索空間が不連続な離散空間の場合や感度分析が行えない場合もある。このような場合は線形計画法や非線形計画法を最適化問題の解法として利用することはできない。また、線形計画法は目的関数および制約条件式が線形関数のみに限られ、動的計画法は非線形関数に対しても適用可能であるが、大規模な組合せ最適化問題や高い非線形性を有する問題に対しては非効率的である [2.11]。

問題のモデルを定式化することが可能であれば、非線形関数においてもそのモデルの目的関数の勾配 (Gradient) を用いることで問題に対する解を求めることができる。勾配情報を用いる最適化手法を勾配法 (Gradient method) と呼び、最急降下法 (Steepest decent method)、ニュートン法 (Newton's method)、準ニュートン法 (Quasi-Newton method)、共役勾配法 (Conjugate gradient method) などがある [2.3]。これらの解法では、目的関数の勾配が $\mathbf{0}$ となる点を見つけることを目的としているため、一般に微分不可能な問題に対しては適用することができない。

これまで説明してきた最適化問題の解法はいずれも対象とする問題が具体的な数式で表すことができる最適化問題に対してのみ有効である。しかし、世の中の多くの問題では、数式で定式化できないことがある。このように定式化することはできないが、設計変数 \mathbf{x} の目的関数値 $f(\mathbf{x})$ のみ知ることができる問題を “Black-box optimization” (ブラックボックス最適化問題) と呼ぶ [2.6]。また、定式化が可能な問題においても、その定式化には深い専門的な知識を求められることがある [2.5]。

ブラックボックス最適化問題は実世界において多数存在するが、このような場合でも、計算機上のシミュレーションによって最適化する目的関数値を計算できることがある [2.5]。ただし、シミュレーションによって計算された目的関数値には一般的に誤差やノイズが含まれ、勾配情報を計算することも困難である。そのため、シミュレーションによって求まる目的関数値のみを用いて最適解を求める手法の研究が活発になされている [2.5],[2.6]。例えば、オプション価格の推定にモンテカルロ法によって価格計算を行うことなどがある [2.5]。

線形計画法や動的計画法は解の厳密性に重点を置くため、それらによって解ける問題の範囲も限定されていたのに対して、その厳密性は少々犠牲にしてもより広い適応解を求めることに重点を置く、ヒューリスティック解法 (Heuristics) が様々な提案されている [2.2]。

2.2 メタヒューリスティクス

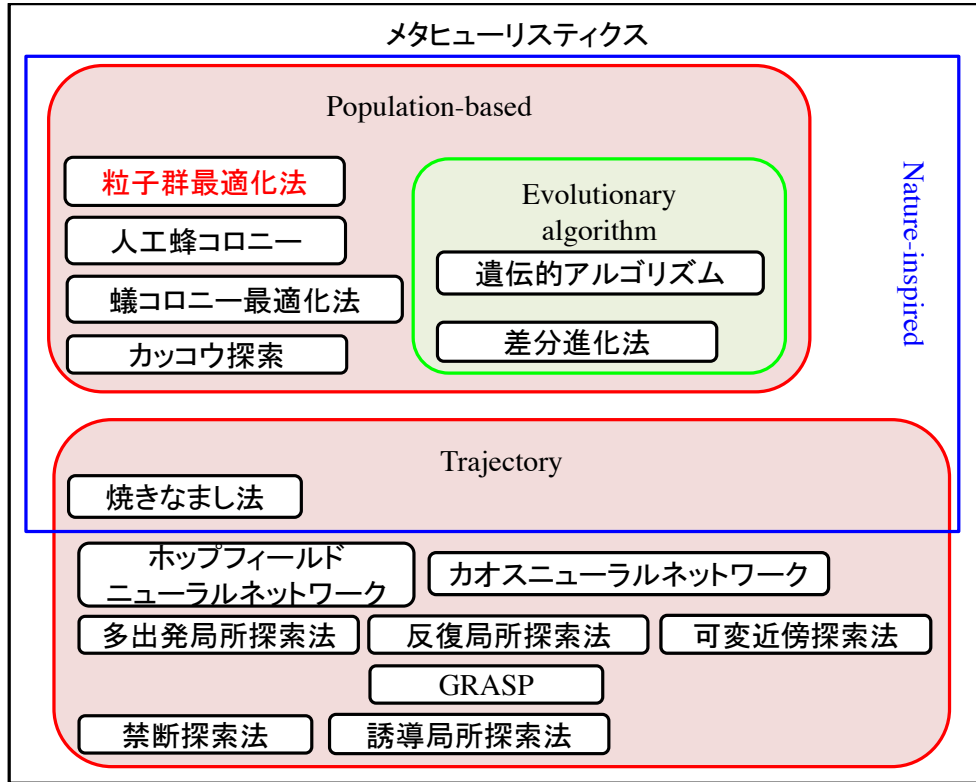


図 2.3: メタヒューリスティクスの分類

ヒューリスティクス (Heuristics; 発見的手法) とは、「許容性や最適性の保証がないが、実用的な計算時間内に良好な近似解あるいは最適解を発見する手法」のことである [2.11]. ただし、ヒューリスティクスは特定の問題にしか適用できないことが多い [2.7].

このようなヒューリスティクスに近似解に修正を加えながらより最適解に近づけようとする戦略をメタヒューリスティクス (Metaheuristics) と呼び、広いクラスの問題に適用可能となった [2.7]. ここで、メタ (meta) とは、「high level (上級)」, 「beyond (超えた)」 [2.10],[2.11] を意味する接頭語である.

メタヒューリスティクスは NP 困難な組合せ最適化問題やブラックボックス最適化問題に対して有力な解法であり、様々な種類の手法が提案されている [2.2],[2.15]-[2.17]. これらのメタヒューリスティクスに共通する操作として (1) 過去の探索履歴を利用し新しい解を生成する, (2) 生成した解を評価し、次の解探索に必要な情報をフィードバックする、があり、これらの操作の反復により解空間の探索を行う [2.9].

多くのメタヒューリスティクスにおいて、「近傍」の概念が用いられている [2.8]. 近傍とは、ある解が与えられたとき、その解に小さな摂動を加えて得られた解の集合を表す. 実行可能解 $x \in \mathcal{F}$ において、 $f(x) \leq f(y)$, $\forall y \in N(x)$ を満たすものは近傍 N に対する局所解であり、多くのメタヒューリ

スティクスは近傍を基礎として構築されている。これは「良い解同士は似通った構造を持つ」POP (Proximate optimality principle) の概念に基づいているためである。また、ランダム性の利用も多く、メタヒューリスティクスに用いられている。ランダム性は近傍探索からより大域的な探索に遷移することを可能とし、局所解に収束することを避けることができる。ただし、問題の最適化においては毎回違う解が生成されることがあるため注意する必要がある。

メタヒューリスティクスの主要な要素として集中化 (Intensification または Exploitation) と多様化 (Diversification または Exploration) がある [2.7]-[2.10]。集中化は POP の概念に基づいて良解の近傍を集中的に探索して局所解を発見することである。一方、多様化は集中化とは逆にこれまで探索してきた解とは異なる構造を持つ解を生成することである。つまり、解空間を大域的に探索し、これまでに探索していない領域を探索することである。集中化によって良解への収束を促し、多様化によって局所解の回避や多様性を向上することができる。メタヒューリスティクスにおいて集中化と多様化のバランスを適切にとることで効率的な解探索を実現することができる。

図 2.3 にメタヒューリスティクスの分類を示す [2.8],[2.9],[2.11],[2.15]。図 2.3 において、メタヒューリスティクスは母集団型 (Population-based) と軌道型 (Trajectory) の 2 つの種類に分類できる。

軌道型の手法は探索個体 (Agent) が 1 つのみの単点探索型のメタヒューリスティクスである。多出発局所探索法 (Multiple start local search; MLS), 反復局所探索法 (Iterated local search; ILS), 模擬焼きなまし法 (Simulated annealing; SA) [2.18], 禁断探索法 (Tabu search; TS) [2.19],[2.20], 誘導局所探索法 (Guided local search; GLS), GRASP (Greedy randomized adaptive search procedure), 可変近傍探索法 (Variable neighborhood search; VNS), ホップフィールドニューラルネットワーク (Hopfield neural network; HNN), カオスニューラルネットワーク (Chaotic neural network; CNN) [2.21] などがある。軌道型の手法では、個体の近傍内に改善解がなくなるまで移動操作を続ける。このため、多峰性問題においては局所解に収束しやすく、発見した解の質が低下しやすい。SA や CNN は近傍に良好な解が存在しなくてもその局所解からの脱出ができるため、大域的に解空間を探索できる強力な手法である。

母集団型の手法は複数の探索個体を用いてこれらの探索個体間の情報交換により良好な近似解を探索する多点探索型のメタヒューリスティクスである。遺伝的アルゴリズム (Genetic algorithm; GA) [2.22], 蟻コロニー最適化法 (Ant colony optimization; ACO) [2.23], 粒子群最適化法 (Particle swarm optimization; PSO) [2.24], 差分進化法 (Differential evolution; DE) [2.25], 人工蜂コロニー (Artificial bee colony; ABC) [2.26], カッコウ探索 (Cuckoo search; CS) [2.27] などがある。GA や DE は進化的アルゴリズム (Evolutionary algorithm; EA) の一種であり、その他は群知能 (Swarm intelligence; SI) に属する手法である。また、これらの手法は “Nature-inspired algorithm” と呼ばれている。

母集団型の手法では、多数の探索個体が並列的に探索するため軌道型のメタヒューリスティクスよりも多様な探索ができる。一般的には多数の探索個体を制御するような中央処理機構は存在せず、探索個体同士の相互作用のみ用いて解探索を行うため、探索個体間の独立性は高い。多峰性問題に

表 2.1: メタヒューリスティクスの特徴

	GA	PSO	DE	ABC	CS
発表年	1975	1995	1997	2005	2009
解探索性能 ($n = 2 \sim 20$)	Δ	\bigcirc	\bigcirc	\times	\times
解探索性能 ($n = 20 \sim 30$)	Δ	\times	\bigcirc	Δ	\bigcirc
パラメータ数	7	4	4	4	2
パラメータ	G	w	F	N_o	P_a
	:Gray coding の有無	:慣性速度定数	:Scale parameter	:Onlooker 数	:廃棄確率
	E	c_1	C_r	N_e	N
	:エリート保存の有無	:加速度係数	:交叉確率	:Employed Bee 数	:母集団サイズ
	S	c_2	S	N_s	—
	:選択方法	:加速度係数	:戦略 ($x/y/z$)	:Scout 数	—
	C	P	P	$Limit$	—
	:交叉方法	:母集団サイズ	:母集団サイズ	:Scout Limit	—
	M	—	—	—	—
	:突然変異方法	—	—	—	—
	A	—	—	—	—
	:解精度	—	—	—	—
	P	—	—	—	—
	:母集団サイズ	—	—	—	—
利点	・適切なパラメータ設定 を選択時に解探索性能が 高い	・パラメータ数が少ない ・アプリケーション適用 が容易 ・粒子の並列計算性が高い	・パラメータ数が少ない ・母集団の多様性が高い	・パラメータ数が少ない ・アプリケーション適用 が容易	・パラメータ数が少ない ・母集団の多様性が高い
欠点	・交叉, 突然変異がラン ダムのため, 解への早期 収束が未保証	・母集団の多様性が低い	・解への収束が遅い	・局所探索が不十分	・ランダム探索に依存 しているため, 解への 早期収束が未保証

* 解探索性能は $\bigcirc > \Delta > \times$ の順に優れている* n は最適化すべき設計変数の総数 (次元数) を表す

においても高精度な質の近似解を探索できるが, 多数の探索個体の計算を必要とするため, 計算量は軌道型の手法よりも多い。

本論文では, 多峰性問題に対しても有力な母集団型のメタヒューリスティクスを取り上げる。母集団型のメタヒューリスティクスにおける有力な手法の性能比較を行った結果を表 2.1 にまとめた [2.28]-[2.30]。なお, 表における解探索性能は計算量まで考慮している。表 2.1 において, 次元数が比較的低い ($n = 2 \sim 20$) 問題に対しては PSO, DE は優れている。PSO はその粒子群 (探索個体群) が最良解の方向へ移動する習性を有するため, 粒子群は最良解へ収束しやすく, 多様性は低下しやすい。ただし, このような解空間が比較的に狭い問題に対しては素早く高精度な近似解を発見することができる。なお, GA, ABC は局所探索性能が弱く, CS はその探索性能がランダム性へ強く依存しており, これらの手法は良解への収束までに時間 (イタレーション) がかかる手法である。

次元数が比較的高い ($n = 20 \sim 30$) 問題に対しては DE, CS は優れた手法である。探索すべき解空間が広く, 母集団の多様性が高い手法は大域的に解空間を探索できるため, 最適解を発見できる可能性も高くなる。DE や CS は探索個体群が最良解の影響を受けにくい手法であるため, それらの探索個体は解空間を独自に探索する傾向が強くなるため, その多様性は PSO と比較すると高く維持することができ, 多峰性問題に対しても局所解に陥りにくく, 良解を探索できる。

表 2.1 より, PSO の粒子群はその解空間が多峰性の場合は局所解へ収束しやすく, 解空間が広大になるにつれて最適解を発見できる可能性が低くなる。このような解空間が広大かつ複雑なブラックボックス最適化問題に対して高精度な解を発見するためには, PSO の多様性を向上させる工夫が

必要となる。なお、メタヒューリスティクス手法において多様性を増加させることは解収束が遅くなることを意味することから、解へ収束するまでの計算量が増大する。ただし、PSO は粒子の並列計算性が高く、並列計算環境への実装も容易である。よって、PSO アルゴリズムの多様性の向上を高い計算並列性を維持して成し遂げることで計算量が増加したとしても多数の計算機(プロセッサ)に分散することで計算時間は抑えることが可能となる。

本論文では、PSO の解探索性能および並列計算性に注目し、その問題点を改善する手法を提案し、その有効性を明らかにする (4, 5 章)。

参考文献

- [2.1] 茨木 俊秀, 福島 雅夫, “最適化の手法.” 共立出版株式会社, 1993.
- [2.2] 山川 宏, “最適設計ハンドブック -基礎・戦略・応用-.” 朝倉書店, 2003.
- [2.3] 茨木 俊秀, “最適化の数学.” 共立出版株式会社, 2011.
- [2.4] 小和田 正, 沢木 勝茂, 加藤 豊, “OR 入門 意思決定の基礎.” 実教出版株式会社, 2013.
- [2.5] 山下 信雄, “非線形計画法.” 朝倉書店, 2015.
- [2.6] 田邊 遼司, 串田 淳一, 畠中 利治, “関数最適化における進化計算,” 計測と制御, Vol. 54, No. 8, pp. 567-572, 2015.
- [2.7] 古川正志, 川上敬, 渡辺美知子, 木下正博, 山本雅人, 鈴木育男, “メタヒューリスティクスとナチュラルコンピューティング.” コロナ社, 1 2012.
- [2.8] 久保幹雄, J. P. ペドロソ, “メタヒューリスティックの数理.” 共立出版, 5 2009.
- [2.9] 柳浦睦憲, 茨木俊秀, “組合せ最適化 -メタ戦略を中心として-.” 朝倉書店, 1 2001.
- [2.10] X. S. Yang, “Nature-Inspired Metaheuristic Algorithms Second Edition.” Luniver Press, 2010.
- [2.11] Z. Beheshti and S. M. H. Shamsuddin, “A review of population-based meta-heuristic algorithm,” *Int. J. Advance. Soft Comput. Appl.*, Vol. 5, No. 1, pp. 1-35, March 2013
- [2.12] T. Eswar and P. L. Chapman, “Comparison of photovoltaic array maximum power point tracking techniques,” *IEEE Trans. Energy Convers.*, Vol. 22, No. 2, pp. 439-449, 2007.
- [2.13] T. Sasaki, H. Nakano, A. Utani, A. Miyauchi and H. Yamamoto, “An efficient flooding scheme using chaotic neural networks in wireless sensor networks,” in *Proc. of NOLTA*, pp. 523-526, 2009.
- [2.14] A. Oyama, “Multidisciplinary optimization of transonic wing design based on evolutionary algorithms coupled with CFD solver,” *CD-Rom Proc. of ECCOMAS*, pp. 1-17, 2000.
- [2.15] 柳浦 睦憲, 茨木俊秀, “組合せ最適化問題に対するメタ戦略について,” 電子情報通信学会論文誌 *D-I Vol. J83-D-I No. 1*, pp. 3-25, 2000.
- [2.16] X. S. Yang, “Engineering optimization : an introduction with metaheuristic applications.” John Widely & Sons, 2010.

- [2.17] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, Vol. 52, pp. 141-152, 1985.
- [2.18] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by simulated annealing," *Science*, 220, pp. 671-680, 1983.
- [2.19] F. Glover, "Tabu search I," *ORSA Journal on Computing* 1, pp. 190-206, 1989.
- [2.20] F. Glover, "Tabu search II," *ORSA Journal on Computing* 2, pp. 4-32, 1989.
- [2.21] K. Aihara, T. Takabe and M. Toyoda, "Chaotic neural networks," *Phys. Lett. A*, Vol. 144, pp. 333-340, 1990.
- [2.22] J. Holland, "Adaption in natural and artificial systems," *University of Michigan Press, Ann Arbor*, 1975.
- [2.23] M. Dorigo, "Learning and natural algorithms," *PhD thesis, Politenecnico di Milano, Italy*, 1992.
- [2.24] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. of IEEE Int. Conf. on Neural Networks*, pp. 1942-1948, 1995.
- [2.25] R. Storn, K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, Vol. 11, No. 4, pp. 341-359, 1996.
- [2.26] D. Karaboga, "An idea based on bee swarm for numerical optimization," *Tech. Rep. TR-06, Erciyes University, Engineering Faculty, Computer Engineering Department*, 2005.
- [2.27] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. of World Congress on Nature & Biologically Inspired Computing (NaBic)*, pp. 210-214, 2009.
- [2.28] P. Civicioglu and E. Besdok, "A conceptual comarison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial Intelligence Review*, Springer, pp. 315-346, 2013.
- [2.29] M. Sidorov, E. Semenkin and W. Minker, "Unconstrained global optimization : A benchmark comparison of population-based algorithms," in *Proc. of 12th International Conference on Informatics in Control, Automation and Robotics*, pp. 230-237, 2015.
- [2.30] M. N. Ab Wahab, S. N.-Meziani and A. Atyabi, "A comprehensive review of swarm optimization algorithms," *PLoS ONE* 10(5), pp. 1-36, doi:10.1371/journal.pone.0122827, 2015.

第三章 本論文に関わる諸研究

3 本論文に関わる諸研究

本章では、本論文において取り組む問題の意義、重要性、提案する種々の方法の位置付けを明らかにするために、本論文と関連のある諸研究を分類し、その説明を行う。そして、本論文において著者が取り組む問題の意義・重要性と提案する種々の方法の位置付けを明らかにする。

本論文では、最適化法の性能を示す用語として (1) 多様性 (Diversity), (2) 解の探索性能, (3) 計算量, (4) 計算時間, を主として用いている。いずれの最適化手法においても収束状態を迎えて解の探索が終了する。

PSO に対する「多様性」とは解収束に至る (全ての粒子が同じ位置に至る) までの時間 (イタレーション) の長さを意味する。解収束が早い手法は、多峰性問題に対しローカルミニマム (局所解への収束) の問題を抱えていることが多い。すなわち、最適化手法として解探索を長時間に亘って行うことで探索範囲を広げ、そのことによって最適解への収束可能性が増加するものと考えられ、多様性が最適化手法の 1 つの性能指標と考える。

「解の探索性能」とは、解の探索領域の広さを示す用語として用いる。当然、解探索範囲の広い手法ほど優れた手法である。「多様性」と「解の探索性能」は強い相関を持っているが、等価ではないことから、これらの用語を使い分けることとする。

「計算量」は最適化手法を計算するにあたっての全ての計算量と考える。1 台の PC (Personal computer) 等に手法を実装し、種々の手法の計算時間の比較を行う場合は「計算量」の比較を行っていることと等価である。一方、「計算時間」とは解探索開始から収束するまでの計算時間を意味する。仮に「計算量」が多い手法であっても FPGA 等に並列実装可能であれば「計算時間」を短くすることが可能である。

高性能な手法は一般的に「計算量」が多くなる傾向があるが、手法自体に並列性が保証されていれば、ハードウェア実装を行うことで「計算時間」は短くすることが可能となる。本論文では「計算時間」を短くできる (並列実装可能な) 手法が優れた手法であるという考え方をとっている。

3.1 粒子群最適化法

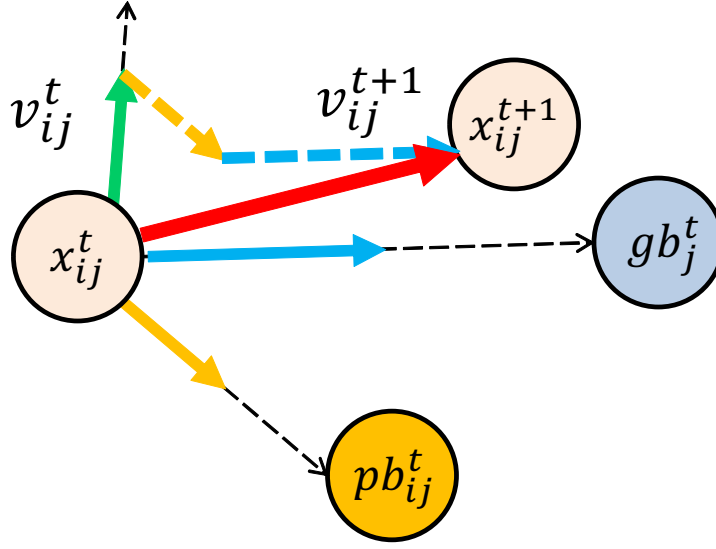


図 3.1: PSO の粒子モデル

粒子群最適化法 (Particle swarm optimization; PSO) [3.1] は母集団型の確率的なメタヒューリスティクスの一手法であり, 1995 年に J. Kennedy と R. C. Eberhart らによって提案された. PSO は鳥や魚などの群れを成して移動する生物の行動を模倣したアルゴリズムである. これらの群れを成す生物を粒子 (Particle) としてモデル化し, 粒子は最適化問題における解候補を表す. 粒子群を多次元解空間に配置し, 各粒子が他粒子との相互作用により探索を行うことで問題に対する近似解を求めることができる. 粒子群における i 番目の粒子 (粒子 i) は, 速度ベクトル $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, 位置ベクトル $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, 自身の探索過程における最良の位置ベクトル (Personal best solution; $pbest$) $\mathbf{pb}_i = (pb_{i1}, pb_{i2}, \dots, pb_{iD})$ を持つ. また, 粒子は粒子群の探索過程における最良の位置ベクトル (Global best solution; $gbest$) $\mathbf{gb} = (gb_1, gb_2, \dots, gb_D)$ を共有する. D は最適化問題の次元数を表す.

t ステップ目の粒子 i における j 次元目の速度と位置の更新式 [3.2] を式 (3.1), (3.2) に示す. また, 粒子モデルを図 3.1 に示す.

$$v_{ij}^{t+1} = wv_{ij}^t + c_1r_1(pb_{ij}^t - x_{ij}^t) + c_2r_2(gb_j^t - x_{ij}^t) \quad (3.1)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (3.2)$$

ここで, w は慣性速度定数, c_1, c_2 は $pbest, gbest$ それぞれに対する加速度係数 (Cognitive coefficient; c_1 , Social coefficient; c_2), r_1, r_2 は $[0, 1]$ の一様乱数を表す. 粒子は図 3.1 および更新式に示すように自身の速度と $pbest, gbest$ の影響を受けて探索を行い, 最終的に全ての粒子が $gbest$ に到達すると粒子は停止するため, 探索が終了する.

Algorithm 3.1 PSO の擬似コード

```

1:  $t = 0$ ;
2: 全ての粒子の  $\mathbf{v}^0$  と  $\mathbf{x}^0$  を初期化;
3: 全ての粒子の評価値を  $f(\mathbf{x}^0)$  により計算;
4: 全ての粒子の  $\mathbf{pb}^0$  を自身の  $\mathbf{x}^0$  で初期化;
5:  $\mathbf{gb}^0$  を最良の  $\mathbf{pb}^0$  で初期化;
6: while  $t \leq t_{max}$  do
7:   for  $i = 1$  to  $N$  do
8:     for  $j = 1$  to  $D$  do
9:       式 (3.1), (3.2) より, 粒子  $i$  の  $j$  次元要素の速度と位置を更新;
10:    end for
11:    粒子  $i$  の評価値を計算;
12:    if  $f(\mathbf{x}_i^{t+1}) < f(\mathbf{pb}_i^t)$  then
13:       $\mathbf{pb}_i^{t+1} = \mathbf{x}_i^{t+1}$ ;
14:    end if
15:  end for
16:   $k = \arg \min_i f(\mathbf{pb}_i^{t+1})$ ;
17:  if  $f(\mathbf{pb}_k^{t+1}) < f(\mathbf{gb}^t)$  then
18:     $\mathbf{gb}^{t+1} = \mathbf{pb}_k^{t+1}$ ;
19:  end if
20:   $t = t + 1$ ;
21: end while

```

Algorithm 3.1 に最小化問題の目的関数 $f(\mathbf{x})$ における PSO の擬似コードを示す. ここで, t は現在のイタレーション数, t_{max} は最大イタレーション数, N は総粒子数を表す.

PSO は他のメタヒューリスティクスと比較して次の利点を有する.

- 探索アルゴリズムが簡素であり, 最適化問題に適用することが容易である
- 解探索性能を制御するパラメータ数が比較的少ない
- 最良解への収束が早く, 近似解を素早く発見することができる
- 粒子の計算独立性が高く, 並列計算環境への実装が容易である

このため, PSO はニューラルネットワークの学習や電気回路の電力制御の最適化など様々なアプリケーションに適用されている [3.3]-[3.7],[3.21]-[3.25]. しかし, 標準的な PSO は発見した解への収束性が高いため, 解の多様性が失われやすく, 粒子群が解空間の大域的な探索を十分に行う前に局所解に収束 (Premature convergence; 初期収束) することがある [3.12]. 初期収束は多峰性問題において起こることが多く, 近似解の質が低下する要因である.

3.2 PSO の従来研究

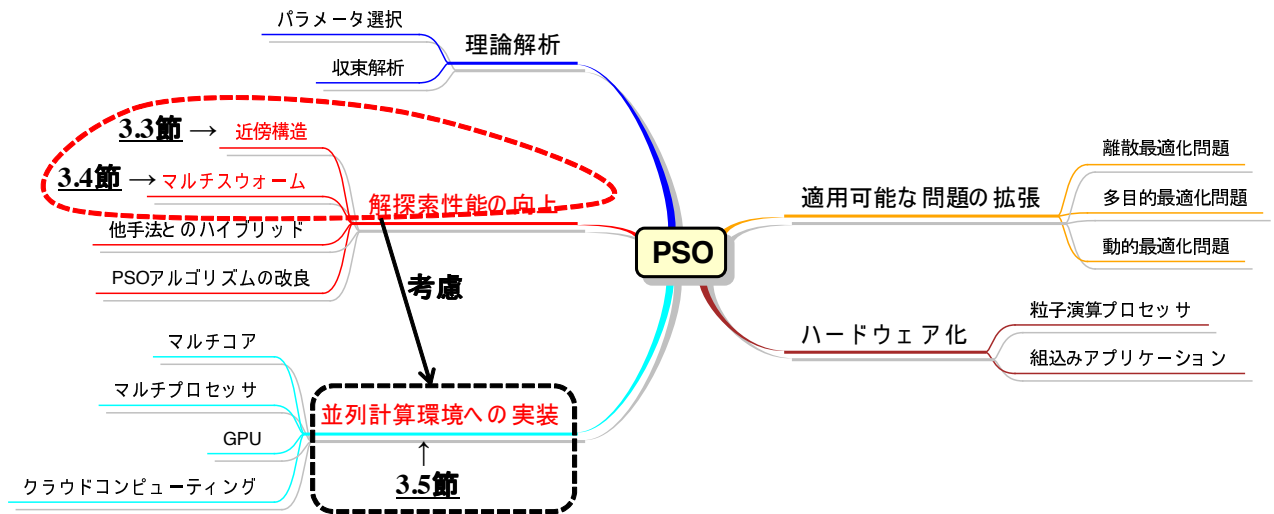


図 3.2: PSO の研究の分類

図 3.2 に PSO の従来研究の分類を示す。すなわち、PSO の研究は以下の 5 つに分類することができる。

- 解探索性能の向上
- 理論解析
- 並列計算環境への実装
- 適用可能な問題の拡張
- ハードウェア化

以下では、それぞれについて説明を加える。

3.2.1 解探索性能の向上に関する研究

ここに分類される研究は既存の PSO の解探索性能を向上させ、様々な最適化問題に対して高精度な近似解を求めることを目的としている研究である。PSO の欠点である初期収束性の改善に関する研究やより高精度な近似解の探索に関する研究などがあり、これらの従来手法は標準的な PSO よりも高い解探索性能を有している。

PSO の粒子間の近傍構造 (*i.e.*, 粒子間の情報共有の構造) にグラフ理論 (Graph theory) などのネットワーク構造を導入する手法や粒子群の母集団を複数の小規模なサブ集団に分割し、解空間を複数

のサブ集団で並列的に探索するマルチスウォーム (Multi-swarm) 手法がある。これらの研究では、探索初期における粒子群の多様性の低下を抑制することで初期収束を回避することができ、より効率的に高精度な近似解を発見することができる。また、標準的な PSO のように単一の集団により探索を行う PSO をシングルスウォーム PSO (Single-swarm PSO) と呼ぶ。これらの研究に関しては、本論文と関係が深い関連研究であるため、それぞれ、「3.3 粒子間に近傍構造を有する PSO の研究」と「3.4 マルチスウォーム PSO の研究」で詳細に説明する。

これら以外にも、他のメタヒューリスティクス (*e.g.*, 遺伝的アルゴリズムなど) の良い性質を PSO に組み込む手法や PSO のアルゴリズムに新しい処理の導入や粒子更新式を改良する手法がある。前者において、PSO と GA を組み合わせた PSO-GA [3.8], ACO と組み合わせた HAP [3.9], SA と組み合わせた PSO-SA [3.10], DE と組み合わせた HPSO-DE [3.11] などが明らかにされている。後者において、量子力学の概念を粒子に導入した Quantum-behaved PSO (QPSO) [3.13], 粒子の位置を標準正規分布により決定する Bare-bones PSO (BBPSO) [3.14] やファジー理論を導入した Fuzzy PSO (F-PSO) [3.15] などがある。

3.2.2 理論解析に関する研究

PSO の粒子の動作を理論解析し、PSO の制御パラメータと粒子軌道の収束性を明らかにすることを目的としている研究が属する。

式 (3.1) より、粒子のダイナミクスは w , c_1 , c_2 により決定され、これらのパラメータの設定により粒子は収束、発散のいずれかの軌道をとる。パラメータの設定と粒子の軌道の関係についての解析 [3.16]-[3.20] が行われている。これらの解析では、粒子の確率要素を 1 (*i.e.*, $r_1 = r_2 = 1$) とした決定論的な PSO (Deterministic PSO; D-PSO) を用いている [3.19],[3.20]。D-PSO における粒子 i における j 次元要素の速度と位置の更新式は次式となる。

$$c = c_1 + c_2 \quad (3.3)$$

$$q_{ij}^t = \frac{c_1 pbest_{ij}^t + c_2 gbest_j^t}{c_1 + c_2} \quad (3.4)$$

$$y_{ij}^t = x_{ij}^t - q_{ij}^t \quad (3.5)$$

$$v_{ij}^{t+1} = wv_{ij}^t + (c_1 + c_2)(q_{ij}^t - x_{ij}^t) \quad (3.6)$$

ここで、 q_{ij} は平衡点、 y_{ij} は平衡点を基準とした粒子の相対位置を表す。式 (3.2), (3.6) より、速度と位置情報をベクトル表記すると

$$\begin{bmatrix} v_{ij}^{t+1} \\ y_{ij}^{t+1} \end{bmatrix} = \begin{bmatrix} w & -c \\ w & 1 - c \end{bmatrix} \begin{bmatrix} v_{ij}^t \\ y_{ij}^t \end{bmatrix} \quad (3.7)$$

となる。ここで、粒子の各次元要素は互いに影響を与えないため、1次元の場合においても一般性

を失わない。また、解析を簡素にするために q_{ij} を理想的な固定点とし、探索過程において変化しないものとする。式 (3.7) より、粒子軌道は A の固有値により決定される。

$$A = \begin{bmatrix} w & -c \\ w & 1-c \end{bmatrix}$$

そして、 A の固有値 λ は次のように表せる。

$$\lambda = \frac{(1+w-c) \pm \sqrt{(1+w-c)^2 - 4w}}{2} \quad (3.8)$$

D-PSO が安定して平衡点に収束するためには、固有値 λ が複素平面上の単位円内に存在する必要がある。したがって、 λ が実固有値における粒子の収束条件は式 (3.9)、 λ が複素固有値における粒子の収束条件は式 (3.10) となる。

$$\begin{cases} |w| < 1 \\ 0 < c < 2w+2 \end{cases} \quad (3.9)$$

$$0 < w < 1 \quad (3.10)$$

粒子軌道の収束条件より、 $q_{ij} = 0$ におけるパラメータによる粒子の収束、発散を図 3.3 に示す。図 3.3 において、赤い領域は実固有値における収束領域、緑の領域は複素固有値における収束領域を表す。各青い印は粒子軌道の遷移を表す図のパラメータを表す。まず、収束領域内において複素固有値の粒子軌道は 0 に減衰振動し、実固有値の粒子軌道は 0 に単調減少する、収束領域外において粒子軌道は発散する。

文献 [3.17] においては、M. Clerc と J. Kennedy らによって理論解析により粒子の速度が収束することを保証する“constriction factor”が提案されている。Clerc らは式 (3.7) の速度と位置の更新式を 5 個の係数 α , β , γ , δ , η を用いて以下の式に示した。

$$\begin{bmatrix} v_{ij}^{t+1} \\ y_{ij}^{t+1} \end{bmatrix} = \begin{bmatrix} \alpha & -\beta\varphi \\ \gamma & \delta - \eta\varphi \end{bmatrix} \begin{bmatrix} v_{ij}^t \\ y_{ij}^t \end{bmatrix} \quad (3.11)$$

ここで、 $\varphi = c = c_1 + c_2$ ($\varphi > 0$) とする。式 (3.11) において、 $\alpha = \gamma = w$, $\beta = \delta = \eta = 1$ のときは式 (3.7) と等価となる。Clerc らはこれら 5 個のパラメータ間の関係性について分類し、これらのパラメータ選択により粒子の速度が収束、発散することを示している。

これらのパラメータ間の分類について *Class 1'* (i.e., $\alpha = \beta = \gamma = \eta$, $\delta = 1$) では、パラメータ α に“constriction coefficient (χ)”を設定することによって粒子が安定して平衡点に収束することが解析的に明らかにされている。*Class 1'* の粒子の速度更新式と“constriction coefficient”は次の式で与

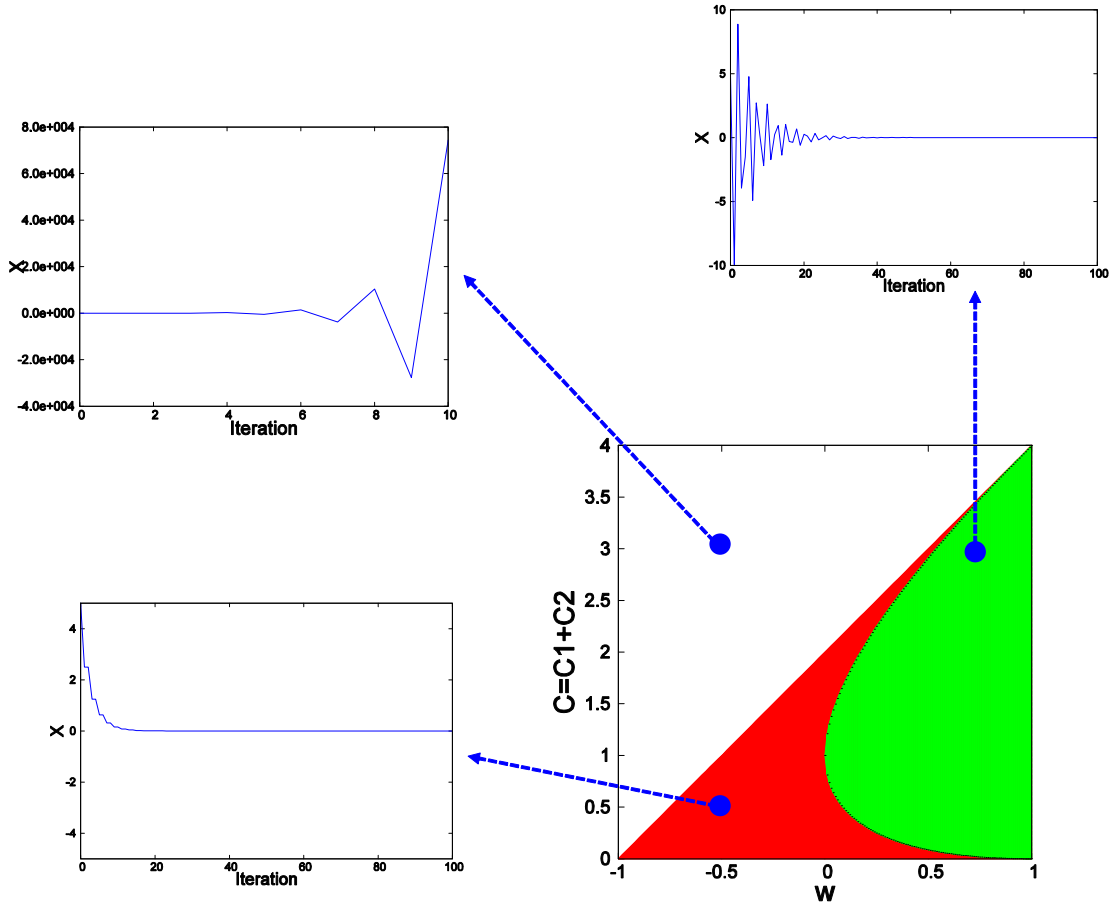


図 3.3: 粒子の収束, 発散の例

えられる。

$$v_{ij}^{t+1} = \chi \left(v_{ij}^t + \varphi_1 r_1 (pb_{ij}^t - x_{ij}^t) + \varphi_2 r_2 (gb_j^t - x_{ij}^t) \right) \quad (3.12)$$

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (3.13)$$

$$\varphi = \varphi_1 + \varphi_2 > 4 \quad (3.14)$$

3.2.3 並列計算環境への実装に関する研究

ここに分類される研究はマルチコア, PC クラスタ, GPU (Graphics processing unit), クラウドコンピューティングなどの並列計算環境を用いて PSO のアルゴリズムを並列化し, その計算量を分散させ, より短い計算時間で近似解を発見することを目的としている。

近年, 技術の発達に伴い最適化問題の解空間は広大かつ複雑になってきている [3.25]. 例えば, 文書ファイルの分類 [3.23] やポートフォリオ (Portfolio) の最適化 [3.24], 計算流体力学 (Computational fluid dynamics; CFD) のシミュレーションなどでは取り扱うデータ数の増加や計算の複雑度が増し膨大な計算量を必要とする. PSO においても, このような大規模な最適化問題に対して高精度な近似

解を求めるためにはより多くの粒子数が必要となるため膨大な計算時間を要することがある。

PSO では、粒子ごとに計算が独立しているため、並列計算環境への実装は容易であり、多数の粒子を並列的に計算することで計算時間の低減ができる。

本論文では、計算時間の短縮を考慮し、PSO の解探索性能の向上手法を提案するため、これらの研究は本論文との関係が深い。そこで、「3.5 PSO アルゴリズムの並列計算環境への実装の研究」に PSO を並列計算環境へ実装する研究の説明を行う。

3.2.4 適用可能な問題の拡張に関する研究

動的最適化問題、離散最適化問題、多目的最適化問題などに対して PSO が適用可能にすることを目的とする研究群である。標準的な PSO は設計変数が連続値をとり、静的な解空間である最適化問題を解くために設計されている。上記のような最適化問題を解くために、PSO アルゴリズムの改良が行われてきている。

離散最適化問題を解くための離散 PSO (Discrete particle swarm optimization; DPSO) [3.28] は無線センサネットワークにおけるフラッディングの効率化 [3.29] や Zigbee センサネットワークのルータ配置 [3.30] などに適用されている。

多目的最適化問題を解くための多目的 PSO (Multi-objective PSO; MOPSO) [3.31] は超音速飛行機 (HTV-2) の空力最適化設計 [3.32] やポートフォリオの最適化 [3.24] などに適用されている。

3.2.5 ハードウェア化に関する研究

PSO アルゴリズムを組み込みアプリケーションに実装し、最適化を行うことや粒子演算を高速に行うための専用プロセッサの開発など PSO アルゴリズムをハードウェア上に実装することを目的としている研究が分類される。アンテナの向きの調整や電気回路の電力制御、オンチップ NN の学習などの組み込みアプリケーションの最適化 [3.4]-[3.7] を目的とする研究では、PSO アルゴリズムのハードウェア化が成されている。

粒子演算に特化した専用プロセッサの開発 [3.33]-[3.39] も行われている。これらの手法の多くは標準的な PSO アルゴリズムを回路化し、FPGA (Field-programmable gate array) へ実装することで、PSO アルゴリズムの計算時間の低減を実現している。このような粒子演算の専用プロセッサを多数用いて並列計算させることで 3.2.3 で示した研究と同様に計算時間を低減できる。

本論文では、3.2.1 で示した PSO アルゴリズムの解探索性能を向上させる手法の研究を 3.2.3 の並列計算環境への実装を考慮して行うものである。また、3.2.2 に属する粒子軌道の安定性についても議論する。

PSO の解探索性能を向上させる手法は多数提案されているが、これらの多くは並列計算環境への実装を考慮しておらず、並列計算環境への PSO アルゴリズムの実装に関する研究も標準的な PSO を対象とした手法が多い。このため、PSO アルゴリズムの解探索性能を向上させ、かつ計算時間の短縮を考慮するアルゴリズムの研究は大規模な最適化問題を効率的に解くために意味深い。

本論文における PSO アルゴリズムの解探索性能は“多様性を向上させる”ことによって成し遂げられることを考えていて、マルチスウォームに関する手法と粒子間の近傍構造に関する手法に着目した。多数の計算機を並列的に用いるためにはマルチスウォーム手法は適しており、計算機同士の情報共有に近傍構造に関する手法を導入することで多様性の向上を図ることができる。これらの近傍構造の手法は複雑な処理を必要としないものが多い。

次節以降は「粒子間に近傍構造を有する PSO の研究」、「マルチスウォーム PSO (Multi-swarm PSO; MS-PSO) の研究」、「PSO アルゴリズムの並列計算環境への実装の研究」について詳細な説明を行う。

3.3 粒子間に近傍構造を有する PSO の研究

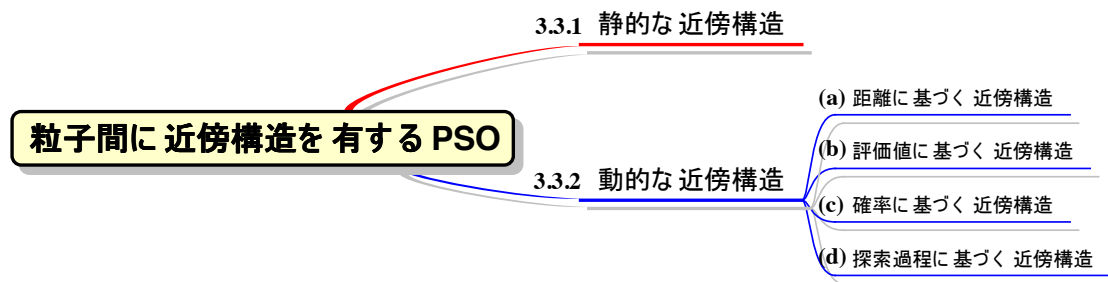


図 3.4: 粒子間に近傍構造を有する PSO の研究の分類

粒子間の近傍構造に関する手法は PSO の多様性を制御するための手段の 1 つである。PSO の近傍構造は粒子が他粒子に与える影響範囲を決定する。PSO において、粒子間に近傍構造を有する PSO の研究は図 3.4 に示すように「静的な近傍構造の研究」と「動的な近傍構造の研究」に大別される。ここでは、まず、「静的な近傍構造の研究」について説明する。

3.3.1 静的な近傍構造の研究

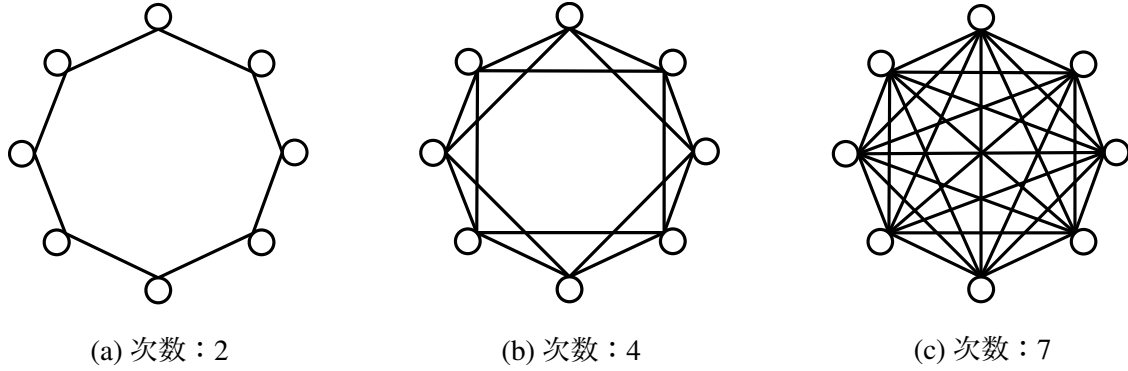


図 3.5: PSO における粒子間の近傍構造例 (総粒子数：8)

静的な近傍構造においては、近傍粒子は解空間上の位置関係によって決定されるのではなく、粒子番号 (Index) によって決められることが多い。標準的な PSO では、各粒子は他の全粒子と最良解情報 (i.e., *gbest*) を共有している全結合型の近傍構造 (Fully connected topology または *gbest topology*) を持つ。このため、標準的な PSO は *gbest* に収束しやすく、多峰性問題において初期収束を起こしやすい。

各粒子が特定の近傍粒子間のみで最良解情報 (Local best solution; *lbest*) を共有する “*lbest topology*” が提案されている [3.40]。 “*lbest topology*” では、各粒子の近傍粒子数は次数 (Degree) により静的に決定される。次数が小さい場合は、粒子は情報共有可能な近傍粒子数が少なく、その粒子は群れから独立して探索を行うようになる。また、粒子群における最良解情報を保持する粒子の情報は全ての粒子に伝搬するまでに時間がかかる。したがって、 “*lbest topology*” における次数が小さい場合は “*gbest topology*” よりも多様性を向上させ、多峰性問題においても局所解に陥らずに良好な近似解を探索する可能性が向上する。ただし、情報伝播に遅延が発生するため、発見した最良解への収束に時間がかかる。 “*lbest topology*” における粒子 i の速度更新式は以下で与えられる。

$$v_{ij}^{t+1} = wv_{ij}^t + c_1r_1(pb_{ij}^t - x_{ij}^t) + c_2r_2(nb_{ij}^t - x_{ij}^t) \quad (3.15)$$

ここで、 $\mathbf{nb}_i = (nb_{i1}, nb_{i2}, \dots, nb_{iD})$ は粒子 i における近傍粒子間の最良解情報 (Neighborhood best; *nbest*) を表す。したがって、粒子は *nbest* 周辺の探索を行うことになり、粒子が *gbest* へ一斉に収束することを抑制できる。

図 3.5 に総粒子数が 8 個のときの次数が 2, 4, 7 の “*lbest topology*” の例を示す。図 3.5 において、“○” は粒子を表し、粒子間を結合する “エッジ” は良解情報を共有する関係を表す。

図 3.5 (a) は次数が 2 であり、各粒子は隣接する 2 近傍の粒子とのみ情報共有を行う。次数が 2 のときの近傍構造は円環状態となり、“ring topology” とよぶ。 “ring topology” では、最良解情報の伝播は遅く、粒子は様々な局所解を並列的に探索できるため、多峰性問題に有効である。この性質

表 3.1: 総粒子数 9 個における平均頂点間距離

次数	平均頂点間距離 (L)
2	2.500
4	1.500
6	1.250
8	1.000

は“Niching algorithm” [3.89] と類似している。“Niching algorithm”では、各粒子は母集団から分化し、サブ集団の形成や他サブ集団への統合を繰り返すことで解空間上の局所解を並列的に探索することができる。文献 [3.44] では、“ring topology”にメモリ機構を導入した PSO が提案されており、“Niching algorithm”よりも解探索性能が優れていることが示されている。

図 3.5 (b) は次数が 4 であり、各粒子は隣接する 4 近傍の粒子と良解情報の共有を行う。

図 3.5 (c) は次数が 7 であり、“gbest topology”と等価であり、各粒子は他の全ての粒子と良解情報の共有を行う。この場合は最良解情報の伝播が速く、粒子群は局所的な探索を行いやすい。このように、次数が大きくなるに従い、“lbest topology”から“gbest topology”へと変化し、粒子は *gbest* に収束する傾向が強くなり、多様性は徐々に減衰していく。

近傍構造の特徴を表す指標として次数以外に平均頂点間距離がある。平均頂点間距離は粒子と他粒子までの平均的な距離を表し、粒子群における最良解情報の伝播のしやすさを示す。総粒子数が N における平均頂点間距離 (L) は次式で算出される。

$$L = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{j=1, j \neq i}^N d(i, j) \quad (3.16)$$

i, j は粒子番号、 $d(i, j)$ は i 番目の粒子と j 番目の粒子間の最短距離を表す。平均頂点間距離が 1 に近づくにつれて粒子間の情報共有は密となり 1 から大きくなるにつれて粒子間の情報共有は疎となる。このため、平均頂点間距離によって最良解情報の伝搬速度を評価することができる。平均頂点間距離が大きくなるにつれて粒子の探索独立性が向上し多様性が高くなり、平均頂点間距離が 1 に近づくにつれて *gbest* への収束性が強くなり多様性は低くなる。

表 3.1 に総粒子数 9 において、次数 2, 4, 6, 8 としたときの“lbest topology”における平均頂点間距離を示す。表 3.1 より、“lbest topology”において、次数が増加するに従い、平均頂点間距離は 1 に近づいていくことがわかる。特に、次数が 8 の“gbest topology”においては平均頂点間距離は 1 であり、粒子は他の全ての粒子と結合していることを表している。一方、次数が 2 の“ring topology”においては平均頂点間距離が最も大きいことから最良解情報の伝搬速度も他の次数と比較して遅くなる。

粒子間の静的な近傍構造においては“gbest topology”, “lbest topology”とその中間的構造が代表

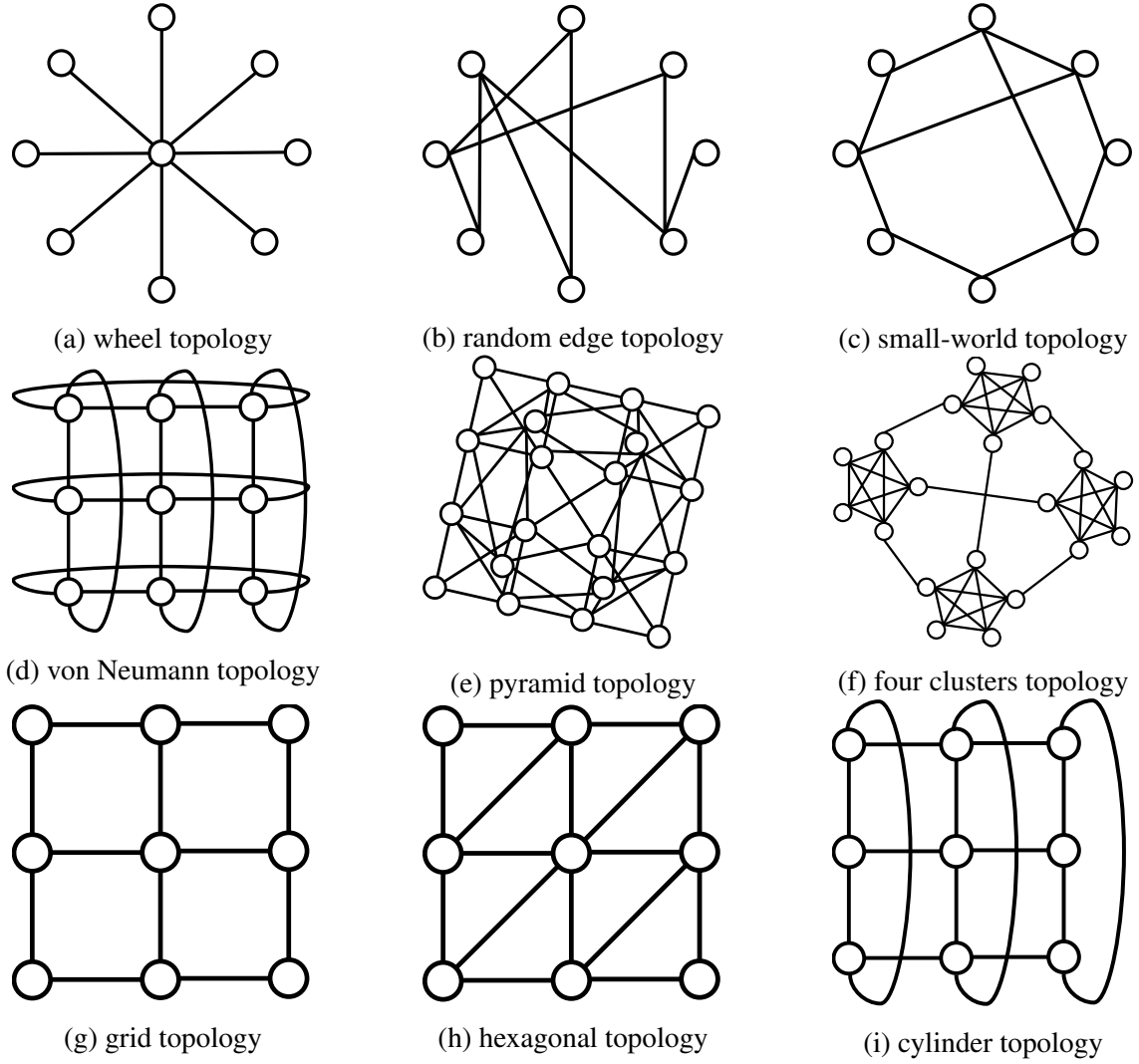


図 3.6: PSO における代表的な静的な近傍構造

的であるが、他にも様々な近傍構造に基づく手法 [3.40]-[3.50] が提案されている。図 3.6 に代表的な静的な近傍構造を示し、それぞれについて以下に説明を加える。

(a) wheel topology

図 3.6 (a) のように、1つの中心粒子と他の全ての粒子と情報共有しており、中心粒子以外の粒子は他の粒子と情報共有しない近傍構造である。

(b) random edge topology

図 3.6 (b) のように、各粒子が他の粒子と情報共有するかどうかを確率的に決定する近傍構造である。

(c) small-world topology

図 3.6 (c) のように、“ring topology” を基に各粒子が2近傍以外の粒子と情報共有するかどうかを確率的に決定する近傍構造である。

(d) von Neumann topology

図 3.6 (d) のように、2 次元正方格子状に各粒子が他粒子と結合している近傍構造である。

(e) pyramid topology

図 3.6 (e) のように、3 次元三角格子状に各粒子が他粒子と結合している近傍構造である。

(f) four clusters topology

図 3.6 (f) のように、4 つのクラスタ内の粒子が他クラスタ内の粒子と結合している構造である。各クラスタ内の近傍構造は “gbest topology” と等価である。

(g) grid topology

図 3.6 (g) のように、2 次元正方格子状に各粒子が他粒子と結合している近傍構造である。“von Neumann topology” と類似の近傍構造であるが、ループ構造になっていない。

(h) hexagonal topology

図 3.6 (h) のように、2 次元三角格子状に各粒子が他粒子と結合している近傍構造である。

(i) cylinder topology

図 3.6 (i) のように、“grid topology” の上下をループ状に結合した近傍構造である。

文献 [3.41] では、様々な静的な近傍構造を持つ PSO の解探索性能の比較が行われており、“von Neumann topology” や “pyramid topology” が優れていることが示されている。ただし、これらの近傍構造は最適化問題の景観に依存するため、問題に応じて適切な近傍構造を選択する必要がある [3.40]。

静的な近傍構造を有する PSO の研究において、粒子の速度更新式は式 (3.15) に基づいて行われる。つまり、各粒子は各々の近傍粒子の中から最良解情報を有する粒子の情報のみを参照する。しかし、この最良解情報を有する粒子が他の近傍粒子よりも良好な解空間を探索できる保証はない。その弊害を回避する手法として FIPS (Fully informed particle swarm) [3.42],[3.43] が明らかにされていて、その手法では、各粒子は近傍粒子の最良解情報のみを参照せずに近傍粒子全ての $pbest$ の情報に基づいて自身の速度情報を更新するものとなっている。

$$v_{ij}^{t+1} = \chi(v_{ij}^t + \varphi(pb_{mj}^t - x_{ij}^t)) \quad (3.17)$$

$$\varphi = \sum_{k \in |N|} \varphi_k \quad (3.18)$$

$$\varphi_k = \left[0, \frac{\varphi_{max}}{|N|} \right] \quad (3.19)$$

$$pb_{mj}^t = \frac{\sum_{k \in |N|} W(k) \varphi_{kj} pb_{kj}^t}{\sum_{k \in |N|} W(k) \varphi_k} \quad (3.20)$$

FIPS の粒子 i における j 次元要素の速度更新式は式 (3.17)-(3.20) で示される。ここで、 φ_k は加速度係数、 $|N|$ は粒子 i の近傍粒子の集合、 $W(k)$ は近傍粒子 k に対する重み係数である。 χ は式 (3.13) で与えられる。 \mathbf{pb}_m は近傍粒子における平衡点を表し、粒子は平衡点の方向へ探索を進める。文献 [3.42],[3.49] では、FIPS の有効性が明らかにされている。

文献 [3.49] では、近傍構造における平均頂点間距離と解情報の伝播速度についても解析されており、平均頂点間距離が最小の “gbest topology” は平均頂点間距離が最大の “ring topology” と比較して解情報伝播速度が速いことが示されている。標準的な PSO においては粒子は最良解情報を参照しているため、解情報の伝搬速度が FIPS と比較して速く、多様性も減少しやすい。ただし、解伝搬速度が遅い方が必ずしも解探索性能が優れているわけではない。

図 3.6 (a)~(i) の静的トポロジーにおける研究では、母集団が 1 つであり、総粒子数も 20 ~ 50 個程度の比較的小さな規模のネットワークを取り扱っていて、小規模なサブ集団が複数あるマルチスウォーム手法にこれらの近傍構造を適用した手法は少ない。

3.3.2 動的な近傍構造の研究

動的な近傍構造の種々の研究 [3.51]-[3.69] について説明する。静的な近傍構造においては、粒子間の近傍構造はその探索過程において変化することはないため、十分に情報が使用されていないことが示唆されている [3.55]。

一方、動的な近傍構造では、探索の状況に応じて各粒子の近傍粒子数が変化する近傍構造であり、探索の状況に応じて近傍粒子を変化させるため、静的な近傍構造よりも多様な探索が行える。そして、動的な近傍構造に関する 4 つの研究は (a) 距離に基づく近傍構造の研究、(b) 評価値に基づく近傍構造の研究、(c) 確率に基づく近傍構造の研究、(d) 探索過程に基づく近傍構造の研究、に分類される。

(a) 距離に基づく近傍構造の研究

距離に基づく近傍構造の研究 [3.51]-[3.56] について説明する。この分類に属する研究では、粒子の近傍構造は解空間上および目的関数の評価値の距離に応じて変化する。

PSO の初期収束を回避するために提案された動的な近傍構造 (Neighborhood operator; NOPSO) [3.51] は PSO の動的な近傍構造において初期に提案された手法の 1 つである。NOPSO では、初期の粒子間の近傍構造を “ring topology” とし、各粒子の探索の独立性が向上されるため、探索初期における粒子群の多様性は高く、大域的な解空間の探索ができる。各粒子は他粒子との距離情報および探索が進むにつれて近傍粒子を徐々に増加させ、最終的に近傍構造は “gbest topology” となる。このため、探索の終盤においては探索してきた最良解周辺の局所探索を行うことができる。NOPSO

の他にも距離情報を用いて近傍粒子が動的に変化する手法が提案されていて、以下では、その代表的なものを説明する。

FDR-PSO (Fitness-distance-ratio PSO) [3.52],[3.53] は他粒子の評価値とその距離に基づいて参照する近傍粒子が変化する手法である。式 (3.21), (3.22) に粒子 i における j 次元要素における速度更新式を示す。

$$n = \arg \max_k \frac{f(\mathbf{pb}_k^t) - f(\mathbf{x}_i^t)}{|pb_{kj}^t - x_{ij}^t|} \quad (3.21)$$

$$v_{ij}^{t+1} = wv_{ij}^t + c_1r_1(pb_{ij}^t - x_{ij}^t) + c_2r_2(gb_j^t - x_{ij}^t) + c_3r_3(pb_{nj}^t - x_{ij}^t) \quad (3.22)$$

ここで、 n は粒子 i における j 次元要素における最良の近傍粒子番号を表し、 pb_{nj} は近傍粒子間の最良解情報 ($nbest$) の j 次元要素を表す。各粒子は $pbest$, $gbest$ に加えて $nbest$ も考慮して解空間の探索を行い、各次元要素の参照する $nbest$ が変化する。 $nbest$ の評価値が良好であり、その次元要素と粒子位置の距離が近いほどその次元における近傍粒子として選ばれやすい。

FER-PSO [3.54] は FDR-PSO の改良手法として提案されており、多峰性問題の多数の局所解を並列的に探索することができる。

さらに、“ring topology” に基づいた PSO-Dring [3.55] や各粒子が近傍粒子とみなすための範囲を有する PSO [3.56] やランダムに近傍粒子を決定する PSO [3.56] などが提案されている。

(b) 評価値に基づく近傍構造の研究

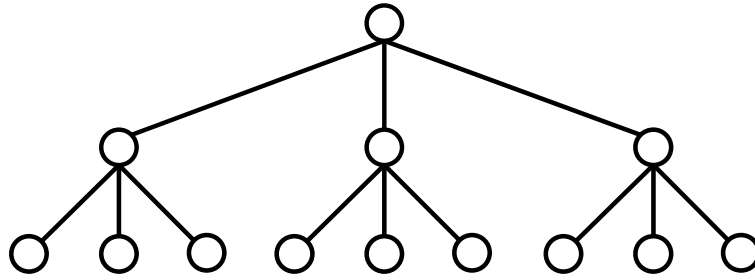


図 3.7: H-PSO における近傍構造の例

評価値に基づく動的な近傍構造の研究 [3.47],[3.57]-[3.59] について説明する。この分類に属する研究では、評価値に応じて近傍粒子の生成やトポロジー内の近傍粒子の交換などを行い近傍構造が変化する。

H-PSO (Hierarchical PSO) [3.57] や TGPSO (Growing tree topology) [3.59] は粒子間の近傍構造に木構造を導入した手法であり、動的に近傍構造の入れ替えや近傍粒子の追加を行う手法である。

H-PSO では、根 (Root) に近い粒子ほど評価値が良くなり、葉 (Leaf) に近い粒子ほど評価値が悪

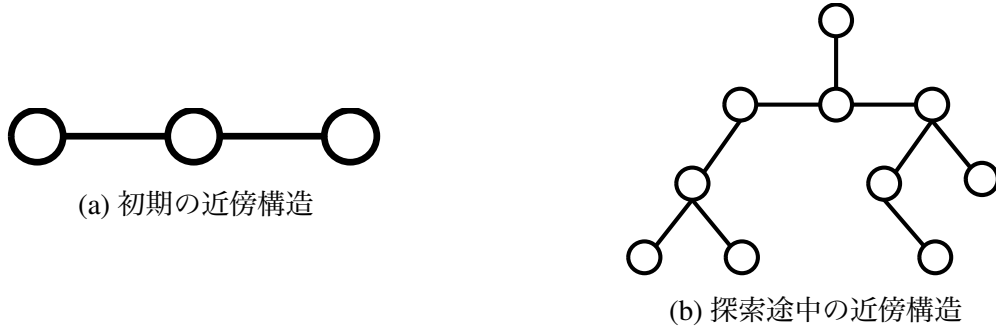


図 3.8: TGPSO における近傍構造の例

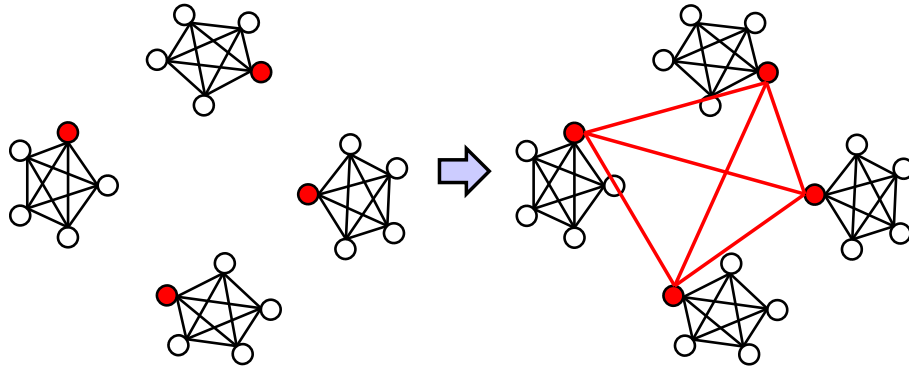


図 3.9: C-PSO における近傍構造の例

くなるように近傍構造を構築する。図 3.7 に H-PSO における近傍構造の例を示す。下位層の粒子 j (子ノード) が上位層の粒子 i (親ノード) より良好な解を発見した場合、粒子 j と粒子 i を入れ替えることで近傍構造の入れ替えを行う。各サブ木は独自に解空間を探索することができる。

TGPSO では、粒子 i の $pbest_i$ が一定期間改善がされなかった場合、解空間上にランダムに生成した粒子 j を粒子 i の近傍として追加する。図 3.8 に示すように、初期の近傍構造は 1 次元格子状であるが (図 3.8 (a)), 粒子の探索が停滞するごとに近傍粒子が増加していく (図 3.8 (b)). 局所解に陥った粒子は新しく生成された近傍粒子の情報を参照することにより、局所解からの脱出ができる。また、解空間に応じて柔軟に近傍構造を構築することができる。このように木構造を動的に構築する近傍構造は様々な多峰性問題に対して効率的に高精度な近似解を発見できることが示されている。

C-PSO (Clan PSO) [3.58] や NS-PSO (Network structured PSO) [3.47] は粒子の評価値に基づいて参照する近傍粒子が変化する手法である。

C-PSO では、“four clusters topology”と同様の近傍構造を有する手法である。図 3.9 に C-PSO の近傍構造の例を示す。同図において、赤い“o”は各サブ集団における最良解情報を有する粒子を表し、サブ集団間において各サブ集団の最良解情報を有する粒子同士が“gbest topology”型の近傍構造を構築する。文献 [3.58] では、各サブ集団の最良解情報を有する粒子間の近傍構造として、“ring topology”も提案されている。また、複数のサブ集団をサブ集団内の最良の粒子同士で情報共有することは、1 つの母集団における静的な“gbest topology”や“ring topology”や“four clusters topology”

よりも解探索性能が優れていることが明らかにされている。

NS-PSO では、イタレーション t における最良の $f(\mathbf{x}_c^t)$ を満たす粒子 c を勝者位置情報として選び、各粒子は近傍粒子の最良解および勝者位置情報のみを参照して粒子情報を更新する手法である。

(c) 確率に基づく近傍構造の研究

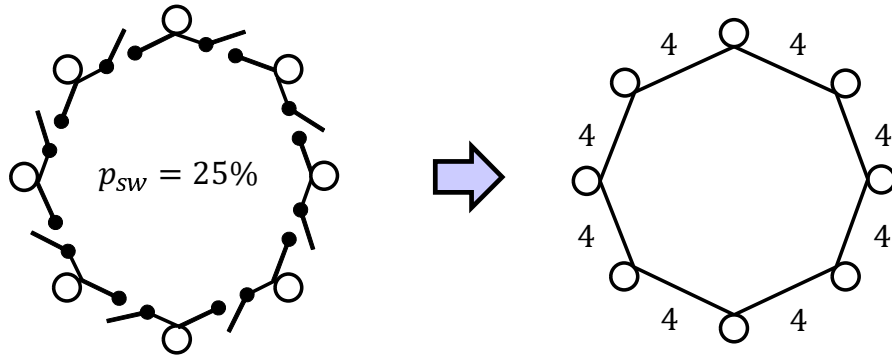


図 3.10: SW-TOPO における近傍構造の例 ($p_{sw} = 25\%$)

確率に基づく近傍構造の研究 [3.60]-[3.65] について説明する。この分類に属する研究では、粒子の近傍構造は確率的に変化し、その近傍粒子数も変化する。

文献 [3.60],[3.62]-[3.64] は確率的に近傍粒子が決定される手法であり、それぞれ、各粒子の近傍粒子数が確率的に決定される手法 [3.60]、スケールフリーなネットワークである Barabási-Albert (BA) モデルを用いて確率的に粒子間の近傍構造を構築する手法 [3.62]、 p_{best} が更新されないイタレーション数 (Age) に応じて変化する手法 [3.63]、各粒子がパラメータ R 個の近傍粒子をランダムに選択する手法 [3.64] 等をあげることができる。これらの手法の根本的な考え方は確率的に近傍粒子を決定することで解探索性能の向上を実現し、トポロジーに柔軟性を与えることで粒子の探索領域が広がることにある [3.64]。

SW-TOPO (switched topology) [3.65] では、図 3.10 に示すように各粒子は確率的に近傍粒子と情報を共有するかどうかを決定するため、粒子の情報伝播速度の制御が容易である。 p_{sw} は近傍粒子の情報を参照するための確率であり、粒子群の平均頂点間距離を制御するパラメータとなる。SW-TOPO における平均頂点間距離は p_{sw} を用いて

$$L_{sw} = \frac{1}{p_{sw}} \cdot L \quad (3.23)$$

と与えられる。ここで、 L_{sw} は SW-TOPO の平均頂点間距離、 L は “lbest topology” の平均頂点間距離 (式 (3.16)) を表す。 p_{sw} が 0 に近づくにつれて平均頂点間距離は長くなるため、多様性の向上および粒子の探索の独立性が促進される。

IPSO (Independent-minded PSO) [3.61] は各粒子は確率的に他粒子と情報共有するかどうかを決定する手法である。粒子 i は式 (3.24) に基づいて他粒子と情報共有するかどうかを決定する。

$$rand_i \leq C \quad (3.24)$$

ここで、 $rand_i$ は $[0, 1]$ の一様乱数を示しているから閾値パラメータ C で確率を可変とする。式 (3.24) を満たした粒子間において情報共有を行い、条件を満たさなかった粒子は情報共有を行わず、群れから独立して探索を行う。式 (3.25), (3.26) に IPSO における粒子 i の j 次元要素の速度更新式および $gbest$ の更新式を示す。

$$\begin{cases} v_{ij}^{t+1} = wv_{ij}^t + c_1 r_1 (pb_{ij}^t - x_{ij}^t) + c_2 r_2 (gb_j^t - x_{ij}^t) & rand_i \leq C \\ v_{ij}^{t+1} = wv_{ij}^t + c_1 r_1 (pb_{ij}^t - x_{ij}^t) & rand_i > C \end{cases} \quad (3.25)$$

$$gb^{t+1} = \arg \min_i f(\mathbf{pb}_i^{t+1}), \quad rand_i \leq C \quad (3.26)$$

したがって、粒子 i が最良値の情報を持っていたとしても、 $rand_i > C$ の場合はその最良値を $gbest$ の参照および更新に用いることはできない。 $C = 1$ の場合は標準的な PSO と等価となる。 $C = 0$ の場合、全ての粒子は $gbest$ の参照および更新を行わず、それぞれ独立して探索を行う。IPSO は近傍粒子の設定や近傍構造を逐次再構築するといった複雑な処理を必要しないが文献 [3.60],[3.64],[3.65] に多峰性問題に対して優れた解探索性能を有することが示されている。

(d) 探索過程に基づく近傍構造の研究

探索過程に基づく近傍構造の研究は文献 [3.66]-[3.69] で明らかにされている。この分類には、粒子の近傍構造がイタレーションや目的関数の評価回数に応じて変化する方法が属する。

3.4 マルチスウォーム PSO の研究

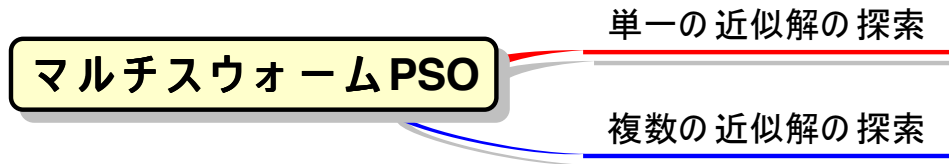


図 3.11: マルチスウォーム PSO の研究の分類

マルチスウォーム PSO (Multi-swarm PSO; MS-PSO) では、粒子群の母集団 (Population) を複数の小規模なサブ集団 (Sub-swarm) に分割し、各サブ集団は他サブ集団との相互作用によって解空間の

探索を行う。

MS-PSO の多くの手法において、母集団は S 個のサブ集団に分割され、 g 番目のサブ集団 (サブ集団 g) はその局所的な最良解情報 ($\mathbf{lb}_g = (lb_{g,1}, lb_{g,2}, \dots, lb_{g,D}); lb_{best}$) を持つ。MS-PSO は各サブ集団が独自に探索を行うため、標準的な PSO よりも多様性が高く、サブ集団同士の相互作用のための情報交換の方法は多様性制御において重要である。

マルチスウォーム PSO の研究は「単一の近似解を探索する研究」と「複数の近似解を探索する研究」に大別される。以下では、それぞれについて説明を加える。

3.4.1 単一の近似解を探索する研究

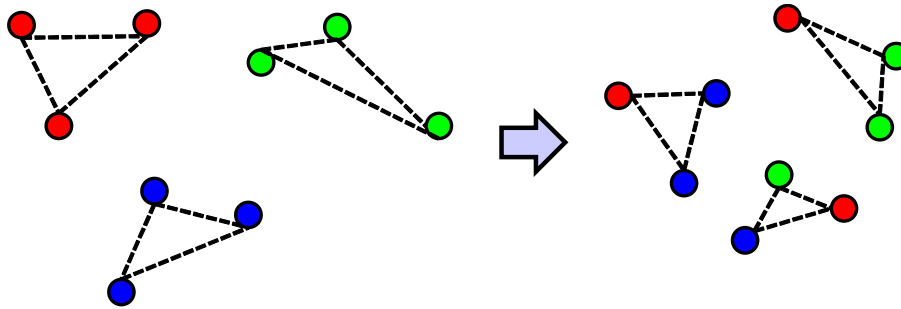


図 3.12: DMS-PSO における再グループ化の例

単一の近似解を探索する研究 [3.71]-[3.80] について説明する。粒子群の多様性を向上させることで初期収束を回避し、1つの高精度な近似解を発見することを目的としている研究が分類される。

DMS-PSO (Dynamic multi-swarm PSO) [3.72] は一定期間ごとに全てのサブ集団がランダムに再グループ化される機構を有する手法である。図 3.12 に再グループ化の概念図を示す。粒子は複数のサブ集団にランダムに割り当てられ、再グループ化期間 R (Regrouping period) ごとに全ての粒子はこれらのサブ集団にランダムに割り当てられる。この操作により各サブ集団の良解情報は再グループ化ごとにサブ集団間で交換されることで相互作用が行われる。探索の終盤においては全てのサブ集団が探索してきた中における最良解周辺の探索を行うために、全てのサブ集団は1つの母集団に統合され、標準的な PSO として探索を行う。DMS-PSO は多様性は高いものの、局所探索能力が弱いため、様々な改良手法が提案されている [3.73],[3.74]。また、フローショップスケジューリング問題 (Flowshop scheduling problem; FSP) や経路探索問題などの実アプリケーションへの適用もなされている [3.75],[3.76]。

DMS-PSO の他にも、各サブ集団から粒子を選び、メモリに保存し、このメモリからランダムに選択した親粒子を用いて子粒子を生成する交叉処理を導入した手法 [3.71] や最良解を持つサブ集団以外のサブ集団に属する粒子の位置情報と速度情報を定期的に初期化する手法 [3.77] などが提案されている。

PSO を用いて高次元な問題 (*i.e.*, 設計変数が多数) を解くために CPSO (Cooperative PSO) [3.78] が提案されている。CPSO における各サブ集団は D 次元の解空間の各次元に対して割り当てられる。つまり, D 個のサブ集団がそれぞれ 1 次元の解空間を探索し, 各サブ集団における良解情報を統合する。このため, CPSO は設計変数間に依存のない多峰性問題に対して有効であるが, 設計変数間に依存関係のある問題に対する解探索性能は高くない。この問題を解決するための CPSO 手法 [3.79] や 1000 ~ 2000 次元の問題を解くための CPSO 手法 [3.80] などが提案されている。

3.4.2 複数の近似解を探索する研究

複数の近似解を探索する研究 [3.81]-[3.97] について説明する。ここに分類される研究は, 複数のサブ集団が並列的に解空間を探索する性質を利用することで複数の良好な近似解を同時に求めることを目的としている。

複数の近似解を並列的に探索する代表的な手法として “Niching” 手法 [3.70] があり, “Niching” を PSO に導入した NichePSO [3.89] では, 粒子は母集団から分化しサブ集団を形成する。各粒子は他のサブ集団との距離に基づいて他のサブ集団への移住する。サブ集団同士の探索範囲が競合した場合はそれらのサブ集団を 1 つに統合し, 新しいサブ集団の生成を行う。このように, NichePSO はサブ集団の生成や統合を繰り返すことで複数解の探索を行う手法である。

NichePSO の解探索性能を向上させるための改良手法 [3.90]-[3.94] や実アプリケーションへの適用手法 [3.95]-[3.97] などが様々に提案されている。

NichePSO の他にも, 母集団からの分化, サブ集団間の粒子移住やサブ集団生成を伴う手法に DMPSO (Diversity-based multiple PSO) [3.85], MSPSO (Multi-species PSO) [3.81], PSO-PAS (PSO for computing plural acceptable solutions) [3.86] が提案されている。

母集団からの分化や移住操作を必要としない手法として免疫機構のメカニズムを導入した Suppression PSO [3.88] や CoPSO (Competitive PSO) [3.82] が提案されている。

CoPSO では, 予め母集団から複数のサブ集団を生成し, 各サブ集団は独立して標準的な PSO として探索を行う。各サブ集団は他のサブ集団と同一の解領域を探索することを抑制するために優先評価範囲を有する。例えば, サブ集団 g の優先評価範囲内にサブ集団 h の粒子 i が移動した場合, 粒子 i が \mathbf{pb}_i の評価値よりも良好な評価値である解を発見したとしても, 粒子 i は評価および \mathbf{pb}_i の更新を行うことはできない。このため, 各サブ集団は他サブ集団が探索している領域を探索することなく, 複数の近似解を並列的に探索することができることが示されている。

PSO-PAS [3.87], Suppression PSO [3.88], CoPSO [3.83],[3.84] は無線センサネットワークのアプリケーションに適用されている。

3.5 PSO アルゴリズムの並列計算環境への実装の研究

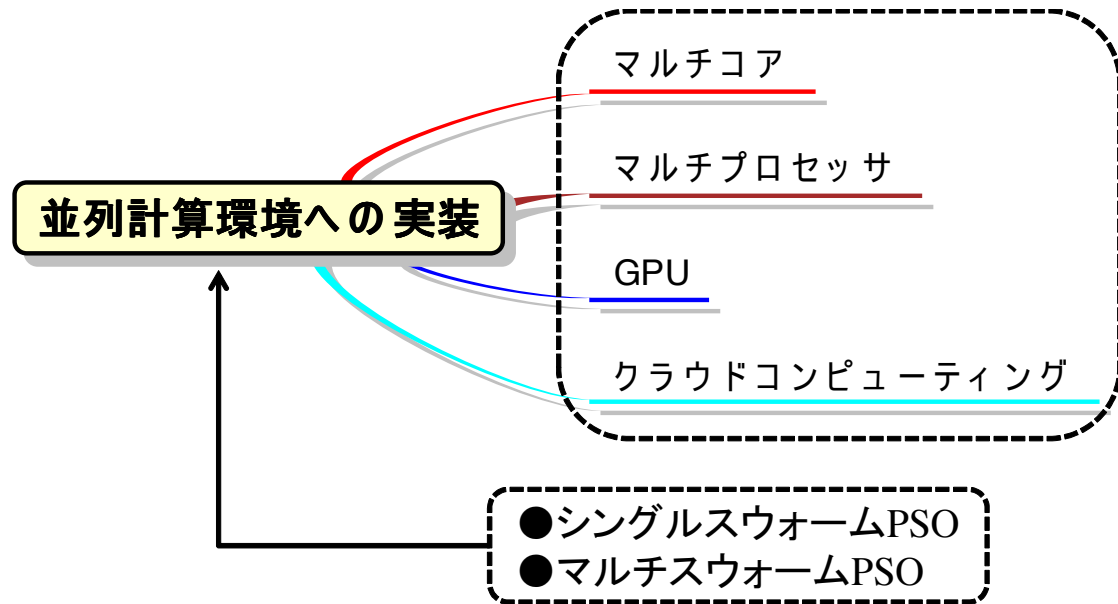


図 3.13: PSO アルゴリズムの並列計算環境への実装の研究の分類

大規模な最適化問題に対して高精度な近似解を発見するためには、多数の粒子を用いることが望ましいが、計算量も増大する。数百、数千のセンサノードのパラメータの最適化 [3.27]、数千、数万のセンサノードにおける FNSP の最適化 [3.26]、大多数のタスク割り当ての最適化 [3.21]、経済給電の最適化 [3.22]、大多数の文書分類の最適化 [3.23]、大多数のポートフォリオの最適化 [3.24] などは計算量が膨大である。このため、PSO アルゴリズムの並列計算環境への実装により計算時間を低減することが求められている。

3.3, 3.4 で説明した手法の多くは並列計算環境への実装は考慮されておらず、粒子間の近傍構造やサブ集団間を調整するために複雑な処理を必要とする手法も少なくない。

本節では、PSO において計算時間の短縮を目的とした PSO アルゴリズムの並列計算環境への実装に関する研究 [3.98]-[3.116] について説明する。

PSO アルゴリズムを PC クラスタやマルチコアの並列計算環境へ実装する手法 [3.98]-[3.102],[3.110]-[3.112] や GPU (Graphics processing unit) へ実装する手法 [3.103]-[3.109],[3.113]-[3.116] が提案されており、多数の粒子の計算負荷を分散させることで高速に近似解を発見できることが示されている。

本節では、PSO の母集団が 1 つであるシングルスウォーム PSO と複数個の集団からなるマルチスウォーム PSO についてそれぞれの実装法に対する研究を紹介する。

3.5.1 シングルスウォーム PSO の実装に関する研究

シングルスウォーム PSO の並列計算環境への実装に関する研究として [3.98]-[3.109] がある。

文献 [3.98]-[3.100] では、全ての粒子の状態更新を終えてから、イタレーションの更新を行う同期型の実装手法 (Parallel synchronous PSO; PSPSO) と非同期的に粒子の状態更新を行う非同期型の実装手法 (Parallel asynchronous PSO; PAPSO) が提案されている。PSPSO および PAPSO では、1 つの粒子を 1 つの計算機 (プロセッサ) に割り当てる実装手法により実現されている。

PAPSO は PSPSO よりも待機状態となる計算機の数を抑えることができ、計算時間も短縮できることが示されている [3.99],[3.100]。

文献 [3.103]-[3.109] では、PSO アルゴリズムを GPU へ実装する手法が提案されており、これらの手法は PC などの計算機における計算時間よりも大幅に計算時間を短縮できる。

3.5.2 マルチスウォーム PSO の実装に関する研究

マルチスウォーム PSO の実装手法 [3.101],[3.110]-[3.116] では、プロセッサ間通信のボトルネックを小さくするために各サブ集団における *gbest* の共有、サブ集団間の粒子の移住、CPSO の実装など簡素な情報共有の構造を有する手法が多い。

参考文献

- [3.1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. of IEEE Int. Conf. on Neural Networks*, pp. 1942-1948, 1995.
- [3.2] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. of IEEE Int. Conf. evolutionary computation*, pp. 69-73, 1998.
- [3.3] R. C. Eberhart and J. Kennedy, "Human tremor analysis using particle swarm optimization," in *Proc. of Congr. Evol. Comput.*, pp. 1927-1930, 1999.
- [3.4] H. Yoshitaka, K. Kawata, Y. Fukuyama, S. Takayama and Y. Nakanishi, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment," *IEEE Trans. Power Syst.*, Vol. 15, No. 4, pp. 1232-1239, 2000.
- [3.5] G. Kókai, T. Christ and H. H. Frhauf, "Using hardware-based particle swarm method for dynamic optimization of adaptive array antennas," in *Proc. of First NASA/ESA Conference on Adaptive Hardware and Systems*, pp. 51-58, 2006.
- [3.6] A. Farmahini-Farahani, S. M. Fakhraie and S. Safari, "Scalable architecture for on-chip neural network training using swarm intelligence," *EDAA 2008, Design, Automation, and Test in Europe*, pp. 1340-1345, 2008.
- [3.7] M. R. Alrashidi and M. E. El-Hawary, "A survey of particle swarm optimization applications in electric power systems," *IEEE Trans. Evol. Comput.*, Vol. 13, pp. 913-918, 2009.
- [3.8] W. F. Abf-El-Wahed, A. A. Mousa and M. A. El-Shorbagy, "Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems," *Journal of Computational and Applied Mathematics*, Vol. 235, No. 5, pp. 1446-1453, 2011.
- [3.9] M. S. Kiran, E. Özceylan, M. Gündüz and T. Paksoy, "A novel hybrid approach based on particle swarm optimization and ant colony algorithm to forecast energy demand of turkey," *Energy Conversion and Management*, Vol. 53, No. 1, pp. 75-83, 2012.
- [3.10] T. Niknam, M. R. Narimani and M. Jabbari, "Dynamic optimal power flow using hybrid particle swarm optimization and simulated annealing," *International Transactions on Electrical Energy Systems*, Vol. 23, No. 7, pp. 975-1001, 2013.
- [3.11] X. Xu, J. Cao, H. Shan, L. Zhu and J. Guo, "An adaptive hybrid algorithm based on particle swarm optimization and differential evolution for global optimization," *The Scientific World Journal*, Vol. 2014, Article ID 215472, 16 pages, 2014.

- [3.12] P. J. Angeline, "Evolutionary optimization versus particle swarm optimization : philosophy and performance differences," in *Proc. of 7th Annu. Conf. Evolutionary Programming VII*, pp. 601-610, 1998.
- [3.13] J. Sun, B. Feng, and W. B. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proc. of IEEE CEC*, pp. 325-331, 2004.
- [3.14] J. Kennedy, "Bare bones particle swarms," in *Proc. of IEEE SIS*, pp. 80-87, 2003.
- [3.15] Y.-T. Juang, S.-L. Tung and H.-C. Chiu, "Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions," *Information Sciences*, Vol. 181, No. 20, pp. 4539-4549, 2011.
- [3.16] E. Ozcan and C. K. Mohan, "Particle swarm optimization : Surfing the waves," in *Proc. of IEEE CEC*, pp. 1939-1944, 1999.
- [3.17] M. Clerc and J. Kennedy, "The particle swarm-Explosion, stability and convergence in a multidimensional complex space," *IEEE Trans. on Evolutionary Computation*, Vol. 6, No. 2, pp. 58-73, 2002.
- [3.18] V. Kadirkamanathan, K. Selvarajah and P. J. Fleming, "Stability analysis of the particle dynamics in particle swarm optimizer," *IEEE Trans. on Evolutionary Computation*, Vol. 10, No. 3, pp. 245-255, 2006.
- [3.19] K. Jin'no, "A novel deterministic particle swarm optimization system," *Journal of Signal Processing*, Vol. 13, No. 6, pp. 507-513, 2009.
- [3.20] K. Jin'no and T. Shindo, "Analysis of dynamical characteristic of canonical deterministic PSO," in *Proc. of IEEE CEC*, pp. 1105-1110, 2010.
- [3.21] S.-Y. Ho, H.-S. Lin, W.-H. Liauh and S.-J. Ho, "OPSO: Orthogonal particle swarm optimization and its application task assignment problems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, Vol. 38, No. 2, pp. 288-298, 2008.
- [3.22] R. Singh, K. Jain and M. Pandit, "Comparison of PSO variants with traditional solvers for large scale multi-area economic dispatch," in *Proc. of SEISCON*, pp. 304-309, 2011.
- [3.23] J. Platos, V. Snasel, T. Jezowicz, P. Kromer and A. Abraham, "A PSO-based document classification algorithm accelerated by the CUDA platform," in *Proc. of IEEE SMC*, pp. 1936-1941, 2012.

- [3.24] J. J. Liang and B. Y. Qu, "Large-scale portfolio optimization using multi-objective dynamic multi-swarm particle swarm optimizer," in *Proc. of IEEE SIS*, pp. 1-6, 2013.
- [3.25] J.-W. Lee, T. Choi, H. Do, D. Park, C. Park and Y.-S. Son, "Experimental results of heterogeneous cooperative bare bones particle swarm optimization with Gaussian jump for large scale global optimization," in *Proc. of IEEE CEC*, pp. 1979-1985, 2015.
- [3.26] T. Sasaki, H. Nakano, A. Miyauchi and H. Yamamoto, "An efficient flooding scheme using chaotic neural networks in wireless sensor networks," in *Proc. of NOLTA*, pp. 523-526, 2009.
- [3.27] 伊藤 啓太, 長島 淳也, 宇谷 明秀, 山本 尚生, "無線センサネットワークの高効率化を目指したフラッディング送信電力設計・運用法の一検討," 電子情報通信学会論文誌 *B*, Vol. 95-B, No. 6, pp. 760-764, 2012.
- [3.28] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. of IEEE SMC*, pp. 4104-4108, 1997.
- [3.29] J. Nagashima, A. Utani and H. Yamamoto, "Efficient flooding method using discrete particle swarm optimization for long-term operation of sensor networks," *ICIC Express Letters*, Vol. 3, No. 3(B), pp. 833-840, 2009.
- [3.30] 水田 夏美, 中野 秀洋, 宮内 新, "Zigbee センサネットワークにおけるルータ配置の最適化手法," 電子情報通信学会技術研究報告. *NLP*, 非線形問題研究会 113(486), pp. 73-77, 2014.
- [3.31] L. B. Zhang, C. G. Zhou, X. H. Liu, Z. Q. Ma, M. Ma and Y. C. Liang, "Solving multi objective optimization problems using particle swarm optimization," in *Proc. of IEEE CEC*, Vol. 4, pp. 2400-2405, 2003.
- [3.32] L. Cao and D. Zhang, "Aerodynamic configuration optimization for hypersonic gliding vehicle based on improved hybrid multi-objective PSO algorithm," in *Proc. of IEEE ICSPCC*, pp. 1-5, 2015.
- [3.33] J. Peña and A. Upegui, "A population-oriented architecture for particle swarms," in *Proc. of Second Nasa/ESA Conference on Adaptive Hardware and Systems (AHS)*, pp. 563-570, 2007.
- [3.34] A. Farmahini-Farahani, S. M. Fakhraie and S. Safari, "SOPC-based architecture for discrete particle swarm optimization," in *Proc. of IEEE ICECS*, pp. 1003-1006, 2007.
- [3.35] D. M. Muñoz, C. H. Llanos, L. dos S. Coelho and M. Ayala-Rincón, "Hardware architecture for particle swarm optimization using floating-point arithmetic," in *Proc. of International Conference on Intelligent Systems Design and Applications*, pp. 243-248, 2009.

- [3.36] D. M. Muñoz, C. H. Llanos, L. dos S. Coelho and M. Ayala-Rincón, “Hardware particle swarm optimization based on the attractive-repulsive scheme for embedded applications,” in *Proc. of International Conference on Reconfigurable Computing and FPGAs*, pp. 55-60, 2010.
- [3.37] D. M. Muñoz, C. H. Llanos, L. dos S. Coelho and M. Ayala-Rincón, “Comparison between two FPGA implementations of the particle swarm optimization algorithm for high-performance embedded applications,” in *Proc. of IEEE BIC-TA*, pp. 1637-1645, 2010.
- [3.38] D. M. Muñoz, C. H. Llanos, L. dos S. Coelho and M. Ayala-Rincón, “Hardware particle swarm optimization with passive congregation for embedded applications,” in *Proc. of SPL*, pp. 173-178, 2011.
- [3.39] H. Nakano and A. Miyauchi, “Design of a processor system for particle swarm optimizers,” in *Proc. of NOLTA*, pp. 606-609, 2015.
- [3.40] J. Kennedy, “Small worlds and mega-minds : effects of neighborhood topology on particle swarm performance,” in *Proc. of IEEE CEC*, pp. 1931-1938, 1999.
- [3.41] J. Kennedy and R. Mendes, “Population structure and particle swarm performance,” in *Proc. of IEEE CEC*, pp. 1671-1676, 2002.
- [3.42] R. Mendes, J. Kennedy and J. Neves, “Watch try neighbor or how the swarm can learn from its environment,” in *Proc. of IEEE*, pp. 88-94, 2003.
- [3.43] R. Mendes, J. Kennedy and J. Neves, “The fully informed particle swarm : simpler, maybe better,” *IEEE Trans. on Evolutionary computation*, Vol. 8, No. 3, pp. 204-210, 2004.
- [3.44] X. D. Li, “Niching without niching parameters : Particle swarm optimization using a ring topology,” *IEEE Trans. Evolutionary Computation*, Vol. 14, No. 1, pp. 150-169, 2010.
- [3.45] J. Rada-Vilela, M. J. Zhang and W. Seah, “A performance study on synchronicity and neighborhood size in particle swarm optimization,” *Soft Computing*, Vol. 17, No. 6, pp. 1019-1030, 2013.
- [3.46] H. Wang, H. Sun, C. Li, S. Rahnamayan and J. -S. Pan, “Diversity enhanced particle swarm optimization with neighborhood search,” *Information Sciences*, Vol. 223, pp. 119-135, 2013.
- [3.47] H. Matsushita, Y. Nisio and C. K. Tse, “Network-structured particle swarm optimizer that considers neighborhood distance and behaviors,” *Journal of Signal Processing*, Vol. 18, No. 6, pp. 291-302, 2014.

- [3.48] J. Lane, A. Engelbrecht and J. Gain, "Particle swarm optimization with spatially meaningful neighbors," in *Proc. of IEEE SIS*, 2008.
- [3.49] S. Cheng, Y. Shi and Q. Qin, "Population diversity based study on search information propagation in particle swarm optimization," in *Proc. of IEEE CEC*, pp. 1272-1279, 2012.
- [3.50] J. Kennedy and R. Mendes, "Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms," *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 36, Issue : 4, pp. 515-519, 2006.
- [3.51] P. N. Suganthan, "Particle swarm optimiser with neighborhood operator," in *Proc. of IEEE CEC*, pp. 1958-1963, 1999.
- [3.52] T. Peram, K. Veeramachaneni and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," in *Proc. of IEEE SIS*, pp. 174-181, 2003.
- [3.53] K. Veeramachaneni, T. Peram, C. Mohan and L. A. Osadciw, "Optimization using particle swarms with near neighbor interactions," in *Proc. of Genetic and Evol. Comput.*, Vol. 2723, pp. 110-121, 2003.
- [3.54] X. Li, "A multimodal particle swarm optimizer based on fitness euclidean-distance ratio," in *Proc. of GECCO '07*, S. Zhang and R. Jarvis, Eds., Vol. 3809, Springer, pp. 776-785, 2005.
- [3.55] Y. X. Wang and Q. L. Xiang, "Particle swarms with dynamic ring topology," in *Proc. of IEEE CEC*, pp. 419-423, 2008.
- [3.56] S. B. Akat and V. Gazi, "Particle swarm optimization with dynamic neighborhood topology : Three neighborhood strategies and preliminary results," *proc. of IEEE SIS*, pp. 1-8, 2008.
- [3.57] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant," *IEEE Trans. on System Man and Cybernetics B* 35(6), pp. 1272-1282, 2005.
- [3.58] D. F. Carvalho and Carmelo J. A. Bastos-Filho, "Clan particle swarm optimization," in *Proc. of IEEE CEC*, pp. 3044-3051, 2008.
- [3.59] E. Miyagawa and T. Saito, "Particle swarm optimizers with growing tree topology," *IEICE Trans. Fundamentals*, Vol. E92-A, No. 3, pp. 2275-2282, 2009.
- [3.60] A. S. Mohais, C. Ward and C. Posthoff, "Randomized directed neighborhoods with edge migration in particle swarm optimization," in *Proc. of IEEE CEC*, pp. 548-555, 2004.

- [3.61] H. Matsushita, Y. Nishio and T. Saito, "Particle swarm optimization with novel concept of complex network," in *Proc. of NOLTA*, pp. 197-200, 2010.
- [3.62] C. Zhang and Z. Yi, "Scale-free informed particle swarm optimization algorithm," *Information Science*, Vol. 181, pp. 4550-4568, 2011.
- [3.63] B. Jiang, N. Wang and L. P. Wang, "Particle swarm optimization with age-group topology for multimodal functions and data clustering," *Communication in Nonlinear Science and Numerical Simulation*, Vol. 18, No. 11, pp. 3134-3145, 2013.
- [3.64] F. Li and J. Guo, "Topology optimization of particle swarm optimization," *Advances in Swarm Intelligence Vol. 8794 of the series Lecture Notes in Computer Science*, pp. 142-149, 2014.
- [3.65] K. Jin'no, R. Sano and T. Saito, "Particle swarm optimization with switched topology," *Non-linear Theory and its Applications, IEICE*, Vol. 6, No. 2, pp. 181-193, 2015.
- [3.66] M. A. Montes de Oca, T. Stutzle, M. Birattari and M. Dorigo, "Frankenstein's PSO : A composite particle swarm optimization algorithm," *IEEE Trans. on evolutionary computation*, Vol. 13, No. 5, pp. 1120-1132, 2009.
- [3.67] Y. Marinakis and M. Marinaki, "A hybridized particle swarm optimization with expanding neighborhood topology for the feature selection problem," in : *M. Blesa et al. (Eds) : Hybrid Metaheuristics. Springer, Berlin, Heidelberg*, pp. 37-51, 2013.
- [3.68] Y. Marinakis and M. Marinaki, "Particle swarm optimization with expanding neighborhood topology for the permutation flow shop scheduling problem," *Soft Computiong*, Vol. 17, No. 7, pp. 1159-1173, 2013.
- [3.69] W. H. Lim and N. A. Mat Isa, "Particle swarm optimization with increasing topology connectivity," *Engineering Applications of Artificial Intelligence*, Vol. 27, pp. 80-102, 2014.
- [3.70] J. Horn, "The nature of niching : Genetic algorithms and the evolution of optimal, cooperative populations," *Doctoral dissertation, 95001, Urbana, University of Illinois, Illinois Genetic Algorithm Lab*, 1997.
- [3.71] M. Løvbjerg, T. K. Rasmussen and T. Krink, "Hybrid particle swarm optimiser with breeding and subpopulations," in *Proc. of GECCO*, 2001.
- [3.72] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proc. of IEEE SIS*, pp. 124-129, 2005.

- [3.73] S. Z. Zhao, J. J. Liang, P. N. Suganthan and M. F. Tasgetiren, "Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization," in *Proc. of IEEE CEC*, pp. 3845-3852, 2008.
- [3.74] S. Z. Zhao, P. N. Suganthan and S. Das, "Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search," in *Proc. of IEEE CEC*, pp. 1-8, 2010.
- [3.75] J. J. Liang, Q. Pan and T. Chen, "A dynamic multi-swarm particle swarm optimizer for blocking flow shop scheduling," in *Proc. of IEEE BIC-TA*, pp. 323-327, 2010.
- [3.76] J. J. Liang, H. Song and B. Y. Qu, "Performance evaluation of dynamic multi-swarm particle swarm optimizer with different constraint handling methods on path planning problems," in *Proc. of IEEE MC*, pp. 65-71, 2013.
- [3.77] M. Sugimoto, T. Haraguchi, H. Matsushita and Y. Nishio, "Particle swarm optimization containing plural swarms," in *Proc. of NCSP*, pp. 419-422, 2009.
- [3.78] F. Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. on Evolutionary Computing*, Vol. 8, No. 3, pp. 225-239, 2004.
- [3.79] X. Li and X. Yao, "Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms," in *Proc. of IEEE CEC*, pp. 1546-1553, 2009.
- [3.80] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Trans. on Evolutionary Computing*, Vol. 16, No. 2, pp. 210-224, 2012.
- [3.81] M. Iwamatsu, "Multi-species particle swarm optimizer for multimodal function optimization," *IEICE Trans. Inf. and Syst.*, Vol. E89-D, No.3, pp. 1181-1187, 2006.
- [3.82] Y. Taguchi, H. Nakano, A. Utani, A. Miyauchi and H. Yamamoto, "A competitive particle swarm optimization for finding plural acceptable solutions," *ICIC Express Letters*, Vol. 4, No. 5, pp. 1-6, 2010.
- [3.83] Y. Taguchi, Y. Kanamori, H. Nakano, A. Utani, A. Miyauchi and H. Yamamoto, "A competitive PSO based on evaluation with priority for finding plural solutions," in *Proc. of IEEE CEC*, pp. 1-6, 2010.
- [3.84] H. Nakano, Y. Taguchi, Y. Kanamori, A. Utani, A. Miyauchi and H. Yamamoto, "A competitive particle swarm optimizer and its application to wireless sensor networks," *IEEJ Trans.*, 7, SI, pp. 52-58, 2012.

- [3.85] G. G. Yen and M. Daneshyari, "Diversity-based information exchange among multiple swarm in particle swarm optimization," in *Proc. of IEEE CEC*, pp. 6150-6157, 2006.
- [3.86] A. Kumamoto, A. Utani and H. Yamamoto, "Advanced particle swarm optimization for computing plural acceptable solutions," *Int. J. Innovative Computing, Information and Control*, Vol. 5, No. 11, pp. 4383-4392, 2009.
- [3.87] 長島 淳也, 宇谷 明秀, 山本 尚生, "複数許容解探索型粒子群最適化法の無線センサネットワークへの適用 -フラッディング効率化のための各センサノードの送信電力調整-, 知能と情報 (日本知能情報ファジィ学会誌), Vol. 23, No. 1, pp. 65-77, 2011.
- [3.88] M. Yoshimura, H. Nakano, A. Utani, A. Miyauchi and H. Yamamoto, "An effective allocation scheme for sink nodes in wireless sensor networks using suppression PSO," *ICIC Express Letters*, Vol. 3, No. 3 (A), pp. 519-524, 2009.
- [3.89] A. E. R. Brits and F. van den Bergh, "A niching particle swarm optimizer," in *Proc. of 4th Asia-Pracif. Conf. Simul. Evol. Learn. (SEAL 2002)*, pp. 692-696, 2002.
- [3.90] A. Nickabadi, M. M. Ebadzadeh and R. Safabakhsh, "DNPSO : A dynamic niching particle swarm optimizer for multimodal optimization," in *Proc. of IEEE CEC*, pp. 26-32, 2008.
- [3.91] M. Zhou and J. Wang, "A clustering based niching particle swarm optimization for locating multiple optimum solutions," in *Proc. of ICICTA*, pp. 211-214, 2009.
- [3.92] J. Wang, D. Liu and H. Shang, "Hill valley function based niching particle swarm optimization for multimodal functions," in *Proc. of AICI*, pp. 139-144, 2009.
- [3.93] B. Y. Qu, J. J. Liang and P. N. Suganthan, "Niching particle swarm optimization with local search for multi-modal optimization," *Information Sciences 197*, pp. 131-143, 2012.
- [3.94] T. Hsieh, C. Cheng and W. Yeh, "A hybrid niching-based evolutionary PSO for numerical optimization problems," in *Proc. of Computational Intelligence and Cybernetics*, pp. 133-137, 2012.
- [3.95] J. Liao, Y. Liu, X. Zhu, T. Xu and J. Wang, "Niching particle swarm optimization algorithm for service composition," in *Proc. of GLOBECOM*, pp. 1-6, 2011.
- [3.96] D. Ma, J. Ma and P. Xu, "An adaptive assistant-aided clustering protocol for WSNs using niching particle swarm optimization," in *Proc. of IEEE ICSESS*, pp. 648-651, 2013.

- [3.97] D. Ma, P. Xu and J. Ma, "An efficient node partition clustering protocol using niching particle swarm optimization," in *Proc. of IEEE ICSESS*, pp. 644-647, 2013.
- [3.98] J. F. Shutte, J. A. Reinbolt, B. J. Fregly, R. T. Haftka and A. D. George, "Parallel global optimization with the particle swarm algorithm," *Int. J. Number Methods. Enq.* 61(13), pp. 2296-2315, 2004.
- [3.99] G. Venter and J. S-. Sobieski, "A parallel particle swarm optimization algorithm accelerated by asynchronous evaluations," in *Proc. of 6th World Congresses of Structural and Multidisciplinary Optimization*, pp. 1-10, 2005.
- [3.100] B. -I. Koh, A. D. George, R. T. Haftka and B. J. Fregly, "Parallel asynchronous particle swarm optimization," *International Journal for Numerical Methods in Engineering*, 67(4), pp. 578-595, 2006.
- [3.101] K. Deep, S. Sharma and M. Pant, "Modified parallel particle swarm optimization for global optimization using message passing interface," in *Proc. of IEEE BIC-TA*, pp. 1451-1458, 2010.
- [3.102] A. McNabb and K. Seppi, "Serial PSO results are irrelevant in a multi-core parallel world," in *Proc. of IEEE CEC*, pp. 3143-3150, 2014.
- [3.103] Y. Zhou and Y. Tan, "GPU-based parallel particle swarm optimization," in *Proc. of IEEE CEC*, pp. 1493-1500, 2009.
- [3.104] L. P. Veronese and R. A. Krohling, "Swarm's flight : Accelerating the particles using C-CUDA," in *Proc. of IEEE CEC*, pp. 3264-3270, 2009.
- [3.105] M. Cárdenas-Montes, M. A. Vega-Rodríguez, J. J. Rodríguez-Vázquez and A. Gómez-Iglesias, "Effect of the block occupancy in GPGPU over the performance of particle swarm algorithm," *Adaptive and Natural Computing Algorithms, Vol. 6593 of the series Lecture Notes in Computer Science*, pp. 310-319, 2011.
- [3.106] L. Mussi, Y. S. G. Nashed and S. Cagnoni, "GPU-based asynchronous particle swarm optimization," in *Proc. of GECCO*, pp. 1555-1562, 2011.
- [3.107] V. Roberge and M. Tarbouchi, "Efficient parallel particle swarm optimizers on GPU for real-time harmonic minimization in multilevel inverters," in *Proc. of IEEE IECON*, pp. 2275-2282, 2012.
- [3.108] R. M. Calazan, N. Nedjah and L. M. Mourelle, "Parallel GPU-based implementation of high dimensional particle swarm optimizations," in *Proc. of LASCAS*, pp. 1-4, 2013.

-
- [3.109] P. Krömer, J. Platoš and V. Snášel, “A brief survey of advances in particle swarm optimization on graphic processing units,” in *Proc. of NaBIC*, pp. 182-188, 2013.
- [3.110] 波平 光洋, 名嘉村 盛和, “並列 Particle swarm optimization における協調処理,” 信学技法, *IEICE Technical Report CAS2007-77, CST2007-28*, pp. 21-26, 2007.
- [3.111] D. Wang, C. -H. Wu, A. Ip, D. Wang and Y. Yan, “Parallel multi-population particle swarm optimization algorithm for the uncapacitated facility location problem using OpenMP,” in *Proc. of IEEE CEC*, pp. 1214-1218, 2008.
- [3.112] X. Lai and G. Tan, “Studies on migration strategies of multiple population parallel particle swarm optimization,” in *Proc. of ICNC*, pp. 798-802, 2012.
- [3.113] M. Rabinovich, P. Kainga, D. Johnson, B. Shafer, J. J. Lee and R. Eberhart, “Particle swarm optimization on a GPU,” in *Proc. of IEEE EIT*, pp.1-6, 2012.
- [3.114] M. Jin and H. Lu, “Parallel particle swarm optimization with genetic communication strategy and its implementation on GPU,” in *Proc. of CCIS*, pp. 99-104, 2012.
- [3.115] R. M. Calazan, N. Nedjah and L. M. Mourelle, “A cooperative parallel particle swarm optimization for high-dimension problems on GPUs,” in *Proc. of BRICS*, pp. 356-361, 2013.
- [3.116] J. Kumar, L. Singh and S. Paul, “GPU based parallel cooperative particle swarm optimization using C-CUDA : A case study,” in *Proc. of IEEE FUZZ*, pp. 1-8, 2013.

第四章 粒子群最適化法のネットワーク化による 解探索性能の向上手法

4 粒子群最適化法のネットワーク化による解探索性能の向上手法

4.1 4章の概要

3章で説明したように、マルチスウォーム PSO と粒子間に近傍構造を有する PSO 手法はアルゴリズムの多様性が高く多峰性問題に対しても有効である。本章では、これら2つの手法の概念を組み合わせることで PSO アルゴリズムの多様性を向上させる手法を提案する。

提案する方法では、粒子群の母集団を複数の小規模な粒子群 (サブ PSO) に分割し、各サブ PSO は静的な近傍構造に基づき近傍サブ PSO が決定される。各サブ PSO における粒子は自身の良解情報 $pbest$ 、粒子が属するサブ PSO の良解情報 $lbest$ 、近傍サブ PSO 間における良解情報 $glbest$ を参照し、解空間上を飛び回る。このため、各サブ PSO は解空間上を独立して探索を行う。

各サブ PSO はそれぞれの近傍サブ PSO と情報交換することができ、各サブ PSO の近傍に対するサブ PSO 数はパラメータ DBG (Degree between groups) によって調整することができる。 DBG の値を小さく設定すれば、近傍のサブ PSO 数は少なくなり近傍構造が疎となり多様性が向上する。一方、 DBG の値を大きくすれば、近傍のサブ PSO 数は多くなり、近傍構造が密となり多様性が減少する。このため、 DBG の調整により解空間の構造に適した近傍構造を設定することができる。

各サブ PSO は自身の $lbest$ が更新されたときに近傍のサブ PSO と情報交換を行い、その際、自身の $lbest$ を通信する。提案手法を並列計算環境へ実装する場合、1つのサブ PSO を1つの計算機 (プロセッサ) に割り当てればよく、計算機間の通信 (プロセッサ間通信) のデータ量 ($lbest$) も小さいため、並列計算環境への実装に適している。

次節以降では、提案手法の説明を行い、その有効性を確認するために数値実験を行う。まず、提案手法の粒子が安定して良解に収束するパラメータ条件を解析的に求め、数値実験により解析結果の妥当性を検証する。次に、 DBG と解空間の構造における解探索性能について調べ、適切な DBG の設定指針を示す。さらに、提案手法の有効性を確認するために、マルチスウォーム PSO や近傍構造を有する他の PSO 手法と解探索性能の比較実験を行う。

4.2 粒子群最適化ネットワーク

本節では、粒子群最適化法を静的な近傍構造によりネットワーク化した手法である PSO Networks (PSON) を提案し、その説明を行う。PSON において、粒子群の母集団は複数の小規模なサブ集団 (サブ PSO) に分割され、各サブ PSO はグラフ理論に基づいた静的な近傍構造によって近傍のサブ PSO が割り当てられる。 g 番目のサブ PSO (PSO_g) の i 番目の粒子 (粒子 i) は、速度ベクトル $\mathbf{v}_{g,i} = (v_{g,i1}, v_{g,i2}, \dots, v_{g,iD})$ 、位置ベクトル $\mathbf{x}_{g,i} = (x_{g,i1}, x_{g,i2}, \dots, x_{g,iD})$ 、自身の探索過程における最良解ベクトル $\mathbf{pbest}_{g,i} = (pb_{g,i1}, pb_{g,i2}, \dots, pb_{g,iD})$ を持ち、 PSO_g の探索過程における最良解ベクトル $\mathbf{lbest}_g = (lb_{g,1}, lb_{g,2}, \dots, lb_{g,D})$ と PSO_g の近傍サブ PSO の探索過程における最良解ベクトル

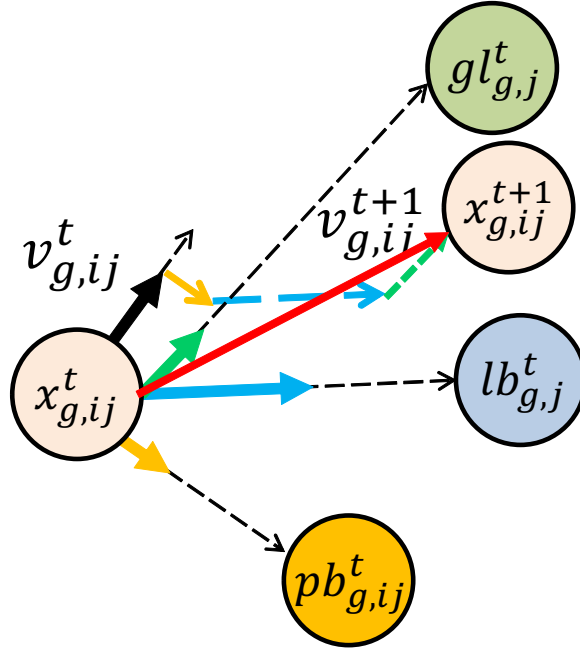


図 4.1: PSQN の粒子の概念図

$\mathbf{glbest}_g = (gl_{g,1}, gl_{g,2}, \dots, gl_{g,D})$ を共有する. D は最適化する設計変数の総数を表す. PSO_g における粒子 i の j 次元要素の速度と位置の更新式を式 (4.1), (4.2) に示し, 図 4.1 に PSQN の粒子の概念図を示す.

$$v_{g,ij}^{t+1} = wv_{g,ij}^t + c_1 r_1 (pb_{g,ij}^t - x_{g,ij}^t) + c_2 r_2 (lb_{g,j}^t - x_{g,ij}^t) + c_3 r_3 (gl_{g,j}^t - x_{g,ij}^t) \quad (4.1)$$

$$x_{g,ij}^{t+1} = x_{g,ij}^t + v_{g,ij}^{t+1} \quad (4.2)$$

ここで, w は慣性速度定数, c_1, c_2, c_3 は加速度係数, r_1, r_2, r_3 は $[0, 1]$ の一様乱数を表す. PSQN における粒子の速度更新式は式 (3.1) で示した標準的な PSO の速度更新に \mathbf{glbest} との差分項を追加したものである. PSO_g において, \mathbf{lbest}_g が更新されたときに PSO_g の近傍サブ PSO と \mathbf{lbest}_g の通信を行い, これらの近傍サブ PSO は自身の \mathbf{glbest} を更新する. その更新式は次式となる (最小化問題 $f(\mathbf{x})$ において).

$$\mathbf{glbest}_k^{t+1} = \begin{cases} \mathbf{lbest}_g^{t+1} & , \text{ if } f(\mathbf{lbest}_g^{t+1}) < f(\mathbf{glbest}_k^t) \\ \mathbf{glbest}_k^t & , \text{ otherwise} \end{cases} \quad (4.3)$$

このことによって, PSO_g が局所解に収束したとしても, 近傍サブ PSO からの良解情報により \mathbf{glbest}_g が更新されることで PSO_g はその局所解から脱出することができ, より良好な解探索を行うことができる.

図 4.2 にサブ PSO 数が 6 個 (1 個のサブ PSO に 6 個の粒子) であり, 各サブ PSO に 2 つの近傍サ

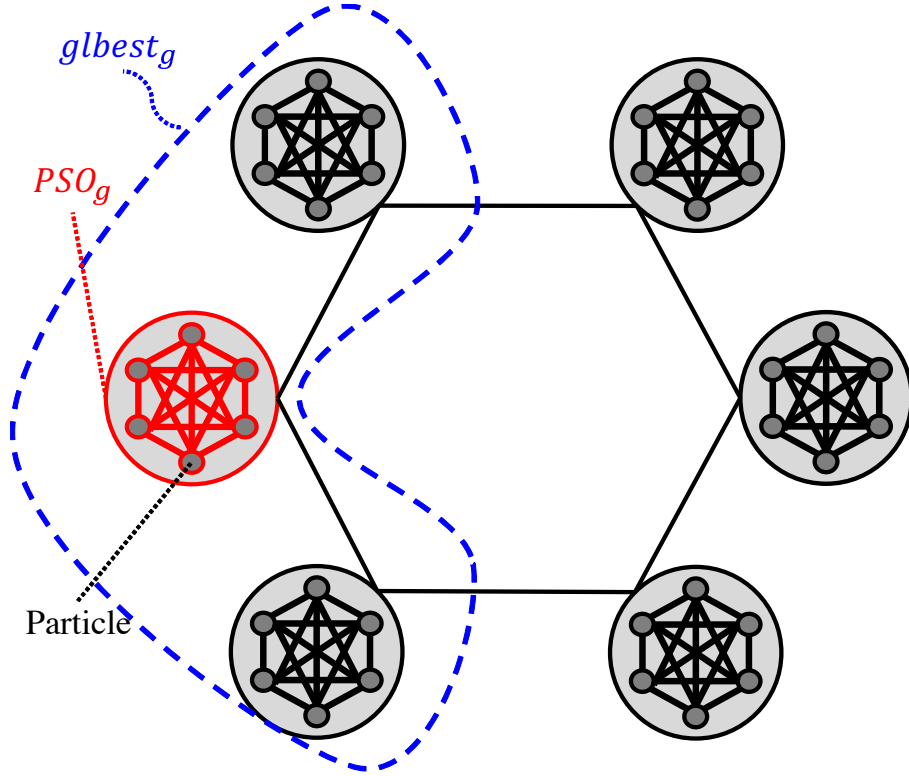
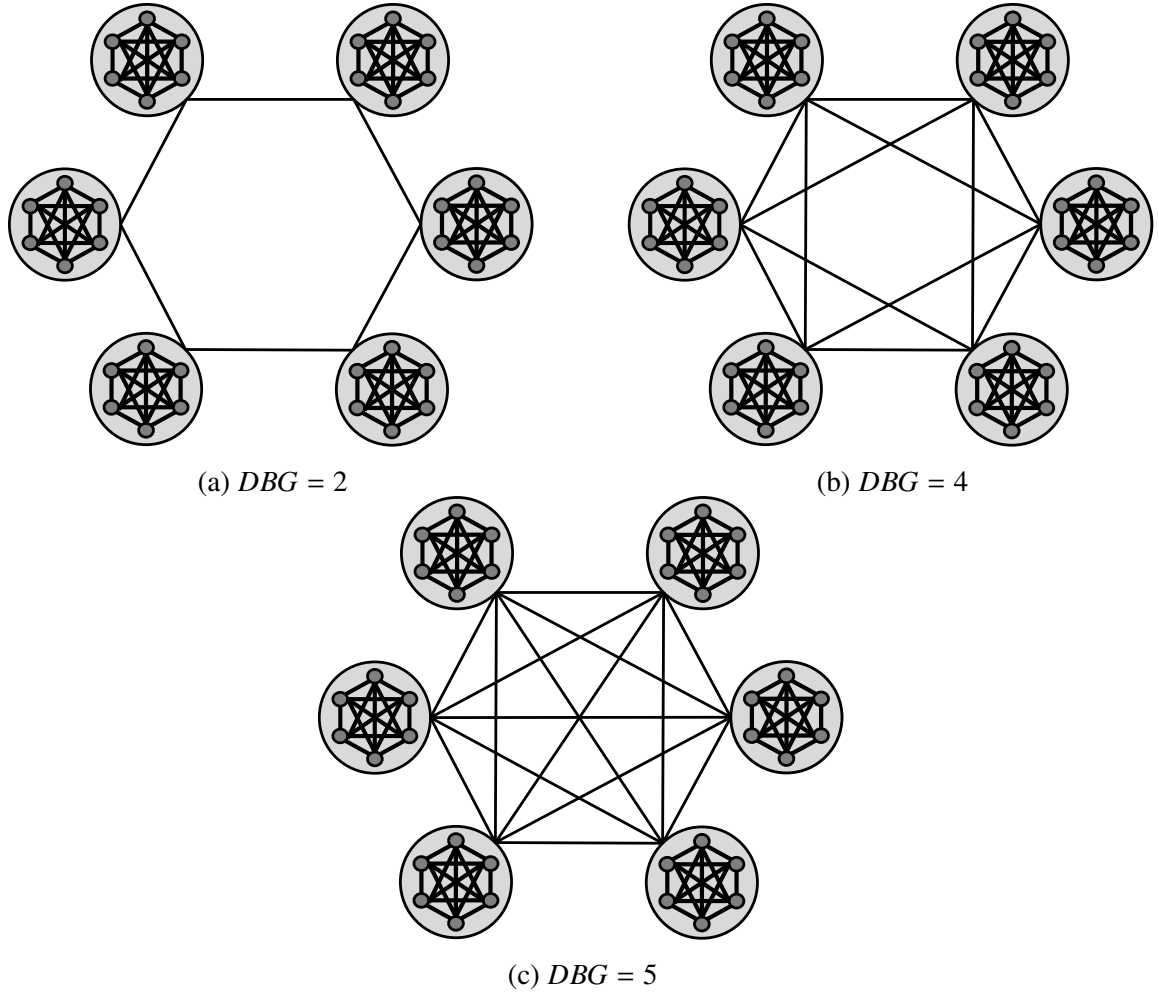


図 4.2: PSO_N の例

ブ PSO が割り当てられている PSO_N の例を示す。同図において、 PSO_g は 2 つの近傍サブ PSO からの通信により $glbest_g$ が更新されることを表し、他の 3 つのサブ PSO の良解情報は直接受け取ることができない。PSO_N において、近傍サブ PSO 数が少ない場合は全サブ PSO における最良解情報 (i.e., $glbest$) が全サブ PSO に伝搬する速度が遅く、逆に近傍サブ PSO 数が多い場合はその伝搬が速くなることから、最適化問題の種類に合わせて適切な近傍構造を設定する必要がある。PSO_N の近傍構造はグループ間次数 (Degree Between Groups; DBG) により決定する。図 4.3 にはサブ PSO 数が 6 個であり、 DBG を 2, 4, 5 と変えた PSO_N の例を示す。図 4.3 (a) において、各サブ PSO は隣り合う 2 つの近傍サブ PSO と情報交換できることを表し (ring topology), 図 4.3 (b) において、各サブ PSO は隣り合う 4 つの近傍サブ PSO と情報交換できることを表す。図 4.3 (c) では、他の全てのサブ PSO と情報交換できることを表している (Fully connected topology)。総サブ PSO 数を S とするとき、 $DBG = S - 1$ とすれば、各サブ PSO は他の全てのサブ PSO が近傍サブ PSO となる。 DBG が小さい場合は良解情報の伝搬は遅くなり、各サブ PSO はそれぞれの良解情報周辺の探索を行うようになる。 DBG が大きい場合は良解情報が伝搬しやすくなり、多くのサブ PSO が同一の最良解情報周辺の探索を行うようになる。したがって、 DBG により局所探索能力と大域的探索能力を制御することができる。

単峰性問題に対しては DBG を大きな値に設定し、最良解周辺の局所探索を行うことが望ましく、逆に多峰性問題に対しては DBG を小さな値に設定し、解空間を大域的に探索することが望ましい。

図 4.3: PSO における DBG による近傍構造例

そのため、PSO において DBG は最適化問題の解空間の構造に合わせて適切に設定する必要がある。

PSO はマルチスウォーム PSO の一種であり、各サブ PSO は独立に計算を行うことが可能である。また、 $lbest$ のデータ量も小さいため、PSO を PC クラスタなどの並列計算環境に実装することは、1つのサブ PSO に1つのプロセッサを割り当て、近傍構造に基づいてプロセッサ間通信を行うことで実現できる。PSO では、複雑な近傍構造の構築や大量のデータを通信する必要がないため、プロセッサ間通信のボトルネックを小さくすることが期待できる。

$f(\mathbf{x})$ を最小化する問題における PSO の疑似コードを **Algorithm 4.1** に示す。ここで、 t_{max} は最大イテレーション数、 S は総サブ PSO 数、 N は各サブ PSO における粒子数、 D は次元数を表す。

Algorithm 4.1 PSQN の擬似コード

```

1:  $t = 0$ ;
2:  $DBG$  を初期化し,  $DBG$  に基づいて, 各サブ PSO の近傍構造を構築;
3: 全ての  $i$  に対して, 粒子  $i$  の  $\mathbf{v}_i^0, \mathbf{x}_i^0$  をランダムに初期化;
4: 全ての  $i$  に対して, 粒子  $i$  の評価値を  $f(\mathbf{x}_i^0)$  により計算;
5: 全ての  $i$  に対して, 粒子  $i$  の  $\mathbf{pbest}_i^0$  を自身の  $\mathbf{x}_i^0$  で初期化;
6: 全ての  $g$  に対して,  $PSO_g$  の  $\mathbf{lbest}_g^0$  を最良の  $\mathbf{pbest}_g^0$  で初期化;
7: 全ての  $g$  に対して,  $PSO_g$  の近傍と情報交換を行い  $\mathbf{glbest}_g^0$  を更新;
8: while  $t < t_{max}$  do
9:   for  $g = 1$  to  $S$  do
10:    for  $i = 1$  to  $N$  do
11:      for  $j = 1$  to  $D$  do
12:         $v_{g,ij}^t, x_{g,ij}^t$  を式 (4.1), (4.2) により更新;
13:      end for
14:      if  $f(\mathbf{x}_{g,i}^{t+1}) < f(\mathbf{pbest}_{g,i}^t)$  then
15:         $\mathbf{pbest}_{g,i}^{t+1} = \mathbf{x}_{g,i}^{t+1}$ ;
16:      end if
17:    end for
18:     $k = \arg \min_i f(\mathbf{pbest}_{g,i}^{t+1})$ ;
19:    if  $f(\mathbf{pbest}_{g,k}^{t+1}) < f(\mathbf{lbest}_g^t)$  then
20:       $\mathbf{lbest}_g^{t+1} = \mathbf{pbest}_{g,k}^{t+1}$ ;
21:    end if
22:  end for
23:  for  $g = 1$  to  $S$  do
24:    if  $\mathbf{lbest}_g^{t+1}$  が更新された場合 then
25:       $PSO_g$  の全ての近傍サブ PSO と通信;
26:       $PSO_g$  の近傍サブ PSO は式 (4.3) により  $\mathbf{glbest}$  を更新;
27:    end if
28:  end for
29:   $t = t + 1$ ;
30: end while

```

4.3 PSO における粒子の理論解析

4.3.1 粒子の収束性の解析

PSO において、粒子のダイナミクスは w , c_1 , c_2 , c_3 ($0 < c_1, c_2, c_3$) により制御されている。これらのパラメータにより、粒子の軌道は (1) 収束, (2) 発散, のいずれかの状態をとる。このため、粒子が安定して良解に収束するパラメータの設定指針を明らかにする必要がある。PSO では、**glbest** の項が追加されており、この影響を考慮する必要がある。PSO の粒子の安定性解析を簡単化するために乱数要素を排除した決定論的な PSO を用いる。PSO の粒子が安定して良解に収束するパラメータ条件を解析し、数値実験により解析結果の妥当性を確認する。特に、**glbest** の加速度係数 c_3 と加速度係数 c_1 , c_2 における粒子の安定条件を示す。

乱数要素を 1 に固定したとき ($r_1 = r_2 = r_3 = 1$)、PSO における PSO_g の粒子 i の j 次元要素における速度更新式は次式となる。

$$v_{g,ij}^{t+1} = wv_{g,ij}^t + \phi_1(pb_{g,ij}^t - x_{g,ij}^t) + \phi_2(lb_{g,j}^t - x_{g,ij}^t) + \phi_3(gl_{g,j}^t - x_{g,ij}^t) \quad (4.4)$$

ここで、 $\phi_x = c_x r_x$ を表す。さらに、 $pbest$, $lbest$, $glbest$ を理想的な固定点とし、その平衡点を G としたときの速度更新式を次に示す。

$$G_{g,ij} = \frac{\phi_1 pb_{g,ij} + \phi_2 lb_{g,j} + \phi_3 gl_{g,j}}{\phi_1 + \phi_2 + \phi_3} \quad (4.5)$$

$$\phi = \phi_1 + \phi_2 \quad (4.6)$$

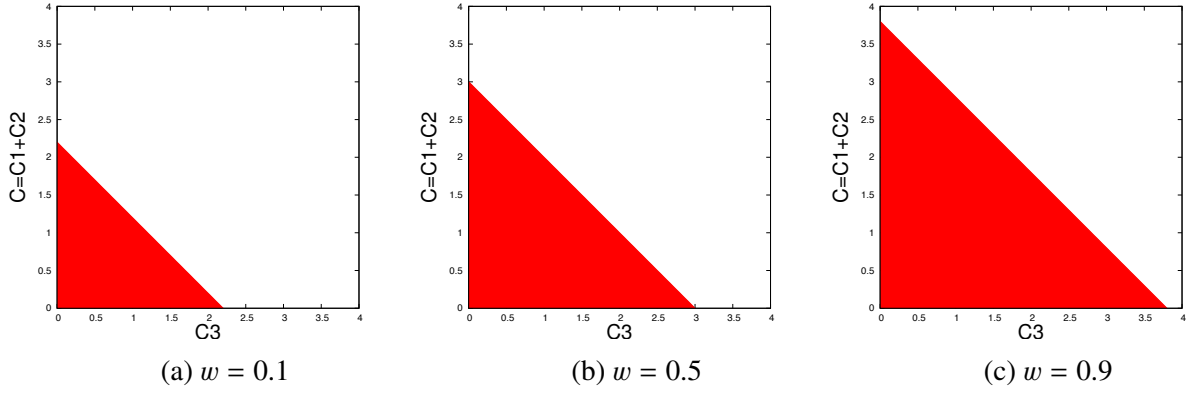
$$v_{g,ij}^{t+1} = wv_{g,ij}^t + (\phi + \phi_3)(G_{g,ij} - x_{g,ij}^t) \quad (4.7)$$

したがって、式 (4.7), (4.2) に基づいて更新される粒子のダイナミクスは決定論的な PSO として表される。本解析においては、解析を簡単化するために 1 次元の解空間を対象とする。PSO の各粒子の各次元要素は独立しているため、1 次元のみの解空間に注目したとしても一般性を失わない。ここで、粒子の位置 $x_{g,i}^t$ を平衡点 $G_{g,i}$ を基準とした相対位置 $y_{g,i}^t = x_{g,i}^t - G_{g,i}$ で表すと、式 (4.7) および相対位置 $y_{g,i}$ は次式で表すことができる。

$$v_{g,i}^{t+1} = wv_{g,i}^t - (\phi + \phi_3)y_{g,i}^t \quad (4.8)$$

$$y_{g,i}^{t+1} = wy_{g,i}^t + (1 - \phi - \phi_3)y_{g,i}^t \quad (4.9)$$

ここで、 $v_{g,i}^\infty = 0$ のとき、粒子 i は平衡点 $G_{g,i}$ に収束する。このため、粒子の収束、発散の解析を容易に行うことができる。


 図 4.4: w によるパラメータ領域

式 (4.8), (4.9) を行列形式で表すと式 (4.10) となる.

$$\begin{aligned}
 \begin{bmatrix} v_{g,i}^{t+1} \\ y_{g,i}^{t+1} \end{bmatrix} &= \begin{bmatrix} w & -(\phi + \phi_3) \\ w & 1 - \phi - \phi_3 \end{bmatrix} \begin{bmatrix} v_{g,i}^t \\ y_{g,i}^t \end{bmatrix} \\
 &= A \begin{bmatrix} v_{g,i}^t \\ y_{g,i}^t \end{bmatrix}
 \end{aligned} \tag{4.10}$$

行列 A の固有値 λ は次の式により計算される.

$$\lambda^2 - (1 - \phi - \phi_3 + w)\lambda + w = 0 \tag{4.11}$$

式 (4.11) より, 行列 A の固有値 λ は

$$\lambda = \frac{(1 - \phi - \phi_3 + w) \pm \sqrt{(1 - \phi - \phi_3 + w)^2 - 4w}}{2} \tag{4.12}$$

となる.

粒子が平衡点 G に収束するための条件は固有値 λ が複素平面上の単位円内に存在することである. よって, PSO における粒子の平衡点への収束条件は

$$\begin{cases} -\phi_3 < \phi < -\phi_3 + 2w + 2 \\ -1 < w < 1 \end{cases} \tag{4.13}$$

となる.

式 (4.13) より, $w = 0.1$, $w = 0.5$, $w = 0.9$ のときの粒子が安定して平衡点に収束する c , c_3 の領域を図 4.4 に示す. 図 4.4 において, 赤い領域は粒子が平衡点に安定して収束する領域を表す. また, 赤い領域外においては粒子は発散する.

4.3.2 安定性解析における数値実験

表 4.1: 実験環境

サブ PSO 数	1 または 2
粒子数/サブ PSO	1
w	0.729
初期位置	$[-5.12, 5.12]$ においてランダム配置
ベンチマーク問題	Rastrigin's
次元数 (D)	1
イタレーション数 (t_{max})	500

表 4.2: 安定性解析のパラメータセット

No.	c_3	c	解析結果
1	0.1955	2.9910	収束
2	0.8000	1.2000	収束
3	0.8000	1.6000	収束
4	0.3000	2.0000	収束
5	0.8000	2.7000	発散
6	2.6000	0.2000	収束
7	1.9955	1.9910	発散
8	1.8955	2.0910	発散
9	1.4955	2.9910	発散
10	1.0955	3.9910	発散
11	2.4955	2.9910	発散
12	4.0000	0.5000	発散
13	0.5000	4.0000	発散
14	3.0000	1.0000	発散
15	3.0000	0.5000	発散
16	3.5000	0.5000	発散
17	3.5000	0.2500	発散
18	5.0000	0.0000	発散
19	0.0000	5.0000	発散
20	3.0000	3.0000	発散
21	8.0000	0.0000	発散
22	0.0000	8.0000	発散
23	4.0000	4.0000	発散
24	5.0000	6.0000	発散

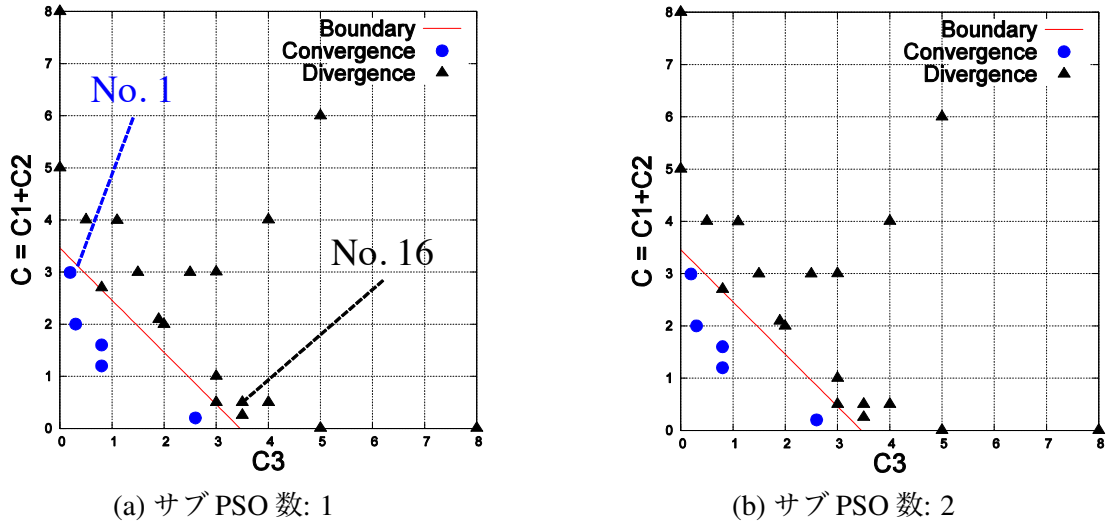


図 4.5: 安定性の実験結果

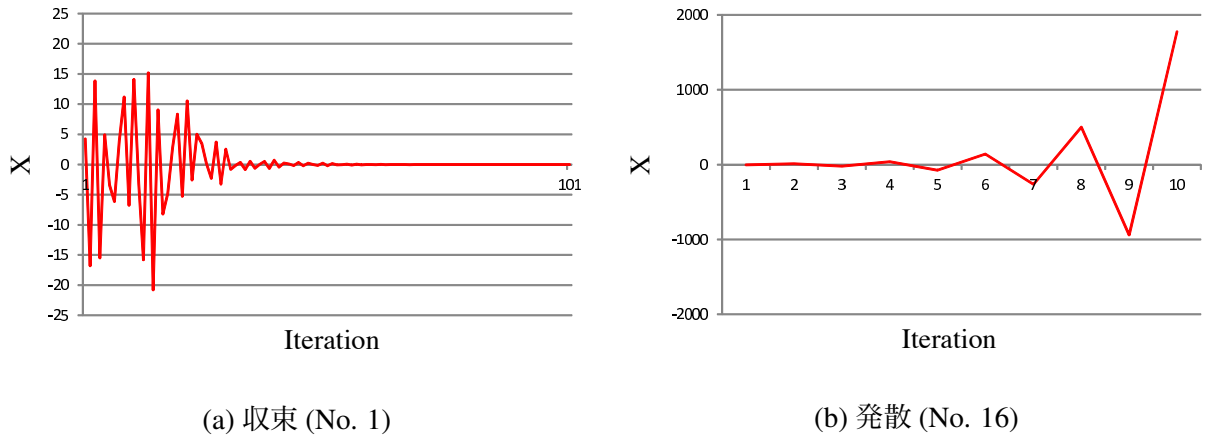


図 4.6: 1 つのサブ PSO における 1 粒子の推移

決定論的な PSO における解析的検討から得られた結果に基づいて加速度係数を収束，発散領域に設定し，サブ PSO 数が 1 と 2 の場合における粒子の状態を実験的に調査する．サブ PSO 数が 2 の場合はサブ PSO 間で情報交換を行う．ここで，収束は粒子が 1 つの平衡点に収束することを表し，発散はその軌道が平衡点から離れていくことを表す．表 4.1 に実験環境を，表 4.2 に本実験で用いるパラメータセットと式 (4.13) による解析結果をそれぞれ示す．表 4.2 における解析結果は式 (4.13) による解析結果に基づいている．

図 4.5 に (a) サブ PSO 数が 1，(b) サブ PSO 数が 2，のときの粒子の安定性の実験結果を示す．図 4.5 において，横軸は c_3 ，縦軸は $c = c_1 + c_2$ を表し，“No.” は表 4.2 の “No.” を表す．また，図 4.6 にサブ PSO 数が 1 の場合における収束 (No. 1)，発散 (No. 16) の 1 粒子における位置の推移を示す．図 4.6 において，横軸はイタレーション数，縦軸は粒子の位置を表す．図 4.5 より，サブ PSO 数が 1 つおよび 2 の場合において表 4.2 の解析結果と実験結果が一致していることが確認でき，サブ

PSO 数に関わらず解析結果が一致することも確認できる。また、図 4.6 では、解析結果において収束、発散するパラメータ設定における 1 粒子の推移を示しているが解析結果と一致していることが確認できる。以上のことから、解析結果はサブ PSO 数に関わらず妥当である。

4.4 数値実験

4.4.1 DBG の設定に関する検討

表 4.3: 実験環境 I

総サブ PSO 数 (S)	8	16
粒子数/サブ PSO (N)	20	10
次元数 (D)	30	
イタレーション数 (t_{max})	3000	
試行回数	100	
w	0.729	
c_1	1.4955	
c_2	1.4955	
c_3	0.3458	

表 4.4: ベンチマーク問題 I

	関数	探索範囲
Rastrigin's	$f(\mathbf{x}) = 10D + \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i))$	[-5.12, 5.12]
Rosenbrock's	$f(\mathbf{x}) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$	[-2.048, 2.048]
Griewank's	$f(\mathbf{x}) = 1 + \sum_{i=1}^D x_i^2 / 4000 - \prod_{i=1}^D (\cos(x_i / \sqrt{i}))$	[-600, 600]
Sphere	$f(\mathbf{x}) = \sum_{i=1}^D x_i^2$	[-5.12, 5.12]
Ackley's	$f(\mathbf{x}) = 20 - 20 \exp \left(-0.2 \sqrt{1/D \sum_{i=1}^D x_i^2} \right) + e - \exp \left(1 / \sum_{i=1}^D \cos(2\pi x_i) \right)$	[-32, 32]
Ridge	$f(\mathbf{x}) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	[-64, 64]
Schaffer's	$f(\mathbf{x}) = \sum_{i=1}^{D-1} (x_i^2 + x_{i+1}^2)^{0.25} \times (1 + \sin^2(50(x_i^2 + x_{i+1}^2)^{0.1}))$	[-100, 100]

PSO の有効性を示すために数値実験を行う。実験環境とベンチマーク問題を表 4.3, 表 4.4 に示す。また, CEC'13 ベンチマーク問題 [4.2] から 6 個のベンチマーク問題を選び, 実験で用いる全てのベンチマーク問題を表 4.5 に示す。表 4.5 において, $F_1 \sim F_5$ は単峰性問題, $F_6 \sim F_{13}$ は多峰性問題であり, “Separable” は設計変数間に依存関係のない問題, “Non-separable” は設計変数間

表 4.5: ベンチマーク問題 II

F	ベンチマーク関数	Landscape	最適解	探索範囲
F_1	Sphere	Separable unimodal	0	$[-5.12, 5.12]$
F_2	Shifted sphere (f_1) [4.2]	Separable unimodal	-1400	$[-100, 100]$
F_3	Different powers (f_5) [4.2]	Separable unimodal	-1000	$[-100, 100]$
F_4	Rosenbrock's	Non-separable unimodal	0	$[-2.048, 2.048]$
F_5	Ridge	Non-separable unimodal	0	$[-64, 64]$
F_6	Rastrigin's	Separable multimodal	0	$[-5.12, 5.12]$
F_7	Ackley's	Separable multimodal	0	$[-32, 32]$
F_8	Griewank's	Non-separable multimodal	0	$[-600, 600]$
F_9	Schaffer's	Non-separable multimodal	0	$[-100, 100]$
F_{10}	Rotated Griewank's (f_{10}) [4.2]	Non-separable multimodal	-500	$[-100, 100]$
F_{11}	Expanded Griewank's plus Rosenbrock's (f_{19}) [4.2]	Non-separable multimodal	500	$[-100, 100]$
F_{12}	Composition 1 (f_{21}) [4.2]	Non-separable multimodal	700	$[-100, 100]$
F_{13}	Composition 8 (f_{28}) [4.2]	Non-separable multimodal	1400	$[-100, 100]$

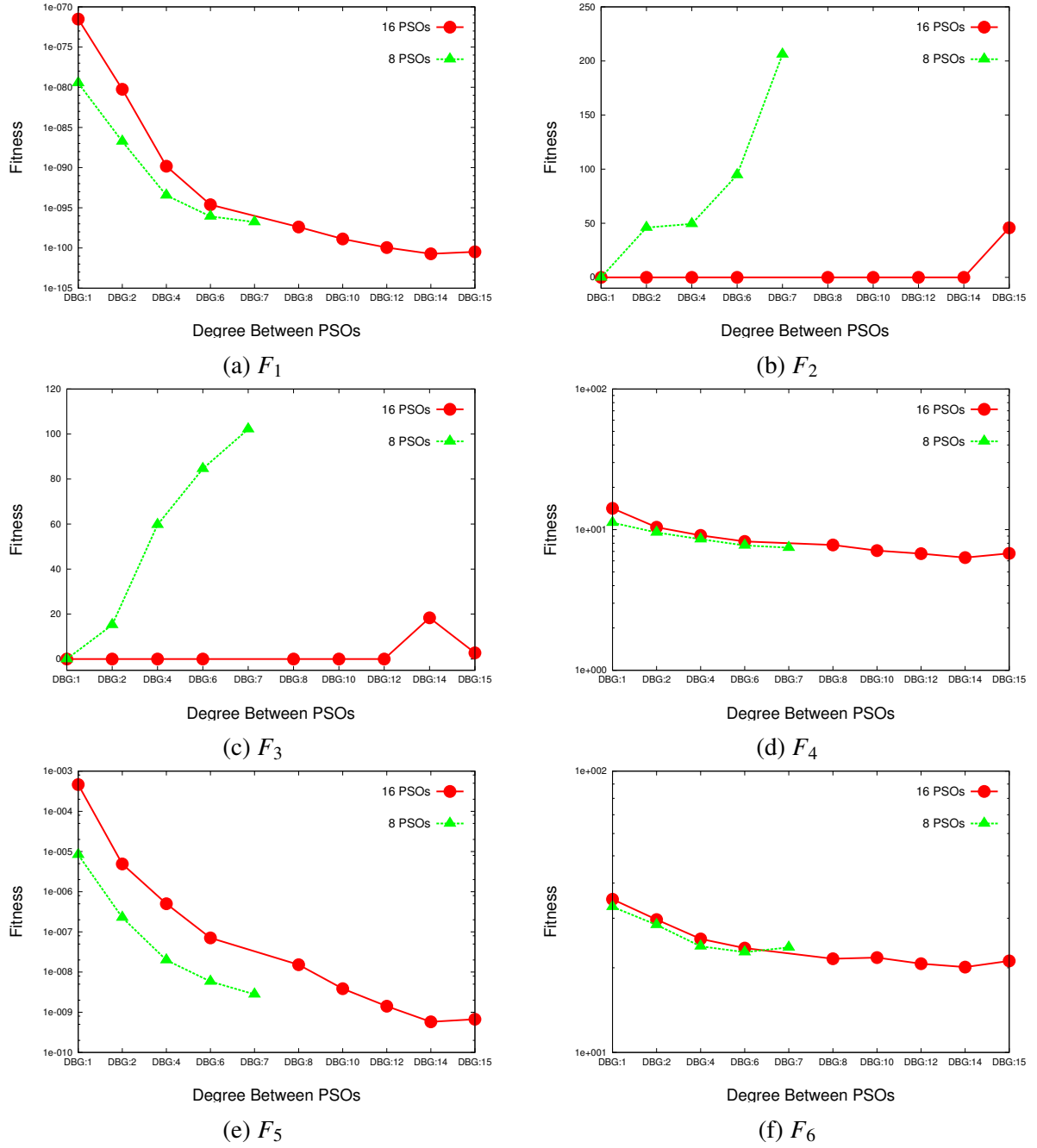
に依存関係のある問題を表す。PSO のパラメータについて、PSO における粒子の収束を保証するパラメータ設定方法 (式 (3.12)-(3.14)) に基づいて、その解探索性能が良好なパラメータ設定 (*i.e.*, $\varphi_1 = \varphi_2 = 2.05$) [4.1] と等価となる w , c_1 , c_2 を PSO のパラメータとして経験的に選んだ。 c_3 の値は式 (4.13) の収束条件を満たし、その固有値 λ が複素固有値となるように設定した。また、 c_3 の値はサブ PSO が探索初期に $glbest$ へ収束することを抑制するために c_1 , c_2 の値よりも小さな値に設定した。なお、 w , c_1 , c_2 , c_3 の値を式 (4.13) における収束領域と発散領域の境界に近づけることで、粒子は最良解周辺を大きく移動するため、広域的な探索を行うようになる。

総粒子数を 160 に固定し、総サブ PSO 数と各サブ PSO 内部の粒子数を変化させる。また、 DBG を変化させて解探索性能を調査する。実験結果は初期値をランダムに変化させた 100 回試行の平均値で示す。また、平均値は探索終了時の最良の $glbest$ と最適解の評価値の誤差値で示す。図 4.7, 図 4.8 に実験結果を示す。図 4.7, 図 4.8 において、横軸は DBG , 縦軸は探索終了時における最良の $glbest$ の評価値を表す。図 4.7, 図 4.8 より、PSO は 8 個のベンチマーク問題 ($F_1 \sim F_3$, F_5 , $F_7 \sim F_{10}$) において最適解を探索できている。また、全てのベンチマーク問題においてサブ PSO 数が 16 個 (16 PSO) の方が 8 個 (8 PSO) のときよりも良い解探索性能が得られていることがわかる。このことから、母集団を複数のサブ PSO に分割する場合、サブ PSO 内部の粒子数を増加させることよりもサブ PSO 数を増加させることの方が多様性や解探索性能を向上させるために有効であることがわかる。また、 DBG は PSO の解探索性能に大きな影響を与えており、解空間の構造に合わせて適切な値に設定する必要がある。

多峰性問題 (*i.e.*, $F_8 \sim F_{10}$, F_{12} , F_{13}) に対して、小さな DBG (Ring topology に近づける) に設定することで PSO の解探索性能が向上していくことが示されている。一方、 $F_1 \sim F_5$ の単峰性問題においては大きな DBG (Fully connected topology に近づける) に設定することで良好な結果を得て

いる。単峰性問題に対しては、発見した最良解の周辺の局所探索を十分に行うことが解探索に良い影響を与えるため DBG を大きく設定する必要がある。

上述した結果より、多峰性問題に対しては DBG を小さく設定し、単峰性問題に対しては DBG を大きく設定することで、精度の良い近似解を求めることができる。さらに、サブ PSO 数を増加させることは PSON の解探索性能に良い影響を与えることができる。


 図 4.7: PSO の解探索性能 ($F_1 \sim F_6$)

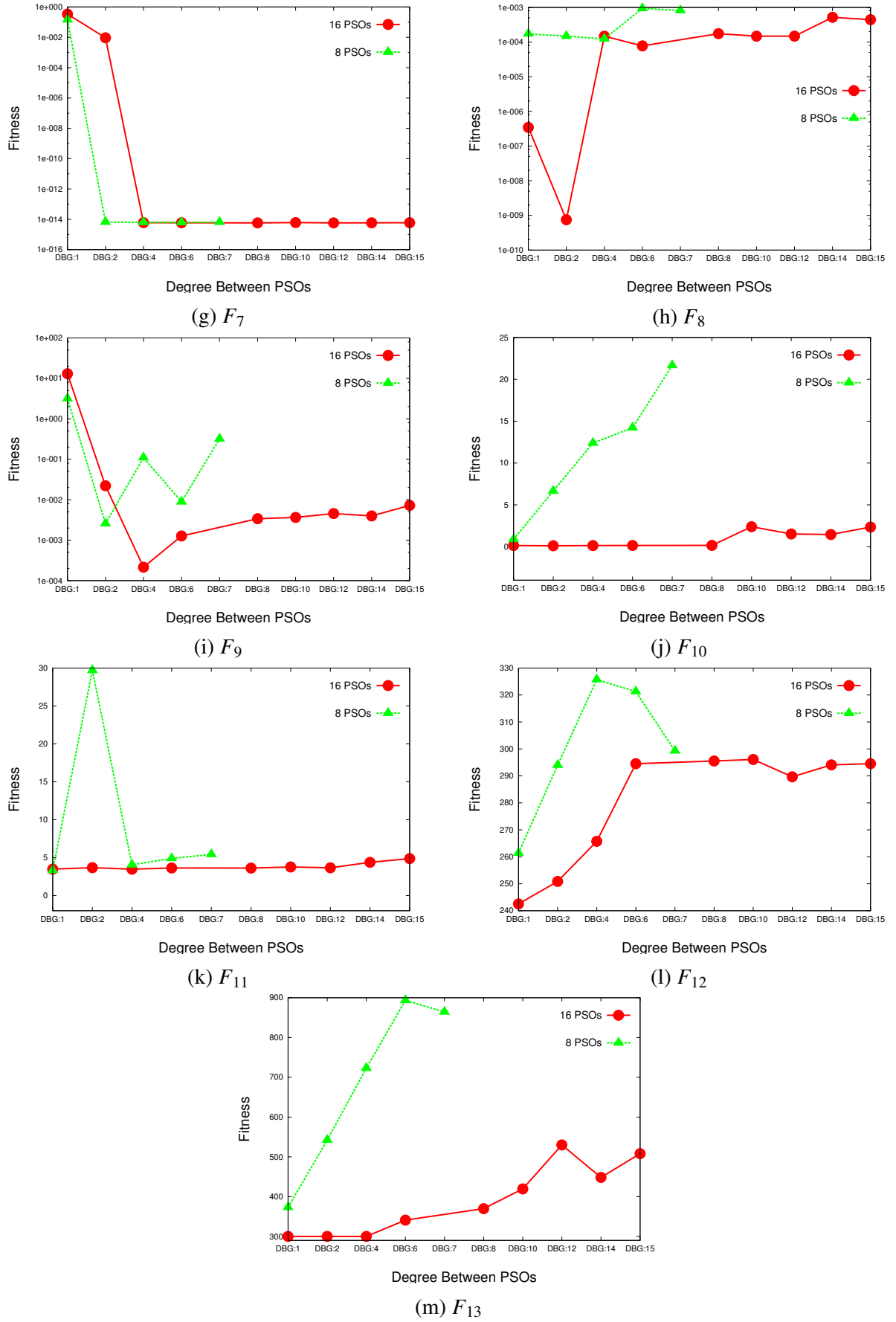


図 4.8: PSON の解探索性能 ($F_7 \sim F_{13}$)

4.4.2 解探索性能の比較実験

表 4.6: 実験環境 II

	PSON	他手法
次元数 (D)	30	
イタレーション数 (t_{max})	3000	
試行回数	100	
w	0.729	
c_1	1.4955	
c_2	1.4955	
c_3	0.3458	–

PSON の有効性を示すために様々な近傍構造を有する PSO, およびマルチスウォーム PSO と解探索性能を比較する. 比較対象として標準的な PSO [4.3], 静的な近傍構造を有する PSO (NPSO) [4.4], 動的な近傍構造を有する PSO (IPSO) [4.5], マルチスウォーム PSO である DMS-PSO [4.6] を選択した. NPSO は PSON と同様に粒子間における近傍構造が次数により静的に決定される. IPSO は近傍構造の構築が確率的に決定される動的な近傍構造である. DMS-PSO は定期的にサブ PSO の再グループ化が行われるマルチスウォーム PSO の一種である. PSON の解探索性能をこれらの手法との数値実験による比較から明らかにする. 総粒子数が 160 個の粒子数が多い場合と 30 個の粒子数が少ない場合の 2 つについて実験を行う. 実験環境は表 4.6 を用い, 表 4.5 のベンチマーク問題を用いる. 総粒子数が 160 個, 30 個において, 各手法のパラメータ候補をそれぞれ表 4.7, 表 4.8 に示し, 各手法において最良の解探索性能を示したパラメータの値をそれぞれ表 4.9, 表 4.10 に示す.

まず, 総粒子数を 160 とし, PSON, DMS-PSO においてはサブ PSO 数を 16, サブ PSO あたりの粒子数を 10 として実験を行う. 表 4.11 に実験結果を示す. 実験結果はランダムに初期値を変化させた 100 回試行における平均値 (Mean), 標準偏差 (SD), 中央値 (Median) により示す. 平均値, 中央値は探索終了時における最良解と最適解の評価値の誤差値で示す. また, 表 4.11 における平均値および標準偏差を用いて t 検定 [4.7] を行った. 有意水準は 5% としたときに PSON の結果が他の手法に有意性を持っているかを検定するものである. t 検定の結果を表 4.12 に示す. 表 4.12 には PSON と比較手法における “ p 値” (p -value; 有意確率) も示している. 表 4.12 において, 検定の結果 “+” は PSON よりも結果が優れているベンチマーク問題の個数を, “=” は PSON と同等の結果を示したベンチマーク問題の個数を, “–” は PSON が有意水準 5% で優れていると判定されたベンチマーク問題の個数を表す.

表 4.11, 表 4.12 より, PSON は多くのベンチマーク問題において他手法よりも優れた解探索性能を示している. PSON における各サブ PSO は独立して各々の解空間の探索を行うことができるため, 他手法よりも効率的に探索を行うことができる. NPSO や DMS-PSO も PSON と同様の構造を有していることから標準的な PSO よりも良好な解探索性能を示している. PSON, NPSO, DMS-PSO は

表 4.7: パラメータ候補 (160 粒子)

NPSO	<i>Degree</i>	{1, 2, 4, 8, 16, 32, 64}
IPSO	<i>C</i>	{0.01, 0.02, 0.05, 0.10, 0.20, 0.50}
DMS-PSO	<i>R</i>	{5, 6, 7}
PSO	<i>DBG</i>	{1, 2, 4, 6, 8, 10, 12, 14, 15}

表 4.8: パラメータ候補 (30 粒子)

NPSO	<i>Degree</i>	{1, 2, 4, 8, 16}
IPSO	<i>C</i>	{0.01, 0.02, 0.05, 0.10, 0.20, 0.50}
DMS-PSO	<i>R</i>	{5, 6, 7}
PSO	<i>DBG</i>	{1, 2, 4, 6, 8, 9}

表 4.9: 各手法におけるパラメータ (160 粒子)

<i>F</i>	NPSO	IPSO	DMS-PSO	PSO
F_1	<i>Degree</i> = 64	<i>C</i> = 0.20	<i>R</i> = 5	<i>DBG</i> = 14
F_2	<i>Degree</i> = 1	<i>C</i> = 0.01	<i>R</i> = 7	<i>DBG</i> = 1
F_3	<i>Degree</i> = 1	<i>C</i> = 0.01	<i>R</i> = 6	<i>DBG</i> = 1
F_4	<i>Degree</i> = 4	<i>C</i> = 0.10	<i>R</i> = 5	<i>DBG</i> = 14
F_5	<i>Degree</i> = 16	<i>C</i> = 0.05	<i>R</i> = 5	<i>DBG</i> = 14
F_6	<i>Degree</i> = 32	<i>C</i> = 0.05	<i>R</i> = 5	<i>DBG</i> = 14
F_7	<i>Degree</i> = 16	<i>C</i> = 0.05	<i>R</i> = 5	<i>DBG</i> = 8
F_8	<i>Degree</i> = 2	<i>C</i> = 0.10	<i>R</i> = 6	<i>DBG</i> = 2
F_9	<i>Degree</i> = 8	<i>C</i> = 0.02	<i>R</i> = 7	<i>DBG</i> = 4
F_{10}	<i>Degree</i> = 4	<i>C</i> = 0.02	<i>R</i> = 5	<i>DBG</i> = 2
F_{11}	<i>Degree</i> = 8	<i>C</i> = 0.05	<i>R</i> = 7	<i>DBG</i> = 4
F_{12}	<i>Degree</i> = 2	<i>C</i> = 0.05	<i>R</i> = 7	<i>DBG</i> = 1
F_{13}	<i>Degree</i> = 2	<i>C</i> = 0.01	<i>R</i> = 7	<i>DBG</i> = 2

表 4.10: 各手法におけるパラメータ (30 粒子)

<i>F</i>	NPSO	IPSO	DMS-PSO	PSO
F_1	<i>Degree</i> = 8	<i>C</i> = 0.10	<i>R</i> = 5	<i>DBG</i> = 4
F_2	<i>Degree</i> = 1	<i>C</i> = 0.01	<i>R</i> = 7	<i>DBG</i> = 1
F_3	<i>Degree</i> = 1	<i>C</i> = 0.01	<i>R</i> = 6	<i>DBG</i> = 1
F_4	<i>Degree</i> = 2	<i>C</i> = 0.05	<i>R</i> = 6	<i>DBG</i> = 4
F_5	<i>Degree</i> = 2	<i>C</i> = 0.05	<i>R</i> = 7	<i>DBG</i> = 4
F_6	<i>Degree</i> = 8	<i>C</i> = 0.10	<i>R</i> = 7	<i>DBG</i> = 9
F_7	<i>Degree</i> = 2	<i>C</i> = 0.02	<i>R</i> = 7	<i>DBG</i> = 4
F_8	<i>Degree</i> = 2	<i>C</i> = 0.05	<i>R</i> = 7	<i>DBG</i> = 2
F_9	<i>Degree</i> = 1	<i>C</i> = 0.05	<i>R</i> = 5	<i>DBG</i> = 6
F_{10}	<i>Degree</i> = 2	<i>C</i> = 0.05	<i>R</i> = 7	<i>DBG</i> = 2
F_{11}	<i>Degree</i> = 1	<i>C</i> = 0.05	<i>R</i> = 7	<i>DBG</i> = 2
F_{12}	<i>Degree</i> = 2	<i>C</i> = 0.05	<i>R</i> = 7	<i>DBG</i> = 4
F_{13}	<i>Degree</i> = 1	<i>C</i> = 0.02	<i>R</i> = 7	<i>DBG</i> = 2

$F_1 \sim F_3$, F_5 , $F_7 \sim F_{10}$ のベンチマーク問題において最適解を発見できている。その中でも PSO は平均値において NPSO や DMS-PSO よりも優れた結果を示していて、初期値に依存せず安定して最適解を得ることができている。

次に、総粒子数を 30 個として粒子数を 160 個から減少させたときの PSO の性能を調査する。PSO, DMS-PSO においては総サブ PSO 数を 10, サブ PSO 当たりの粒子数を 3 とする。表 4.13 に実験結果を示す。また、表 4.13 における平均値および標準偏差を用いて t 検定を行った。ここでも、有意水準は 5% として検定を行った結果を表 4.14 に示す。

表 4.13, 表 4.14 より、PSO は平均値において他手法と比較して良好な解探索性能を示している。また、表 4.11, 表 4.12, 表 4.13, 表 4.14 より、総粒子数が 160 個のときの方が 30 個のときよりも解探索性能が良好となることが確認できる。例えば、 F_9 において、PSO の総粒子数が 160 個のときは最適解を発見することができているが、30 個のときは最適解を発見することができていない。総粒子数の増加により解探索性能の向上が図られることがわかる。

表 4.11: 比較実験結果 (160 粒子)

F		PSO	NPSO	IPSO	DMS-PSO	PSON
F_1	Mean	2.62E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	2.61E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_2	Mean	1.02E+04	0.00E+00	4.87E-05	4.02E+01	0.00E+00
	SD	4.90E+03	0.00E+00	3.49E-05	1.82E+02	0.00E+00
	Median	9.85E+03	0.00E+00	3.95E-05	0.00E+00	0.00E+00
F_3	Mean	1.92E+03	0.00E+00	1.47E+01	2.19E+01	0.00E+00
	SD	1.21E+03	0.00E+00	3.85E+01	5.50E+01	0.00E+00
	Median	2.06E+03	0.00E+00	3.64E-02	0.00E+00	0.00E+00
F_4	Mean	4.45E+01	1.43E+01	2.22E+01	1.93E+01	6.33E+00
	SD	1.16E+02	5.96E+00	1.15E+01	7.66E+00	1.77E+00
	Median	1.36E+01	1.59E+01	2.09E+01	1.80E+01	6.24E+00
F_5	Mean	3.37E+03	2.13E-03	1.51E+00	7.61E+01	0.00E+00
	SD	2.57E+03	1.98E-03	1.22E+00	3.98E+02	0.00E+00
	Median	2.73E+03	1.60E-03	1.07E+00	4.77E-03	0.00E+00
F_6	Mean	9.37E+01	5.49E+01	2.95E+01	3.78E+01	2.01E+01
	SD	2.96E+01	1.68E+01	9.57E+00	1.34E+01	8.07E+00
	Median	9.11E+01	5.38E+01	2.91E+01	3.48E+01	1.89E+01
F_7	Mean	1.67E+00	0.00E+00	8.06E-02	0.00E+00	0.00E+00
	SD	4.60E+00	0.00E+00	2.95E-01	0.00E+00	0.00E+00
	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_8	Mean	2.72E+00	0.00E+00	5.59E-03	3.20E-03	0.00E+00
	SD	1.54E+01	0.00E+00	8.06E-03	5.36E-03	0.00E+00
	Median	8.63E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_9	Mean	7.55E+01	1.80E-02	1.44E+01	1.85E+00	2.15E-04
	SD	2.91E+01	8.89E-02	1.09E+01	5.39E+00	1.50E-03
	Median	7.54E+01	4.21E-05	1.17E+01	0.00E+00	0.00E+00
F_{10}	Mean	1.39E+03	3.02E+01	5.27E+01	1.10E+02	1.19E-01
	SD	6.68E+02	4.38E+01	6.81E+01	1.42E+02	4.71E-02
	Median	1.29E+03	1.77E+01	2.26E+01	6.48E+01	1.19E-01
F_{11}	Mean	7.55E+03	3.93E+00	8.32E+00	3.74E+00	3.48E+00
	SD	1.03E+04	1.38E+00	4.43E+00	3.05E+00	7.97E-01
	Median	3.36E+03	3.78E+00	7.07E+00	3.02E+00	3.38E+00
F_{12}	Mean	1.10E+03	2.14E+02	2.93E+02	3.06E+02	2.43E+02
	SD	4.79E+02	3.47E+01	8.15E+01	1.23E+02	5.53E+01
	Median	1.09E+03	2.00E+02	3.00E+02	3.00E+02	2.05E+02
F_{13}	Mean	2.63E+03	3.00E+02	4.71E+02	8.72E+02	3.00E+02
	SD	4.34E+02	0.00E+00	3.73E+02	4.99E+02	0.00E+00
	Median	2.59E+03	3.00E+02	3.07E+02	8.12E+02	3.00E+02

* 太字は最良の誤差値を示す

表 4.12: PSQN vs 比較手法における t 検定の結果 (160 粒子)

F		PSO	NPSO	IPSO	DMS-PSO	PSQN
F_1	Mean±SD	2.62E-01±2.61E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00
	p -value	3.20E-01	–	–	–	–
F_2	Mean±SD	1.02E+04±4.90E+03	0.00E+00±0.00E+00	4.87E-05±3.49E-05	4.02E+01±1.82E+02	0.00E+00±0.00E+00
	p -value	2.20E-16*	–	2.20E-16*	3.05E-02*	–
F_3	Mean±SD	1.92E+03±1.21E+03	0.00E+00±0.00E+00	1.47E+01±3.85E+01	2.19E+01±5.50E+01	0.00E+00±0.00E+00
	p -value	2.20E-16*	–	2.20E-16*	1.43E-04*	–
F_4	Mean±SD	4.45E+01±1.16E+02	1.43E+01±5.96E+00	2.22E+01±1.15E+01	1.93E+01±7.66E+00	6.33E+00±1.77E+00
	p -value	1.43E-03*	2.20E-16*	2.20E-16*	2.20E-16*	–
F_5	Mean±SD	3.37E+03±2.57E+03	2.13E-03±1.98E-03	1.51E+00±1.22E+00	7.61E+01±3.98E+02	0.00E+00±0.00E+00
	p -value	2.20E-16*	2.20E-16*	2.20E-16*	6.04E-02	–
F_6	Mean±SD	9.37E+01±2.96E+01	5.49E+01±1.68E+01	2.95E+01±9.57E+00	3.78E+01±1.34E+01	2.01E+01±8.07E+00
	p -value	2.20E-16*	2.20E-16*	3.20E-12*	2.20E-16*	–
F_7	Mean±SD	1.67E+00±4.60E+00	0.00E+00±0.00E+00	8.06E-02±2.95E-01	0.00E+00±0.00E+00	0.00E+00±0.00E+00
	p -value	4.71E-04*	–	7.70E-03*	–	–
F_8	Mean±SD	2.72E+00±1.54E+01	0.00E+00±0.00E+00	5.59E-03±8.06E-03	3.20E-03±5.36E-03	0.00E+00±0.00E+00
	p -value	8.17E-02	–	4.97E-10*	4.09E-08*	–
F_9	Mean±SD	7.55E+01±2.91E+01	1.80E-02±8.89E-02	1.44E+01±1.09E+01	1.85E+00±5.39E+00	2.15E-04±1.50E-03
	p -value	2.20E-16*	4.90E-02*	2.20E-16*	9.30E-04*	–
F_{10}	Mean±SD	1.39E+03±6.68E+02	3.02E+01±4.38E+01	5.27E+01±6.81E+01	1.10E+02±1.42E+02	1.19E-01±4.71E-02
	p -value	2.20E-16*	6.51E-10*	1.13E-11*	1.23E-11*	–
F_{11}	Mean±SD	7.55E+03±1.03E+04	3.93E+00±1.38E+00	8.32E+00±4.43E+00	3.74E+00±3.05E+00	3.48E+00±7.97E-01
	p -value	7.77E-11*	5.03E-03*	2.20E-16*	4.02E-01	–
F_{12}	Mean±SD	1.10E+03±4.79E+02	2.14E+02±3.47E+01	2.93E+02±8.15E+01	3.06E+02±1.23E+02	2.43E+02±5.53E+01
	p -value	2.20E-16*	2.37E-05*	8.98E-07*	6.04E-06*	–
F_{13}	Mean±SD	2.63E+03±4.34E+02	3.00E+02±0.00E+00	4.71E+02±3.73E+02	8.72E+02±4.99E+02	3.00E+02±0.00E+00
	p -value	2.20E-16*	–	1.46E-05*	2.20E-16*	–
+		0	1	0	0	–
=		2	6	1	4	–
–		11	6	12	9	–

太字は最良の平均値を示し, * は PSQN に対して有意差があることを示す

表 4.13: 比較実験結果 (30 粒子)

F		PSO	NPSO	IPSO	DMS-PSO	PSON
F_1	Mean	5.24E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	1.11E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Median	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_2	Mean	2.27E+04	9.38E+00	1.46E+02	3.14E+02	0.00E+00
	SD	7.98E+03	9.33E+01	4.00E+02	5.97E+02	0.00E+00
	Median	2.09E+04	0.00E+00	1.99E+00	0.00E+00	0.00E+00
F_3	Mean	7.91E+03	4.62E-01	5.25E+02	5.88E+02	4.62E-01
	SD	5.56E+03	4.60E+00	7.41E+02	1.01E+03	4.60E+00
	Median	6.32E+03	0.00E+00	1.86E+02	1.88E+02	0.00E+00
F_4	Mean	1.19E+02	2.21E+01	3.45E+01	2.58E+01	2.05E+01
	SD	2.59E+02	6.67E+00	2.37E+01	2.35E+01	7.41E+00
	Median	1.70E+01	2.31E+01	2.48E+01	2.13E+01	1.94E+01
F_5	Mean	5.32E+03	1.27E+02	3.39E+02	1.16E+03	2.33E-03
	SD	3.70E+03	3.86E+02	6.08E+02	1.64E+03	3.62E-03
	Median	4.78E+03	3.60E+01	1.37E+02	3.14E+00	1.32E-03
F_6	Mean	1.33E+02	1.00E+02	6.24E+01	7.18E+01	3.83E+01
	SD	3.72E+01	2.73E+01	2.04E+01	2.23E+01	1.13E+01
	Median	1.28E+02	1.02E+02	5.87E+01	6.96E+01	3.71E+01
F_7	Mean	7.84E+00	1.30E+00	3.28E+00	6.26E+00	1.50E+00
	SD	6.77E+00	3.72E+00	3.87E+00	9.09E+00	2.04E+00
	Median	3.46E+00	0.00E+00	2.41E+00	0.00E+00	1.34E+00
F_8	Mean	2.29E+01	2.40E-03	1.41E-02	6.77E-03	4.34E-03
	SD	3.39E+01	4.53E-03	1.77E-02	9.40E-03	8.71E-03
	Median	7.18E+00	0.00E+00	9.86E-03	0.00E+00	2.33E-04
F_9	Mean	1.22E+02	7.82E+00	4.99E+01	2.71E+01	5.03E+00
	SD	2.65E+01	4.03E+00	2.42E+01	1.97E+01	4.35E+00
	Median	1.21E+02	6.47E+00	4.67E+01	2.52E+01	4.09E+00
F_{10}	Mean	3.23E+03	1.77E+02	1.99E+02	2.61E+02	5.20E-01
	SD	1.07E+03	1.75E+02	2.48E+02	2.72E+02	2.27E+00
	Median	3.13E+03	1.35E+02	1.26E+02	1.83E+02	2.69E-01
F_{11}	Mean	1.23E+05	1.66E+01	3.94E+02	3.76E+02	1.00E+01
	SD	1.81E+05	3.12E+01	1.47E+03	1.65E+03	2.53E+00
	Median	5.77E+04	1.35E+01	2.45E+01	8.27E+00	9.52E+00
F_{12}	Mean	2.56E+03	3.19E+02	3.69E+02	3.92E+02	3.02E+02
	SD	4.20E+02	1.58E+02	2.33E+02	2.26E+02	8.56E+01
	Median	2.53E+03	3.00E+02	3.00E+02	3.00E+02	3.00E+02
F_{13}	Mean	4.13E+03	6.11E+02	1.45E+03	1.39E+03	4.33E+02
	SD	7.76E+02	5.46E+02	6.41E+02	6.73E+02	3.96E+02
	Median	3.97E+03	3.00E+02	1.62E+03	1.53E+03	3.00E+02

* 太字は最良の誤差値を示す

表 4.14: PSON vs 比較手法における t 検定の結果 (30 粒子)

F		PSO	NPSO	IPSO	DMS-PSO	PSON
F_1	Mean±SD	5.24E+00±1.11E+01	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00
	p -value	8.74E-06*	–	–	–	–
F_2	Mean±SD	2.27E+04±7.98E+03	9.38E+00±9.33E+01	1.46E+02±4.00E+02	3.14E+02±5.97E+02	0.00E+00±0.00E+00
	p -value	2.20E-16*	3.20E-01	4.71E-04*	9.35E-07*	–
F_3	Mean±SD	7.91E+03±5.56E+03	4.62E-01±4.60E+00	5.25E+02±7.41E+02	5.88E+02±1.01E+03	4.62E-01±4.60E+00
	p -value	2.20E-16*	1.00E+00	2.53E-10*	8.30E-08*	–
F_4	Mean±SD	1.19E+02±2.59E+02	2.21E+01±6.67E+00	3.45E+01±2.37E+01	2.58E+01±2.35E+01	2.05E+01±7.41E+00
	p -value	2.55E-04*	1.14E-01	1.45E-07*	3.63E-02*	–
F_5	Mean±SD	5.32E+03±3.70E+03	1.27E+02±3.86E+02	3.39E+02±6.08E+02	1.16E+03±1.64E+03	2.33E-03±3.62E-03
	p -value	2.20E-16*	1.49E-03*	2.42E-07*	2.43E-10*	–
F_6	Mean±SD	1.33E+02±3.72E+01	1.00E+02±2.73E+01	6.24E+01±2.04E+01	7.18E+01±2.23E+01	3.83E+01±1.13E+01
	p -value	2.20E-16*	2.20E-16*	2.20E-16*	2.20E-16*	–
F_7	Mean±SD	7.84E+00±6.77E+00	1.30E+00±3.72E+00	3.28E+00±3.87E+00	6.26E+00±9.09E+00	1.50E+00±2.04E+00
	p -value	7.70E-15*	6.41E-01	8.48E-05*	1.58E-06*	–
F_8	Mean±SD	2.29E+01±3.39E+01	2.40E-03±4.53E-03	1.41E-02±1.77E-02	6.77E-03±9.40E-03	4.34E-03±8.71E-03
	p -value	2.38E-06*	5.05E-02	2.47E-06*	6.09E-02	–
F_9	Mean±SD	1.22E+02±2.65E+01	7.82E+00±4.03E+00	4.99E+01±2.42E+01	2.71E+01±1.97E+01	5.03E+00±4.35E+00
	p -value	2.20E-16*	5.29E-06*	2.20E-16*	2.20E-16*	–
F_{10}	Mean±SD	3.23E+03±1.07E+03	1.77E+02±1.75E+02	1.99E+02±2.48E+02	2.61E+02±2.72E+02	5.20E-01±2.27E+00
	p -value	2.20E-16*	2.20E-16*	3.13E-12*	1.14E-15*	–
F_{11}	Mean±SD	1.23E+05±1.81E+05	1.66E+01±3.12E+01	3.94E+02±1.47E+03	3.76E+02±1.65E+03	1.00E+01±2.53E+00
	p -value	5.28E-06*	3.85E-02*	1.08E-02*	2.99E-02*	–
F_{12}	Mean±SD	2.56E+03±4.20E+02	3.19E+02±1.58E+02	3.69E+02±2.33E+02	3.92E+02±2.26E+02	3.02E+02±8.56E+01
	p -value	2.20E-16*	3.41E-01	8.81E-03*	3.27E-04*	–
F_{13}	Mean±SD	4.13E+03±7.76E+02	6.11E+02±5.46E+02	1.45E+03±6.41E+02	1.39E+03±6.73E+02	4.33E+02±3.96E+02
	p -value	2.20E-16*	9.41E-03*	2.20E-16*	2.20E-16*	–
+		0	0	0	0	–
=		0	7	1	2	–
–		13	6	12	11	–

太字は最良の平均値を示し, * は PSON に対して有意差があることを示す

4.5 4章の結び

本章では、並列計算環境への実装を考慮し、PSO の解探索性能を向上させるための手法 (PSON) を提案し、その有効性を明らかにした。

まず、PSON の粒子の収束条件を解析し、数値解析によりその妥当性を確認した。PSON では、DBG によりサブ PSO 間の近傍構造を制御することができ、最良解情報の伝播速度を制御することができる。多峰性問題においては DBG を小さな値に設定することで、単峰性問題においては DBG を大きな値に設定することで、それぞれ高精度な解探索ができることを明らかにした。

総粒子数を一定としたときに個々のサブ PSO における粒子数を増加させることよりもサブ PSO 数を増加させることの方が解探索性能に良い影響を与えることを明らかにした。さらに、様々な近傍構造を有する PSO 手法やマルチスウォーム PSO との解探索性能の比較を行った結果、PSON がそれらの手法よりも解探索性能が良好であることを明らかにした。

マルチプロセッサに PSON アルゴリズムを実装する場合、1つのサブ PSO に対して1つのプロセッサを割り当てることで実現できる。各サブ PSO は固定的な近傍構造でネットワーク化されているため、通信するプロセッサを予め指定でき、通信するデータは *lbest* のみであり、そのデータ量が少ないことから通信におけるボトルネックも小さくなることが期待される。つまり、PSON は並列計算による計算時間の短縮が可能なネットワーク構造となっている。ただし、 N 台のプロセッサに対して M 個のサブ PSO ($N < M$) がある場合が考えられる。このような場合においては、サブ PSO の総数を N 個に減らし、各サブ PSO の内部の粒子数をその分増加させることが考えられる。しかしながら、プロセッサの個数がサブ PSO の個数よりも極端に少ない場合は PSON の性能を維持するために、各プロセッサに均等にサブ PSO を割り当てることが考えられる。

今後、PSON の解探索性能の向上やマルチプロセッサ環境へ PSON アルゴリズムを実装し、計算時間の短縮や大規模な実最適化問題を解くことなどを行う必要がある。

参考文献

- [4.1] M. Clerc and J. Kennedy, "The particle swarm -Explosion, stability, and convergence in a multi-dimensional complex space," *IEEE Trans. on Evolutionary Computation*, Vol. 6, No.2, pp. 58-73, 2002.
- [4.2] J. J. Liang, B. Y. Qu, P. N. Suganthan and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," *Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou, China Nanyang Technol. Univ. Singapore, Tech. Rep.*, Vol. 201212, 2013.
- [4.3] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. of IEEE Int. Conf. evolutionary computation*, pp. 69-73, 1998.
- [4.4] J. Kennedy, "Small worlds and mega-minds : effects of neighborhood topology on particle swarm performance," in *Proc. of IEEE CEC*, pp. 1931-1938, 1999.
- [4.5] H. Matsushita, Y. Nishio and T. Saito, "Particle swarm optimization with novel concept of complex network," in *Proc. of NOLTA*, pp. 197-200, 2010.
- [4.6] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proc. of IEEE SIS*, pp. 124-129, 2005.
- [4.7] B. Welch, "The generalization of 'Student's' problem when several different population variances are involved," *Biometrika*, Vol. 34, pp. 28-35, 1947.

第五章 多様性向上のための粒子群最適化

ネットワークに対する確率的な ネットワーク化手法

5 多様性向上のための粒子群最適化ネットワークに対する確率的なネットワーク化手法

5.1 5章の概要

4章では、PSONを提案し、その有効性を明らかにした。PSONでは、サブPSO間において近傍構造を持ち、その近傍構造を制御することで多様性の向上が実現される。しかしながら、難易度の高い多峰性問題においては、その多様性でも対処することができず局所解に収束することがある。この問題はPSONの近傍構造が固定的であり、ネットワークが静的であることから、各サブPSOは $pbest$ 、 $lbest$ 、 $gbest$ で決定される探索領域内にその探索範囲が限定される。仮にその探索範囲に最適解が含まれない場合は最適解を発見することができない。

3章では、動的な近傍構造を導入したPSO手法の多くが静的な近傍構造を有するPSO手法よりも様々なベンチマーク問題に対して良好な解探索性能を持つことを説明した。そこで、多峰性問題に対して良好な近似解を発見するためにPSONに動的な近傍構造を導入した手法を本章で提案する。

本章では、PSONの多様性を向上させるために、PSONの探索過程において動的に変化する確率的な近傍構造を導入した粒子群最適化ネットワーク (PSO Networks with Stochastic Connection; PSON-SC) を提案する。

PSON-SCでは、PSONと同様に母集団を複数のサブPSOに分割し、各サブPSOは独立に解空間の探索を行う。各サブPSOはサブPSO間における最良解情報($gbest$)を有するメモリ($gbest$ information memory; $gbest$ 情報メモリ)と確率的に通信し、 $gbest$ 情報の参照や更新を行うことができる。各サブPSOは局所探索モードと大域探索モードの2つの探索モードを持ち、探索モードに基づいてサブPSOに属する粒子は参照する最良解情報を変更する。局所探索モードにおいて、サブPSOに属する粒子はそのサブPSOにおける最良解情報($lbest$)を参照し、大域探索モードにおいて、サブPSOに属する粒子は $gbest$ 情報メモリ($gbest$)を参照し、解空間の探索を行う。

各イタレーションにおいて、全てのサブPSOの探索モードは確率的に決定し、大域探索モードとなった場合は $gbest$ の更新を行う。また、局所探索モードとなった場合はそのサブPSOの $lbest$ が $gbest$ よりも良好な解情報であったとしても、 $gbest$ の更新に関与することができない。PSON-SCでは、大域探索モードになる確率が低い場合は各サブPSOは $gbest$ 情報メモリと通信する確率が低いことを意味していて、各サブPSOの独立性は向上しそれぞれの解空間の探索を行うようになる。サブPSOが局所解に陥り脱出が困難となっても、低い確率ではあるが大域探索モードになったときに $gbest$ 情報メモリと通信することでその局所解からの脱出ができる。各サブPSOは $gbest$ 情報メモリとのみ確率的に通信するため、イタレーションごとに近傍構造が変化し、PSON-SCの多様性は向上する。

次節以降では、PSON-SCの説明を行い、その有効性を確認するために数値実験を行う。まず、PSON-SCの多様性がPSONよりも高いことを示すためにPSONとの多様性の比較を行う。次に、

PSON-SC の有効性を示すために多数のベンチマーク問題を用いて PSON を含む様々な近傍構造を有する PSO との解探索性能の比較を行う.

5.2 確率的結合を有する粒子群最適化ネットワーク

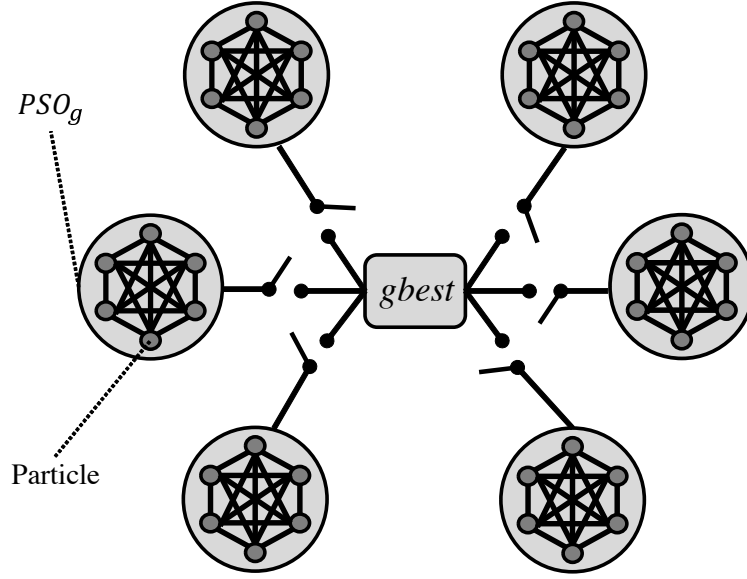


図 5.1: PSO-SC の近傍構造例

本節では、確率的な結合を有する粒子群最適化ネットワーク (PSO Networks with Stochastic Connection; PSO-SC) を提案し、その説明を行う。PSO-SC においては、PSO と同様に粒子群の母集団を複数の小規模なサブ集団 (サブ PSO) に分割し、各サブ PSO はサブ PSO 間における最良解情報を保持するメモリ (gbest information memory; gbest 情報メモリ) と確率的に結合可能な近傍構造を構築する。各サブ PSO は gbest 情報メモリとのみ結合できる wheel topology (図 3.6 参照) と同様の近傍構造を有する。図 5.1 にサブ PSO 数が 6 個 (1 個のサブ PSO に 6 個の粒子) から構成される PSO-SC の例を示す。図 5.1 において、各サブ PSO は独立して探索を行い、確率的に gbest 情報メモリと通信を行う。 g 番目のサブ PSO (PSO_g) の i 番目の粒子 (粒子 i) は、速度ベクトル $\mathbf{v}_{g,i} = (v_{g,i1}, v_{g,i2}, \dots, v_{g,iD})$ 、位置ベクトル $\mathbf{x}_{g,i} = (x_{g,i1}, x_{g,i2}, \dots, x_{g,iD})$ 、自身の探索過程における最良解ベクトル $\mathbf{pbest}_{g,i} = (pb_{g,i1}, pb_{g,i2}, \dots, pb_{g,iD})$ を持ち、 PSO_g は属する粒子群の探索過程における最良解ベクトル $\mathbf{lbest}_g = (lb_{g,1}, lb_{g,2}, \dots, lb_{g,D})$ を持つ。また、gbest 情報メモリはサブ PSO 間における最良解ベクトル $\mathbf{gbest} = (gb_1, gb_2, \dots, gb_D)$ を保持する。 D は最適化する設計変数の総数を表す。

PSO-SC では、各サブ PSO は局所探索モード (Local search mode) と大域探索モード (Global search mode) の 2 つの探索モードを持つ。 PSO_g において、 \mathbf{lbest}_g を参照する局所探索モードと gbest 情報メモリと通信し、 \mathbf{lbest}_g の代わりに \mathbf{gbest} を参照する大域探索モードがあり、大域探索モードに設定される確率は切り替え確率 (Switching Rate; SR) により決定される。局所探索モードにおいて、サブ PSO は gbest 情報メモリと通信せず、独立して解空間の探索を行う。また、局所探索モードである PSO_g における粒子 i は $\mathbf{pbest}_{g,i}$ 、 \mathbf{lbest}_g を参照し、速度と位置の更新を行う。一方、大域探索モードにおいては、サブ PSO は gbest 情報メモリと通信し、 \mathbf{gbest} を参照して更新を行う。すなわ

ち、 g 番目のサブ PSO (PSO_g) の粒子 i は $\mathbf{pbest}_{g,i}$, \mathbf{gbest} を参照し、速度と位置を更新する。 PSO_g における 2 つの探索モードの切り替え方法を次に示す。

$$\begin{cases} \text{Global search mode,} & \text{if } \text{rand}_g \leq SR \\ \text{Local search mode,} & \text{otherwise} \end{cases} \quad (5.1)$$

ここで、 SR ($0 \leq SR \leq 1$) は大域探索モードへの切り替え確率、 rand_g は $[0, 1]$ の一様乱数を表す。各サブ PSO はそれぞれの探索モードをイタレーションごとに式 (5.1) により決定する。 PSO_g における粒子 i の j 次元要素の速度と位置の更新式を次に示す。

$$v_{g,ij}^{t+1} = \begin{cases} wv_{g,ij}^t + c_1r_1(\mathbf{pb}_{g,ij}^t - x_{g,ij}^t) + c_2r_2(\mathbf{gb}_j^t - x_{g,ij}^t), & \text{if Global search mode} \\ wv_{g,ij}^t + c_1r_1(\mathbf{pb}_{g,ij}^t - x_{g,ij}^t) + c_2r_2(\mathbf{lb}_{g,j}^t - x_{g,ij}^t), & \text{if Local search mode} \end{cases} \quad (5.2)$$

$$x_{g,ij}^{t+1} = x_{g,ij}^t + v_{g,ij}^{t+1} \quad (5.3)$$

ここで、 w は慣性速度定数、 c_1 , c_2 は加速度係数、 r_1 , r_2 は $[0, 1]$ の一様乱数を表す。

式 (5.2) において、 SR が大きい (1 に近い) 場合、各サブ PSO は \mathbf{gbest} 情報メモリとの通信が促進され、 \mathbf{gbest} 周辺の探索が促進されるため、多くのサブ PSO により \mathbf{gbest} 周辺の探索が行われる。特に、 $SR = 1.0$ の場合は静的な近傍構造となる。一方、 SR が小さい (0 に近い) 場合、 PSO_g は \mathbf{lbest}_g 周辺の探索が促進される。 \mathbf{gbest} 情報メモリとの通信が抑制されるため、各サブ PSO は独立して解空間を探索するようになり、解空間を大域的に探索するようになる。特に、 $SR = 0.0$ の場合は各サブ PSO は独立して各々の解空間の探索を行う。したがって、 SR は PSO-SC の多様性を制御する重要なパラメータであり、 SR を大きくすることで多様性は失われ、解収束速度が向上し、逆に SR を小さくすることで多様性は向上し、解収束が遅く、最適解への収束可能性が向上する。このため、PSO-SC において SR は PSO における DBG と同様に最適化問題の解空間の構造に合わせて適切に設定する必要がある。

図 5.2 に PSO_g における粒子 i の j 次元要素の探索領域の例を示す。図 5.2 において、紫の領域は大域探索モードにおける探索領域を表し、緑の領域は局所探索モードにおける探索領域を表す。つまり、探索モードが切り替わると粒子の探索領域はドラスチックに変化する。このため、サブ PSO が局所解に陥り、その脱出が困難な状況においても、大域探索モードに切り替わり \mathbf{gbest} を参照することでその局所解から脱出することができる。

PSO-SC は動的な近傍構造を有するアルゴリズムであるがその構造は簡素である。したがって、PSO-SC を PC クラスタなどの並列計算環境への実装は容易に行うことができる。並列計算環境上に PSO-SC を実装する場合、1 つのサブ PSO に対して 1 つの計算機を割り当て、 \mathbf{gbest} 情報メモリをネットワーク上で共有するだけで良い。 SR は \mathbf{gbest} 情報メモリへの通信確率を表すため、 SR を小さくするほど通信は発生せず、また、情報交換するデータも PSO と同様であり、通信にかかる

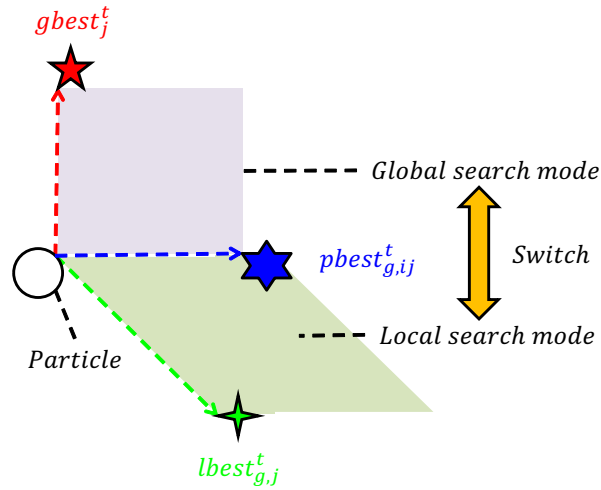


図 5.2: PSO-SC における粒子の探索領域の例

ボトルネックを小さくすることができる。

$f(\mathbf{x})$ を最小化する問題における PSO-SC の擬似コードを **Algorithm 5.1** に示す。ここで、 t_{max} は最大イタレーション数、 S は総サブ PSO 数、 N は各サブ PSO における粒子数を表す。

Algorithm 5.1 PSO-SC の擬似コード

```

1:  $t = 0$ ;
2: 全ての  $i$  に対して, 粒子  $i$  の  $\mathbf{v}_i^0, \mathbf{x}_i^0$  をランダムに初期化;
3: 全ての  $i$  に対して, 粒子  $i$  の評価値を  $f(\mathbf{x}_i^0)$  により計算;
4: 全ての  $i$  に対して, 粒子  $i$  の  $\mathbf{pbest}_i^0$  を自身の  $\mathbf{x}_i^0$  で初期化;
5: 全ての  $g$  に対して,  $\text{PSO}_g$  の  $\mathbf{lbest}_g^0$  を最良の  $\mathbf{pbest}_g^0$  で初期化;
6: gbest 情報メモリと通信を行い  $\mathbf{gbest}^0$  を最良の  $\mathbf{lbest}^0$  で初期化;
7: 全てのサブ PSO の探索モードを局所探索モードで初期化;
8: while  $t < t_{\max}$  do
9:   for  $g = 1$  to  $S$  do
10:    for  $i = 1$  to  $N$  do
11:      for  $j = 1$  to  $D$  do
12:         $v_{g,ij}^t, x_{g,ij}^t$  を式 (5.2), (5.3) により更新;
13:      end for
14:      if  $f(\mathbf{x}_{g,i}^{t+1}) < f(\mathbf{pbest}_{g,i}^t)$  then
15:         $\mathbf{pbest}_{g,i}^{t+1} = \mathbf{x}_{g,i}^{t+1}$ ;
16:      end if
17:    end for
18:     $k = \arg \min_i f(\mathbf{pbest}_{g,i}^{t+1})$ ;
19:    if  $f(\mathbf{pbest}_{g,k}^{t+1}) < f(\mathbf{lbest}_g^t)$  then
20:       $\mathbf{lbest}_g^{t+1} = \mathbf{pbest}_{g,k}^{t+1}$ ;
21:    end if
22:  end for
23:  全ての  $g$  に対して,  $\text{PSO}_g$  の探索モードを式 (5.1) により決定;
24:  for  $g = 1$  to  $S$  do
25:    if  $\text{PSO}_g$  が大域探索モード then
26:      if  $f(\mathbf{lbest}_g^{t+1}) < f(\mathbf{gbest}^t)$  then
27:         $\mathbf{gbest}^{t+1} = \mathbf{lbest}_g$ ;
28:      end if
29:    end if
30:  end for
31:   $t = t + 1$ 
32: end while

```

5.3 PSOn-SC と PSOn の多様性についての検討

表 5.1: 実験環境

	PSOn	PSOn-SC
総サブ PSO 数 (S)		6
粒子数/サブ PSO (N)		4
w		0.7
c_1		1.6
c_2		1.6
c_3	0.4	—
DBG	2, 4, 5	—
SR	—	0.05, 0.10, 0.50
次元数 (D)		10
イタレーション数 (t_{max})		5000
試行回数		100

表 5.2: ベンチマーク問題 I

関数	探索範囲	最適解の評価値
Rastrigin's $F_1(\mathbf{x}) = 10D + \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i))$	[-5.12, 5.12]	0
Schwefel's $F_2(\mathbf{x}) = 418.9829D + \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$	[-512, 512]	0

本節では、PSOn-SC と PSOn の多様性の比較を行う。PSOn では、多様性は DBG により制御され、 DBG を小さく設定すれば多様性が向上する。ただし、PSOn はサブ PSO 間の近傍構造が固定的である。一方、PSOn-SC において、多様性は SR により制御され、近傍構造も動的に変化する。 SR に小さな値を設定することで多様性が向上し、イタレーションごとにサブ PSO 間の近傍構造が変化する。したがって、PSOn-SC は PSOn よりも多様性が高いと考えられる。

表 5.1, 表 5.2 に実験環境とベンチマーク問題をそれぞれ示す。PSOn-SC, PSOn のパラメータは経験的に選択し、ベンチマーク問題はいずれも多峰性問題である。PSOn-SC, PSOn の多様性を調査するために、 PSO_g の多様性を次式で定義される σ_g を用いて定量化する [5.1].

$$\sigma_g = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{k=1}^D (x_{g,ik} - \bar{x}_{g,k})^2} \quad (5.4)$$

$$\bar{x}_{g,k} = \frac{1}{N} \sum_{i=1}^N x_{g,ik} \quad (5.5)$$

ここで、 N は各サブ PSO における粒子数、 D は設計変数の総数、 $\bar{x}_{g,k}$ は PSO_g における k 次元要素

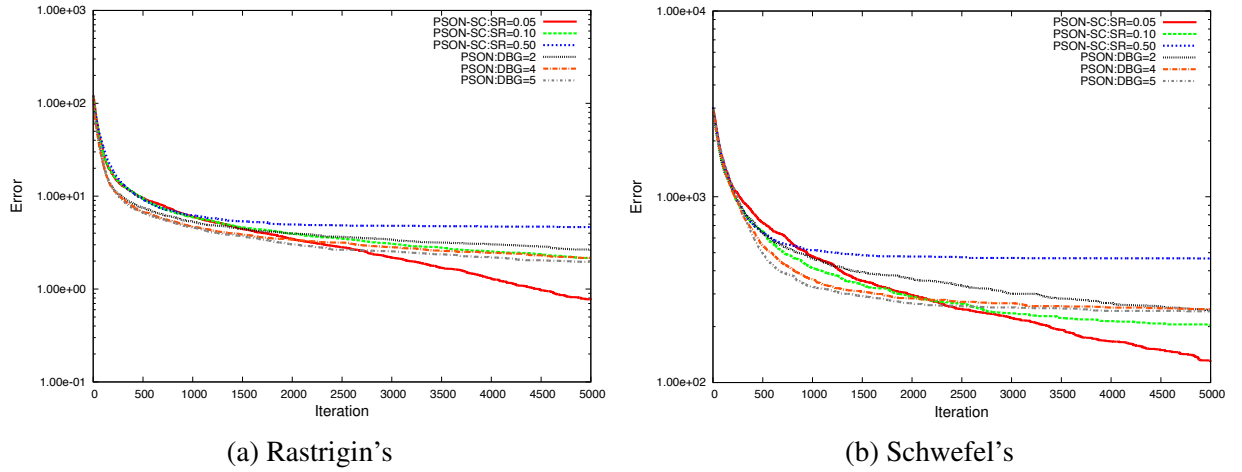


図 5.3: 解探索性能の比較

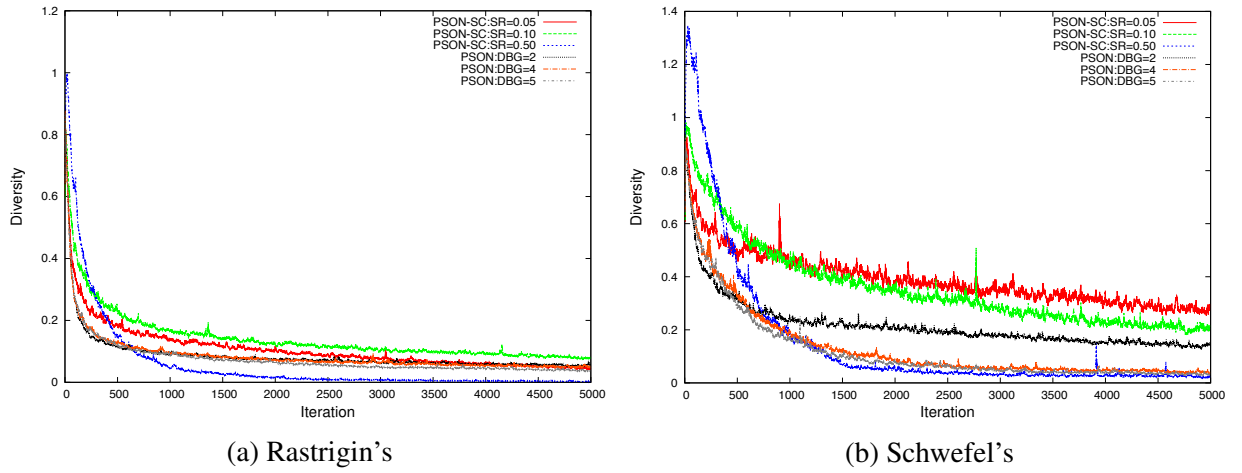


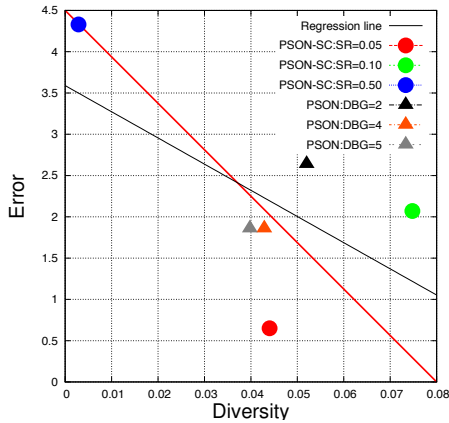
図 5.4: PSON-SC, PSON におけるサブ PSO の平均多様性

の平均点を表す。式 (5.4) において、 $\sigma_g = 0$ となれば PSO_g の多様性が失われたこと (解収束したこと) を意味する。また、多くのサブ PSO が独立的に探索が行われている場合は σ_g は 0 とならず、多様性を失うことなく解探索が継続的に行われていることを表す。PSON-SC, PSON において、各サブ PSO の多様性を式 (5.4) より評価し、実験結果を全サブ PSO の多様性の平均値 (平均多様性) で示す。平均多様性は次の式で計算される。

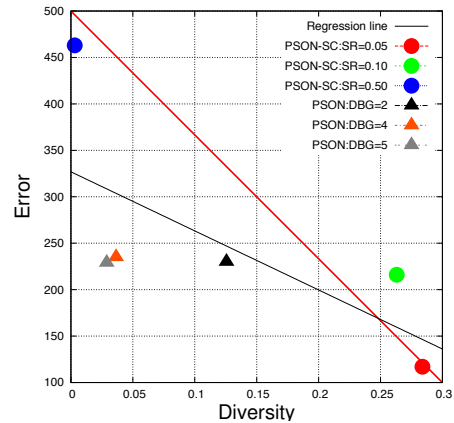
$$Diversity = \frac{\sum_{g=1}^S \sigma_g}{S} \quad (5.6)$$

ここで、 S はサブ PSO の総数を表す。各実験結果はランダムに初期値を変更した 100 回試行の平均値で示す。

図 5.3 に各ベンチマーク問題に対する PSON-SC, PSON における各 SR , DBG の評価値の誤差値



(a) Rastrigin's (相関係数:-0.60832)



(b) Schwefel's (相関係数: -0.68723)

図 5.5: PSOn-SC, PSOn におけるサブ PSO の多様性と解探索性能の相関性

の推移を示す。図 5.3 において、横軸はイタレーション数、縦軸は最良値と最適解の評価値の誤差値の平均値を表す。図 5.3 より、PSOn-SC における $SR = 0.05$ のときの解探索性能が各ベンチマーク問題に対して最良であることが確認できる。また、PSOn-SC ($SR = 0.05$) では、5000 イタレーション経過後も解探索が継続しており、小さな SR に設定することが多峰性問題に対して有効であると考えられる。

次に、PSOn-SC, PSOn における全サブ PSO の式 (5.6) で定義される平均多様性 “Diversity” を調査する。図 5.4 より、PSOn-SC において $SR = 0.05$, $SR = 0.10$ と SR を低確率に設定するとき、PSOn において $DBG = 2$ のように DBG を小さな値に設定するとき、多様性が高く維持されていることが確認できる (図 5.4 (b))。一方、PSOn-SC および PSOn を他のパラメータ (*i.e.*, $SR = 0.50$, $DBG = 4, 5$) に設定したときは、多様性がすぐに失われている。PSOn では、 DBG を小さな値に設定したとしても、サブ PSO 間の近傍構造が静的であるから、その多様性にも限界がある。一方、PSOn-SC では、 SR を小さく設定することで各サブ PSO は独立して解空間を探索するようになり、さらに、サブ PSO 間の近傍構造がイタレーションごとに変化する動的な構造である。したがって、PSOn-SC は PSOn よりも多様性を高く維持することができ、多峰性問題に対する解探索性能も高い。

これまで、本論文では多様性を増すことで解探索性能が向上すると述べてきた。ここで、多様性を定量化したので、解探索性能を示す収束誤差との相関性をチェックしてみる。図 5.5 に 5000 イタレーション時における、平均多様性 “Diversity” を横軸に、最良値と最適解の評価値の誤差値の平均値 “Error” を縦軸に示す。なお、図中の “Regression line” は最小二乗法により求めた。図 5.5 より、明らかに平均多様性 “Diversity” が大きい場合、最良値と最適解の評価値の誤差値の平均値 “Error” は小さくなっていて、多様性と解探索性能の相関が確認できた。

5.4 数値実験

表 5.3: 比較対象

Method	Outline
PSO[5.2]	全結合型の近傍構造
NOPSO[5.3]	近傍粒子数が徐々に増加する動的な近傍構造
RPSO[5.4]	粒子間に円環状の近傍構造
VPSO[5.5]	von Neumann 型の近傍構造
IPSO[5.6]	各粒子は確率的に <i>gbest</i> を参照する動的な近傍構造
PSON	サブ PSO 間に円環状の近傍構造

表 5.4: ベンチマーク問題 II

Function	Range
Ackley's $F_3(\mathbf{x}) = 20 - 20 \exp \left(-0.2 \sqrt{1/D \sum_{i=1}^D x_i^2} \right) + e - \exp \left(1 / \sum_{i=1}^D \cos(2\pi x_i) \right)$	[-32,32]
Griewank's $F_4(\mathbf{x}) = 1 + \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \frac{\cos(x_i)}{\sqrt{i}}$	[-600,600]
Schaffer's $F_5(\mathbf{x}) = \sum_{i=1}^{D-1} (x_i^2 + x_{i+1}^2)^{0.25} \times \left(1 + \sin^2 \left(50(x_i^2 + x_{i+1}^2)^{0.1} \right) \right)$	[-100,100]

本節では、PSON-SC の有効性を示すため、PSON を含めた様々な近傍構造を有する PSO 手法との解探索性能の比較を行う。PSON-SC の比較対象の手法とその概要 (Outline) を表 5.3 に示す。PSO, RPSO, VPSO は静的な近傍構造を粒子間に持つ PSO 手法であり、NOPSO, IPSO は動的な近傍構造を粒子間に持つ PSO 手法である。また、PSON はサブ PSO 間に静的な近傍構造を持つマルチスウォーム PSO 手法である。

表 5.5 に本実験で用いるベンチマーク問題の一覧を示す。ベンチマーク問題は CEC'13 ベンチマーク問題 [5.7] における 6 個のベンチマーク問題 ($F_6 \sim F_{11}$) を含む 11 個を用いる。その他、表 5.4 に示す 3 つのベンチマーク問題 ($F_3 \sim F_5$) と表 5.2 に示す 2 つのベンチマーク問題 (F_1, F_2) を用いる。表 5.5 に示すように、 $F_1 \sim F_5$ は多峰性問題、 F_6, F_7 は単峰性問題、 F_8, F_9 は解空間を回転させた単峰性問題、 F_{10}, F_{11} は解空間を回転させた多峰性問題を表す。また、 $F_6 \sim F_{11}$ は CEC'13 ベンチマーク問題における解空間をシフトさせた問題である [5.7]。

表 5.6 に各手法のパラメータ候補を示し、表 5.7 に実験環境を示す。各手法のパラメータは各ベンチマーク問題に対して良好な結果を示したパラメータを選択した。表 5.8 に各手法において良好な結果を示した適切なパラメータ設定を示す。

実験結果は探索終了時における最良の評価値と最適解の評価値との誤差値で示し、ランダムに初

表 5.5: ベンチマーク問題一覧

F	関数	最適解の評価値
F_1	Rastrigin's function	0
F_2	Schwefel's function	0
F_3	Ackley's function	0
F_4	Griewank's function	0
F_5	Schaffer's function	0
F_6	Sphere function (f_1)	-1400
F_7	Different powers function (f_5)	-1000
F_8	Rotated bent cigar function (f_3)	-1200
F_9	Rotated discus function (f_4)	-1100
F_{10}	Rotated rosenbrock's function (f_6)	-900
F_{11}	Rotated griewank's function (f_{10})	-500

f_* denotes the CEC'13 benchmark function

表 5.6: 各手法のパラメータ候補

Method	Parameters	Set
All methods	w	{0.1,0.3,0.5,0.7,0.9}
	$c_1 = c_2$	{0.2,0.4,0.6,...,1.6}
PSO-SC	SR	{0.005,0.01,0.02,0.05,0.1,0.2,0.4,0.6,0.8}
NOPSO	$Wini$	0.95
	$Wfin$	0.20
	c_1	{0.2,0.4,0.6,...,1.6}
	c_2	{0.2,0.4,0.6,...,1.6}
IPSO	C	{0.005,0.01,0.02,0.05,0.1,0.2,0.4,0.6,0.8}
PSO-N	DBG	{2,8}
	c_3	{0.2,0.4,0.6,...,1.6}

表 5.7: 実験環境

	PSO-SC & PSO-N	the others
The No. of sub-PSOs (S)	9	1
The No. of particles/sub-PSO (N)	4	36
Dimension (D)	30	
Iteration (t_{max})	5000	
Trial	100	

期値を変更させた 100 回試行の平均値で示す。表 5.9 と図 5.6, 図 5.7 に実験結果を示す。図 5.6, 図 5.7 では、実験結果を箱ひげ図 (Boxplot graph) により示している。表 5.9 において、“Mean” は誤差値の平均値, “SD” は標準偏差, “Median” は中央値を表す。中央値は信号の母集団が非ガウス分布の場合、平均値よりも良好に母集団の中心を推定できる。図 5.6, 図 5.7 において、縦軸は 5000 イタレーション終了時における最良解と最適解の評価値の誤差値を表す。

表 5.9 より、全ての手法が単峰性問題 F_6 , F_7 に対して最適解を発見できており、単峰性問題を解く解探索能力を有することが示されている。したがって、 F_6 および F_7 の結果は図 5.6, 図 5.7 から除外して示している。図 5.6, 図 5.7 より、5000 イタレーション終了時において、PSO-N は 7 個のベンチマーク問題において最良の解探索性能を示しているが、RPSO, VPSO, IPSO, そして、PSO-SC と大きな差はない。

表 5.9 と図 5.6, 図 5.7 より、ほとんど全ての手法において比較的簡単な多峰性問題も含めた F_3 , F_4 , F_6 , F_7 に対して 5000 イタレーション終了時点で最適解を発見することができている。そこで、これらのベンチマーク問題以外に対して 30000 イタレーションまで探索を行う。表 5.10, 図 5.8, 図 5.9, 図 5.10 に 30000 イタレーション終了時点における解探索性能の比較結果を示す。また、表 5.10 における平均値および標準偏差を用いて t 検定 [5.8] を行った。有意水準は 5% として t 検定を行った

結果を表 5.11 に示す。表 5.11 には PSON-SC と比較手法における“ p 値” (p -value; 有意確率) も示した。表 5.11 において, “+” は PSON-SC よりも優れた結果を示していると判定されたベンチマーク問題の個数を, “=” は PSON-SC と同等の結果であると判定されたベンチマーク問題の個数を, “-” は PSON-SC が有意水準 5% で優れた結果であると判定されたベンチマーク問題の個数を表す。

図 5.10 において, 横軸はイタレーション数, 縦軸は最良解と最適解の評価値の誤差値の中央値を表す。図 5.8, 図 5.9, 図 5.10 より, PSON-SC 以外の手法は多峰性のベンチマーク問題 F_1 , F_2 , F_5 に対して最適解を求めることができていない。一方, PSON-SC は F_1 および F_5 に対して最適解を発見することができている。また, 表 5.11 より, PSON-SC は多峰性のベンチマーク問題に対して他手法よりも良好な解探索性能を示している。

回転されたベンチマーク問題は非回転のベンチマーク問題よりも解空間が複雑であり, 難易度の高い最適化問題である。 F_8 においては, 全ての手法が最適解を発見することができていない。図 5.10 より, PSON は F_9 , F_{10} に対して最良の解探索性能を示している。PSON-SC や IPSO においても 30000 イタレーション終了時点でも最良解への更新は続いており, PSON が PSON-SC よりも探索性能に優れているのではなく, PSON-SC の収束速度が PSON よりも遅いことを意味している。このことは表 5.11 より, PSON-SC はこれらのベンチマーク問題に対して PSON と同等の解探索性能を有していることから確認できる。また, IPSO も動的な近傍構造により解空間を効率的に探索できる手法である。したがって, PSON-SC, PSON, IPSO は最適解を発見できることが期待できる。

以上の結果から, 実験的な観点から PSON-SC は多峰性問題だけでなく, 広いクラスの回転問題に対しても従来の種々の PSO と比較して最適解を発見することができる手法であることが確認できる。

表 5.8: 各手法のパラメータ設定

F	PSO	NOPSO	RPSO	VPSO	IPSO	PSO-N	PSO-N-SC
F_1	$w = 0.9$	$c_1 = 1.6$	$w = 0.9$	$w = 0.7$	$w = 0.9$	$w = 0.9$	$w = 0.7$
	$c_1 = c_2 = 1.0$	$c_2 = 0.4$	$c_1 = c_2 = 0.2$	$c_1 = c_2 = 0.4$	$c_1 = c_2 = 0.8$	$c_1 = c_2 = 0.2$	$c_1 = c_2 = 1.6$
	–	–	–	–	$C = 0.8$	$c_3 = 1.4$	$SR = 0.02$
	–	–	–	–	–	$DBG = 8$	–
F_2	$w = 0.7$	$c_1 = 1.6$	$w = 0.7$	$w = 0.7$	$w = 0.9$	$w = 0.9$	$w = 0.7$
	$c_1 = c_2 = 1.6$	$c_2 = 1.6$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 0.8$	$c_1 = c_2 = 0.2$	$c_1 = c_2 = 1.6$
	–	–	–	–	$C = 0.2$	$c_3 = 1.4$	$SR = 0.02$
	–	–	–	–	–	$DBG = 8$	–
F_3	$w = 0.9$	$c_1 = 1.6$	$w = 0.7$	$w = 0.7$	$w = 0.7$	$w = 0.7$	$w = 0.7$
	$c_1 = c_2 = 1.0$	$c_2 = 1.6$	$c_1 = c_2 = 1.4$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 1.6$
	–	–	–	–	$C = 0.6$	$c_3 = 0.8$	$SR = 0.02$
	–	–	–	–	–	$DBG = 2$	–
F_4	$w = 0.7$	$c_1 = 1.6$	$w = 0.7$	$w = 0.7$	$w = 0.7$	$w = 0.7$	$w = 0.7$
	$c_1 = c_2 = 1.6$	$c_2 = 1.6$	$c_1 = c_2 = 1.4$	$c_1 = c_2 = 1.4$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 1.6$
	–	–	–	–	$C = 0.4$	$c_3 = 0.6$	$SR = 0.02$
	–	–	–	–	–	$DBG = 2$	–
F_5	$w = 0.7$	$c_1 = 1.6$	$w = 0.9$	$w = 0.9$	$w = 0.9$	$w = 0.7$	$w = 0.9$
	$c_1 = c_2 = 1.6$	$c_2 = 0.4$	$c_1 = c_2 = 0.6$	$c_1 = c_2 = 0.8$	$c_1 = c_2 = 0.8$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 1.0$
	–	–	–	–	$C = 0.2$	$c_3 = 0.6$	$SR = 0.01$
	–	–	–	–	–	$DBG = 2$	–
F_6	$w = 0.7$	$c_1 = 1.2$	$w = 0.3$	$w = 0.3$	$w = 0.7$	$w = 0.9$	$w = 0.9$
	$c_1 = c_2 = 1.6$	$c_2 = 1.6$	$c_1 = c_2 = 1.4$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 0.2$	$c_1 = c_2 = 0.6$
	–	–	–	–	$C = 0.1$	$c_3 = 1.2$	$SR = 0.8$
	–	–	–	–	–	$DBG = 8$	–
F_7	$w = 0.7$	$c_1 = 1.6$	$w = 0.3$	$w = 0.3$	$w = 0.7$	$w = 0.9$	$w = 0.9$
	$c_1 = c_2 = 1.6$	$c_2 = 1.6$	$c_1 = c_2 = 1.4$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 0.2$	$c_1 = c_2 = 0.6$
	–	–	–	–	$C = 0.1$	$c_3 = 1.2$	$SR = 0.8$
	–	–	–	–	–	$DBG = 8$	–
F_8	$w = 0.7$	$c_1 = 1.6$	$w = 0.7$	$w = 0.9$	$w = 0.7$	$w = 0.9$	$w = 0.7$
	$c_1 = c_2 = 1.6$	$c_2 = 1.6$	$c_1 = c_2 = 1.0$	$c_1 = c_2 = 0.6$	$c_1 = c_2 = 1.4$	$c_1 = c_2 = 0.2$	$c_1 = c_2 = 1.2$
	–	–	–	–	$C = 0.8$	$c_3 = 0.8$	$SR = 0.05$
	–	–	–	–	–	$DBG = 2$	–
F_9	$w = 0.7$	$c_1 = 1.6$	$w = 0.7$	$w = 0.9$	$w = 0.7$	$w = 0.9$	$w = 0.9$
	$c_1 = c_2 = 1.4$	$c_2 = 1.6$	$c_1 = c_2 = 1.0$	$c_1 = c_2 = 0.6$	$c_1 = c_2 = 1.4$	$c_1 = c_2 = 0.2$	$c_1 = c_2 = 0.6$
	–	–	–	–	$C = 0.8$	$c_3 = 1.2$	$SR = 0.8$
	–	–	–	–	–	$DBG = 8$	–
F_{10}	$w = 0.9$	$c_1 = 1.2$	$w = 0.7$	$w = 0.7$	$w = 0.7$	$w = 0.9$	$w = 0.9$
	$c_1 = c_2 = 0.8$	$c_2 = 1.6$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 1.0$	$c_1 = c_2 = 1.4$	$c_1 = c_2 = 0.6$	$c_1 = c_2 = 0.4$
	–	–	–	–	$C = 0.6$	$c_3 = 0.4$	$SR = 0.6$
	–	–	–	–	–	$DBG = 8$	–
F_{11}	$w = 0.9$	$c_1 = 1.6$	$w = 0.7$	$w = 0.9$	$w = 0.9$	$w = 0.9$	$w = 0.9$
	$c_1 = c_2 = 0.8$	$c_2 = 1.6$	$c_1 = c_2 = 1.6$	$c_1 = c_2 = 0.4$	$c_1 = c_2 = 0.6$	$c_1 = c_2 = 0.2$	$c_1 = c_2 = 0.4$
	–	–	–	–	$C = 0.6$	$c_3 = 0.6$	$SR = 0.6$
	–	–	–	–	–	$DBG = 8$	–

表 5.9: 解探索性能の比較結果 (5000 イタレーション終了時点)

F		PSO	NOPSO	RPSO	VPSO	IPSO	PSO-N	PSO-N-SC
F_1	Mean	5.18E+001	5.43E+001	5.57E+001	4.38E+001	4.09E+001	1.88E+001	3.70E+001
	SD	2.50E+001	1.30E+001	1.35E+001	1.48E+001	1.14E+001	6.13E+000	8.47E+000
	Median	4.68E+001	5.38E+001	5.67E+001	4.36E+001	4.08E+001	1.80E+001	3.70E+001
F_2	Mean	3.47E+003	3.76E+003	4.15E+003	3.97E+003	2.80E+003	1.64E+003	2.97E+003
	SD	6.42E+002	5.34E+002	4.50E+002	5.44E+002	5.24E+002	5.98E+002	2.91E+002
	Median	3.49E+003	3.79E+003	4.20E+003	4.03E+003	2.79E+003	1.52E+003	2.95E+003
F_3	Mean	5.18E-001	1.21E+000	0.00E+000	2.66E-002	5.30E-002	0.00E+000	0.00E+000
	SD	2.79E+000	9.16E-001	0.00E+000	1.88E-001	2.63E-001	0.00E+000	0.00E+000
	Median	2.10E-005	1.34E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F_4	Mean	1.99E-002	1.59E-002	1.26E-003	5.52E-003	1.56E-002	2.25E-003	1.54E-003
	SD	2.35E-002	2.01E-002	3.20E-003	7.63E-003	1.79E-002	4.86E-003	4.26E-003
	Median	9.87E-003	9.86E-003	0.00E+000	0.00E+000	1.23E-002	0.00E+000	0.00E+000
F_5	Mean	8.52E+001	8.19E+001	2.19E+001	1.17E+001	5.97E+001	6.84E-001	4.19E+001
	SD	3.01E+001	2.02E+001	1.31E+001	1.42E+001	2.98E+001	1.77E+000	2.02E+001
	Median	8.26E+001	7.97E+001	1.96E+001	7.14E+000	5.52E+001	1.95E-001	3.68E+001
F_6	Mean	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
	SD	0.00E+000	0.00E+000	0.00E+000	0.00E+000	1.00E-006	0.00E+000	0.00E+000
	Median	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F_7	Mean	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
	SD	0.00E+000	2.00E-006	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
	Median	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F_8	Mean	4.84E+008	1.41E+009	2.10E+008	5.27E+007	4.48E+008	8.23E+007	1.65E+008
	SD	6.51E+008	2.24E+009	2.69E+008	6.21E+007	5.77E+008	1.27E+008	2.28E+008
	Median	2.32E+008	6.74E+008	1.17E+008	2.94E+007	1.61E+008	2.92E+007	7.92E+007
F_9	Mean	1.03E+004	2.62E+004	2.63E+004	8.11E+003	9.63E+003	2.07E+003	5.18E+003
	SD	6.28E+003	1.02E+004	4.63E+003	3.42E+003	5.39E+003	1.58E+003	2.86E+003
	Median	9.11E+003	2.50E+004	2.67E+004	7.92E+003	8.56E+003	1.59E+003	4.79E+003
F_{10}	Mean	2.48E+001	5.19E+001	2.20E+001	6.16E+000	2.36E+001	2.73E+000	1.09E+001
	SD	2.31E+001	2.59E+001	1.79E+001	1.57E+001	2.17E+001	4.58E+000	1.13E+001
	Median	1.40E+001	4.79E+001	1.63E+001	4.53E-001	1.43E+001	1.80E+000	1.10E+001
F_{11}	Mean	1.19E-001	7.75E-001	1.42E-001	1.62E-001	1.08E-001	1.08E-001	8.43E-002
	SD	1.28E-001	7.18E-001	6.48E-002	8.32E-002	8.06E-002	5.24E-002	7.00E-002
	Median	8.13E-002	5.75E-001	1.28E-001	1.45E-001	9.35E-002	9.11E-002	6.65E-002

* 太字は最良の誤差値を示す

表 5.10: 解探索性能の比較結果 (30000 イタレーション終了時点)

F		PSO	NOPSO	RPSO	VPSO	IPSO	PSO-N	PSO-N-SC
F_1	Mean	2.11E+001	3.27E+001	5.17E+001	4.10E+001	3.94E+001	1.12E+001	2.50E+000
	SD	8.60E+000	8.80E+000	1.11E+001	1.20E+001	1.10E+001	5.88E+000	1.96E+000
	Median	1.99E+001	3.28E+001	5.17E+001	4.18E+001	3.88E+001	1.09E+001	1.99E+000
F_2	Mean	3.52E+003	2.83E+003	4.06E+003	3.77E+003	2.23E+003	1.26E+003	9.57E+002
	SD	6.00E+002	5.04E+002	4.65E+002	6.05E+002	5.45E+002	4.00E+002	3.08E+002
	Median	3.49E+003	2.86E+003	4.05E+003	3.83E+003	2.31E+003	1.16E+003	9.48E+002
F_5	Mean	8.82E+001	5.11E+001	1.56E+001	1.14E+001	4.42E+001	7.81E-001	0.00E+000
	SD	3.40E+001	1.61E+001	1.15E+001	1.16E+001	2.42E+001	1.81E+000	0.00E+000
	Median	9.21E+001	5.07E+001	1.23E+001	7.40E+000	4.28E+001	2.38E-001	0.00E+000
F_8	Mean	2.45E+008	6.09E+008	1.49E+008	3.32E+007	2.60E+008	5.61E+007	7.49E+007
	SD	3.17E+008	9.29E+008	2.27E+008	5.63E+007	4.08E+008	1.42E+008	1.35E+008
	Median	1.12E+008	2.71E+008	6.50E+007	1.65E+007	1.42E+008	1.69E+007	2.82E+007
F_9	Mean	1.04E+003	1.47E+003	6.42E+003	1.68E+002	4.81E+000	7.71E-006	3.50E-001
	SD	1.50E+003	1.27E+003	2.15E+003	2.17E+002	2.73E+001	3.09E-005	1.77E+000
	Median	4.09E+002	1.04E+003	6.19E+003	1.02E+002	4.33E-001	9.45E-007	2.00E-002
F_{10}	Mean	4.56E+000	2.91E+001	6.81E+000	6.71E-001	3.10E+000	2.03E+000	1.29E+000
	SD	1.10E+001	2.58E+001	5.30E+000	3.56E+000	4.99E+000	6.55E+000	4.96E+000
	Median	2.26E-001	1.46E+001	9.18E+000	1.92E-002	2.49E+000	2.51E-006	2.12E-002
F_{11}	Mean	1.37E-001	1.40E-001	1.17E-001	1.64E-001	1.39E-001	9.98E-002	7.82E-002
	SD	1.65E-001	8.39E-002	4.96E-002	8.74E-002	1.51E-001	1.12E-001	6.21E-002
	Median	9.85E-002	1.26E-001	1.16E-001	1.35E-001	9.35E-002	7.14E-002	6.65E-002

* 太字は最良の誤差値を示す

表 5.11: PSO-N-SC vs 比較手法における t 検定の結果 (30000 イタレーション終了時点)

F		PSO	NOPSO	RPSO	VPSO	IPSO	PSO-N	PSO-N-SC
F_1	Mean±SD	2.11E+01±8.60E+00	3.27E+01±8.80E+00	5.17E+01±1.11E+01	4.10E+01±1.20E+01	3.94E+01±1.10E+01	1.12E+01±5.88E+00	2.50E+00±1.96E+00
	p -value	2.20E-16*	2.20E-16*	2.20E-16*	2.20E-16*	2.20E-16*	2.20E-16*	—
F_2	Mean±SD	3.52E+03±6.00E+02	2.83E+03±5.04E+02	4.06E+03±4.65E+02	3.77E+03±6.05E+02	2.23E+03±5.45E+02	1.26E+03±4.00E+02	9.57E+02±3.08E+02
	p -value	2.20E-16*	2.20E-16*	2.20E-16*	2.20E-16*	2.20E-16*	1.90E-08*	—
F_5	Mean±SD	8.82E+01±3.40E+01	5.11E+01±1.61E+01	1.56E+01±1.15E+01	1.14E+01±1.16E+01	4.42E+01±2.42E+01	7.81E-01±1.81E+00	0.00E+00±0.00E+00
	p -value	2.20E-16*	2.20E-16*	2.20E-16*	4.40E-16*	2.20E-16*	4.24E-05*	—
F_8	Mean±SD	2.45E+08±3.17E+08	6.09E+08±9.29E+08	1.49E+08±2.27E+08	3.32E+07±5.63E+07	2.60E+08±4.08E+08	5.61E+07±1.42E+08	7.49E+07±1.35E+08
	p -value	2.44E-06*	1.41E-07*	6.11E-03*	5.41E-03*	3.64E-05*	3.43E-01	—
F_9	Mean±SD	1.04E+03±1.50E+03	1.47E+03±1.27E+03	6.42E+03±2.15E+03	1.68E+02±2.17E+02	4.81E+00±2.73E+01	7.71E-06±3.09E-05	3.50E-01±1.77E+00
	p -value	4.20E-10*	2.20E-16*	2.20E-16*	1.13E-11*	1.08E-01	5.19E-02	—
F_{10}	Mean±SD	4.56E+00±1.10E+01	2.91E+01±2.58E+01	6.81E+00±5.30E+00	6.71E-01±3.56E+00	3.10E+00±4.99E+00	2.03E+00±6.55E+00	1.29E+00±4.96E+00
	p -value	7.95E-03*	2.20E-16*	1.53E-12*	3.12E-01	1.15E-02*	3.74E-01	—
F_{11}	Mean±SD	1.37E-01±1.65E-01	1.40E-01±8.39E-02	1.17E-01±4.96E-02	1.64E-01±8.74E-02	1.39E-01±1.51E-01	9.98E-02±1.12E-01	7.82E-02±6.21E-02
	p -value	1.13E-03*	1.67E-08*	3.11E-06*	1.60E-13*	3.23E-04*	9.51E-02	—
+		0	0	0	1	0	0	—
=		0	0	0	1	1	4	—
—		7	7	7	5	6	3	—

太字は最良の平均値を示し, * は PSO-N-SC に対して有意差があることを示す

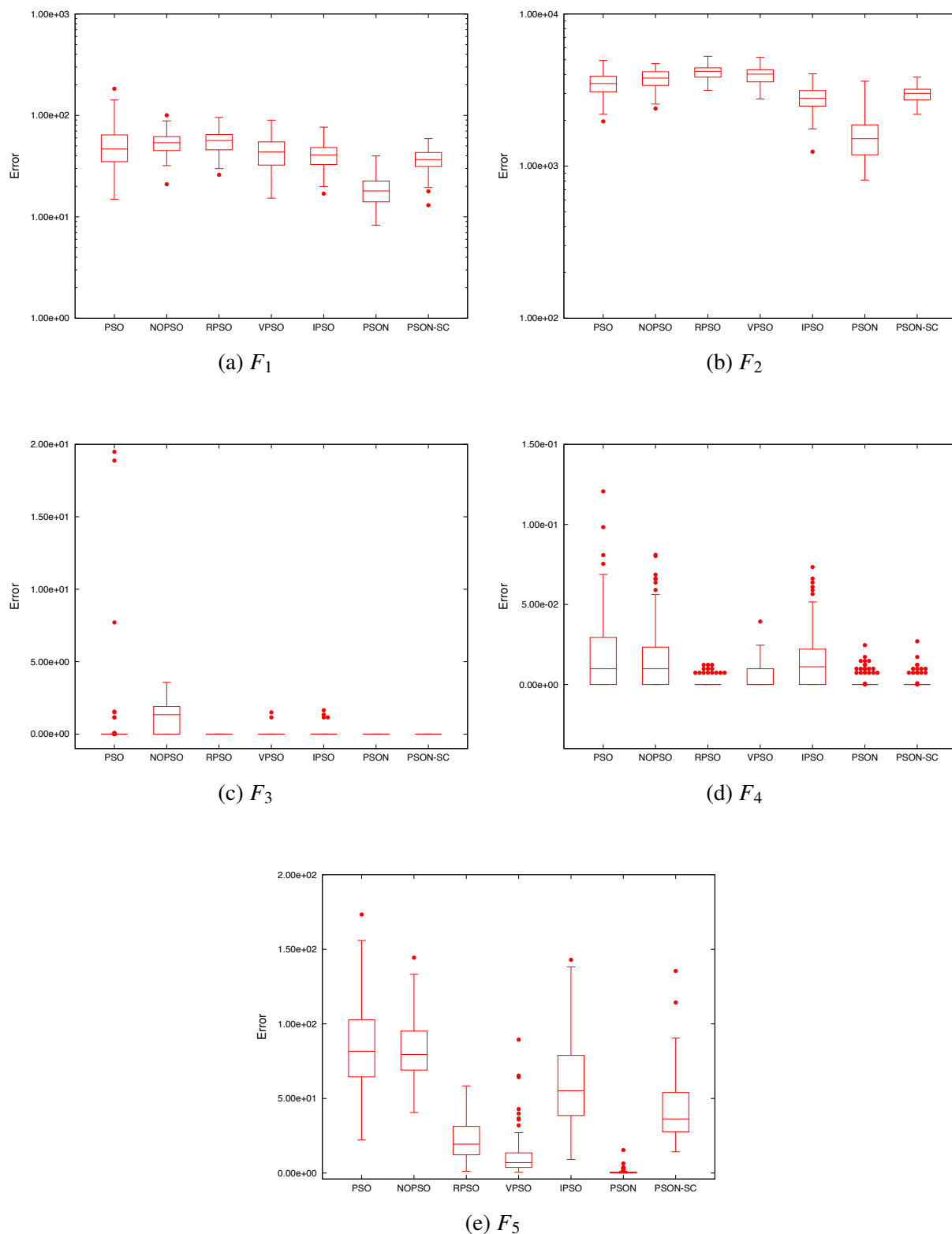
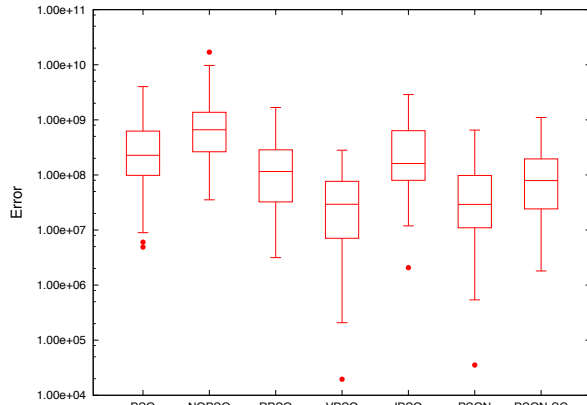
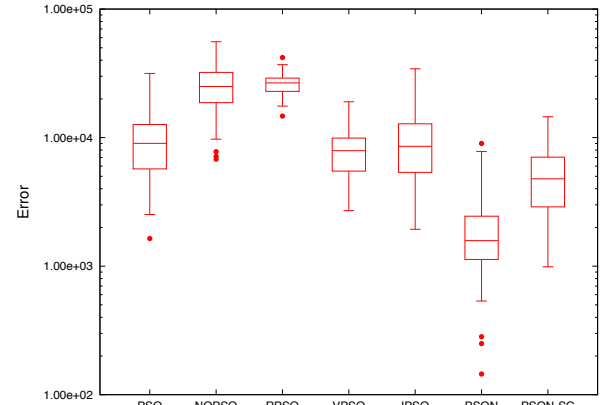


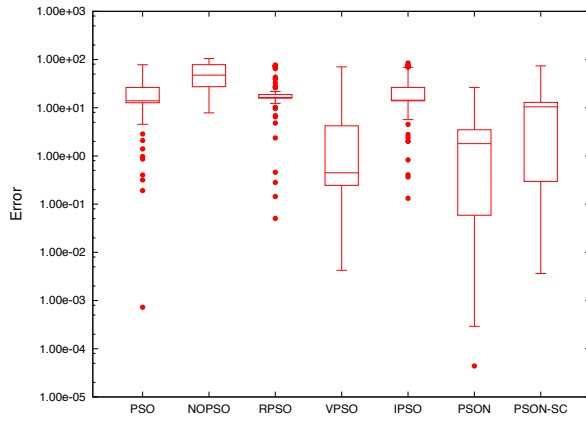
図 5.6: 箱ひげ図による解探索性能の比較結果 ($F_1 \sim F_5$: 5000 イタレーション終了時点)



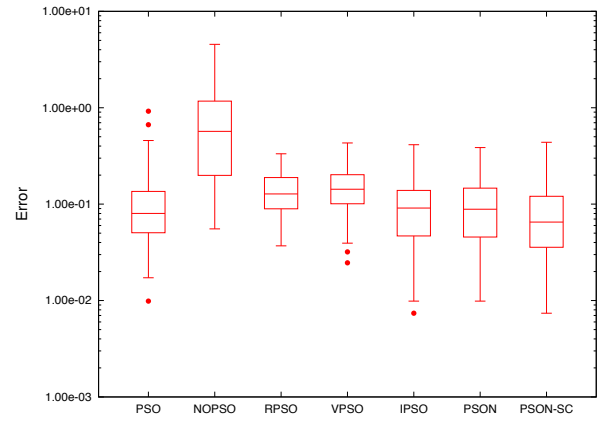
(f) F_8



(g) F_9



(h) F_{10}



(i) F_{11}

図 5.7: 箱ひげ図による解探索性能の比較結果 ($F_8 \sim F_{11}$: 5000 イタレーション終了時点)

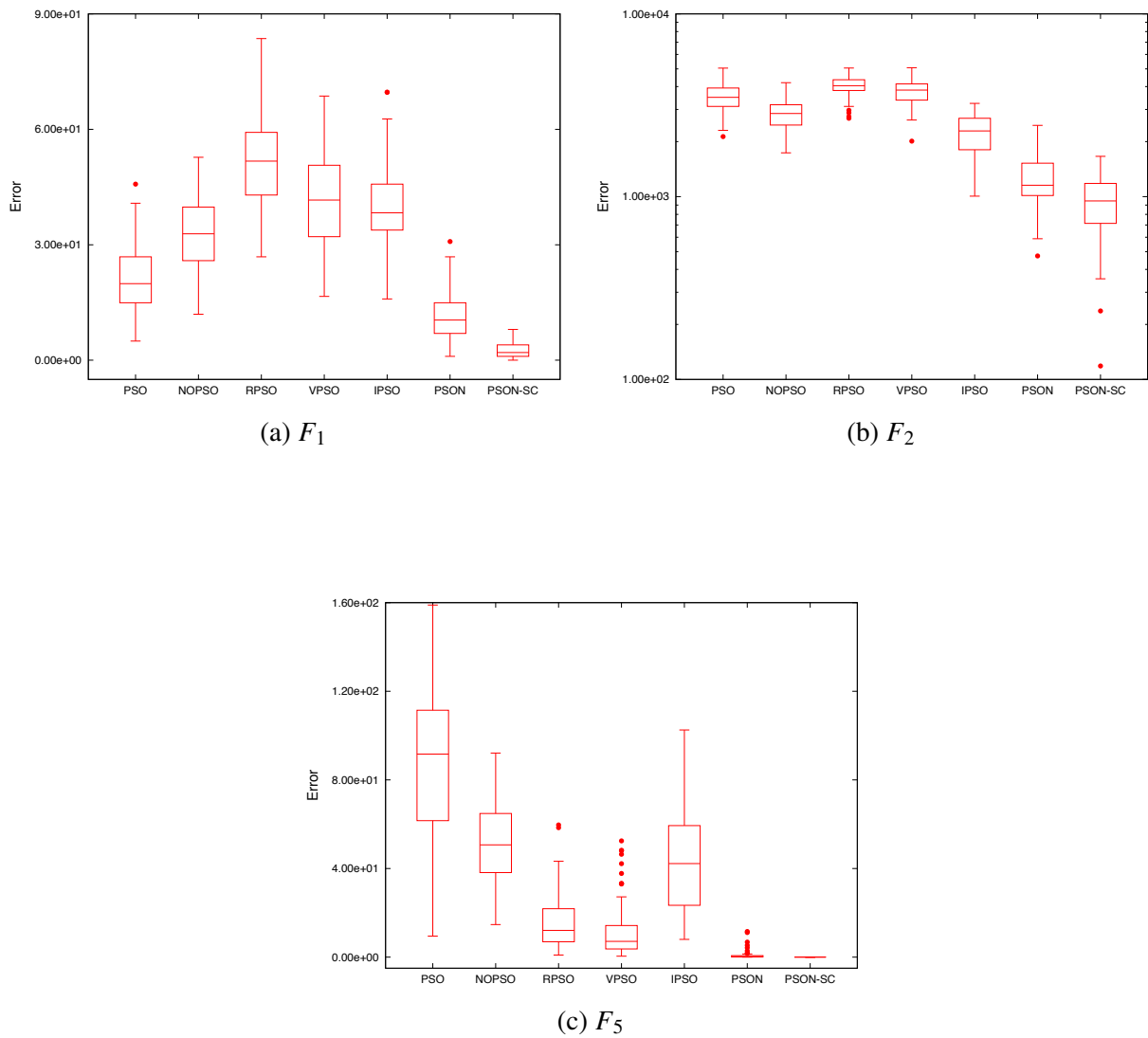


図 5.8: 箱ひげ図による解探索性能の比較結果 (F_1 , F_2 , F_5 : 30000 イタレーション終了時点)

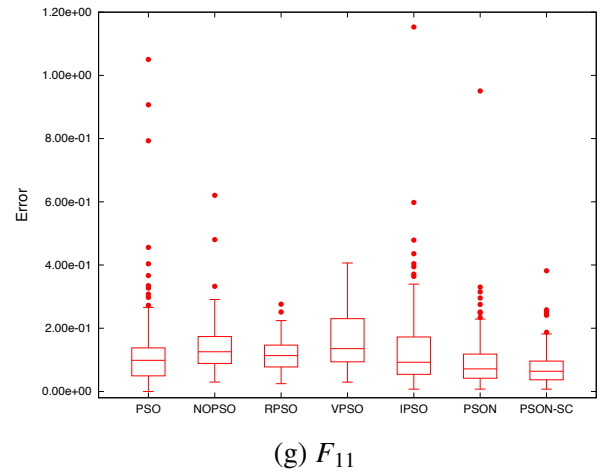
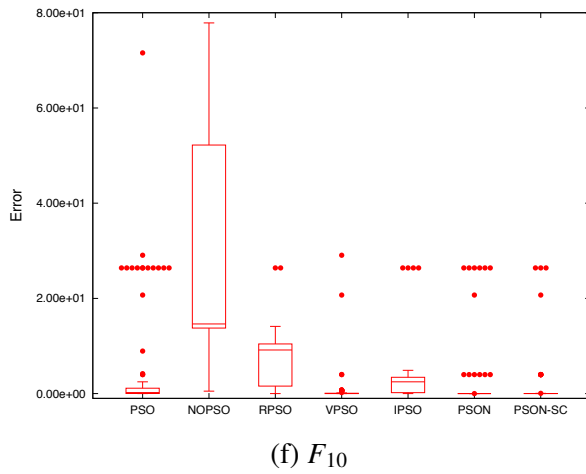
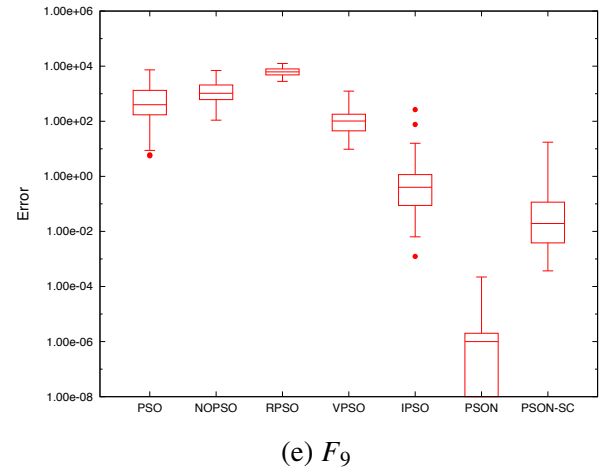
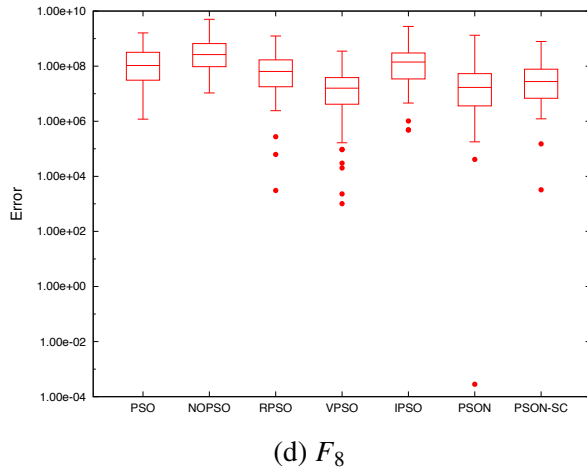


図 5.9: 箱ひげ図による解探索性能の比較結果 ($F_8 \sim F_{11}$: 30000 イタレーション終了時点)

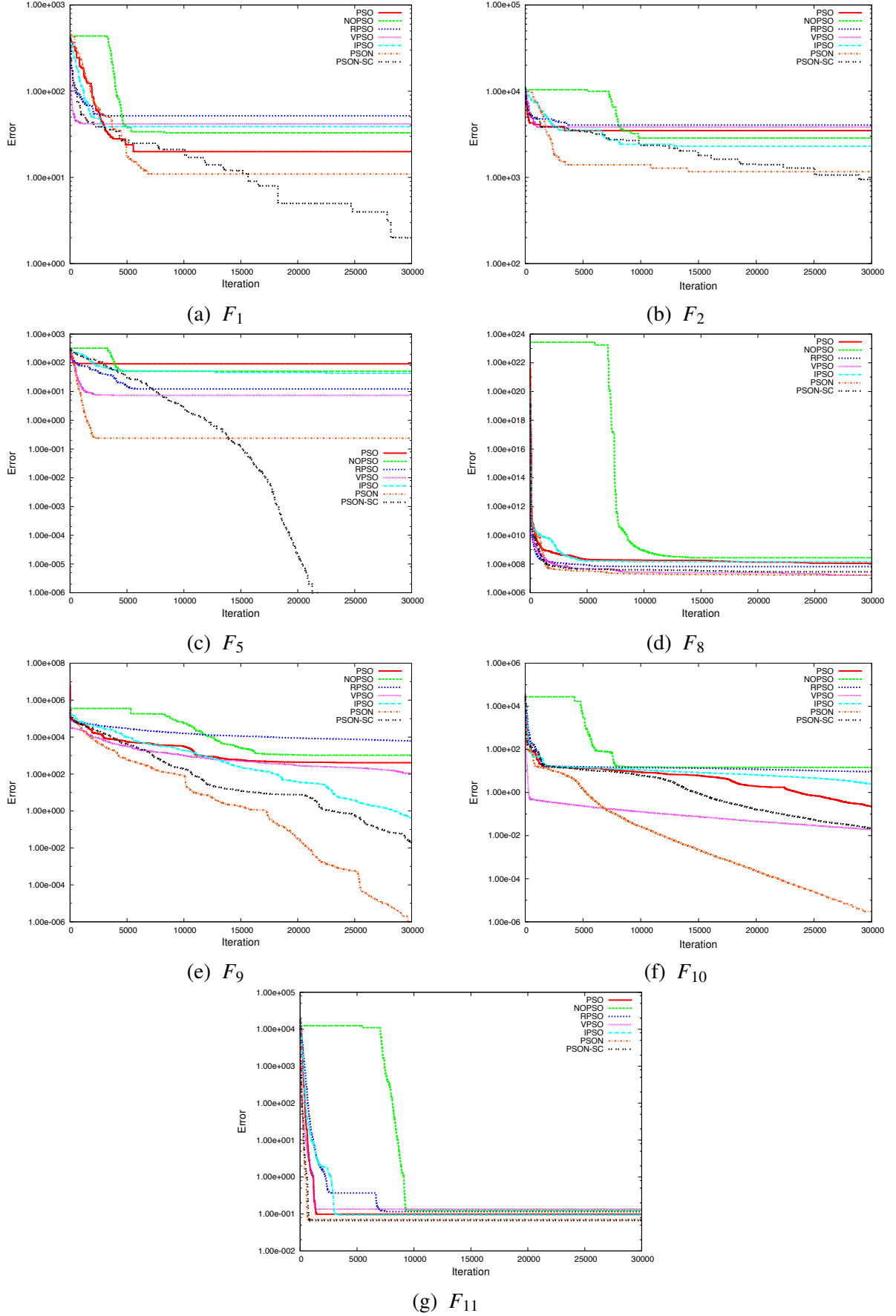


図 5.10: 解探索性能の比較結果 (30000 イタレーションまで)

5.5 計算時間における比較検討

表 5.12: 計測環境

CPU	2.4 GHz Intel Core i5
RAM	16GB 1600 MHz DDR3
OS	Mac OS X El Capitan
Compiler	GCC 4.2.1

表 5.13: PSO を基準とした相対時間

(5000 イタレーション終了時点)

F	NOPSO	RPSO	VPSO	IPSO	PSON	PSON-SC
F_1	1.900	0.982	1.005	0.918	1.111	0.920
F_2	2.065	1.230	1.179	0.998	1.313	1.060
F_3	1.996	0.789	0.815	0.783	1.011	0.793
F_4	2.257	1.061	1.105	0.970	1.273	1.088
F_5	1.447	1.059	1.075	0.991	1.070	1.064
F_6	2.427	0.954	0.978	0.640	1.289	0.917
F_7	1.724	1.008	0.997	0.839	1.561	0.953
F_8	1.411	0.950	0.959	0.942	1.022	0.900
F_9	1.705	0.986	1.001	1.006	1.116	0.962
F_{10}	1.674	1.008	0.990	0.952	1.119	0.939
F_{11}	1.478	1.095	1.021	0.975	1.039	0.965

* 太字は最良の相対時間を示す

表 5.14: PSO を基準とした相対時間

(30000 イタレーション終了時点)

F	NOPSO	RPSO	VPSO	IPSO	PSON	PSON-SC
F_1	2.413	1.203	1.227	0.964	1.167	0.984
F_2	2.002	1.410	1.231	0.851	1.233	1.018
F_3	2.558	1.064	1.087	0.916	1.343	0.985
F_4	2.203	1.113	1.151	0.923	1.294	1.065
F_5	1.559	1.025	1.013	0.923	1.040	0.982
F_8	1.369	0.950	0.923	0.956	1.038	0.914
F_9	1.694	0.967	0.981	1.002	1.147	0.965
F_{10}	1.644	1.015	0.944	0.960	1.101	0.926
F_{11}	1.437	1.049	0.986	0.978	1.084	0.993

* 太字は最良の相対時間を示す

次に、5.4 節における全ての手法に対する計算時間の比較を行う。ここでは、5000 イタレーションと 30000 イタレーションにおける計算時間を調査した。表 5.12 に手法を実装した計算機の仕様を示す。実験結果は PSO の計算時間を基準とした各手法における相対時間 (*i.e.*, [各手法の計算時間]/[PSO の計算時間]) で示し、100 回試行における相対時間の平均値で示す。表 5.13, 表 5.14 に実験結果を示す。

表 5.13, 表 5.14 より、NOPSO は最も計算時間を要することがわかる。NOPSO は動的なネットワークであり、各粒子に対しての近傍数を決定するために粒子間の距離計算を行っているためである。PSON-SC と IPSO の計算時間は短い。それに対して PSON は NOPSO を除く全ての手法よりも計算時間がわずかに長いことが確認できる。したがって、PSON-SC は PSON よりも解探索性能のみならず計算時間の観点からも優れた手法であることがわかる。

5.6 5章の結び

本章では、多峰性問題に対する PSON の解探索性能を向上させるために動的な近傍構造を PSON に導入した PSON-SC を提案し、その有効性を明らかにした。

PSON では、サブ PSO の *lbest* が更新された場合にそのサブ PSO は近傍サブ PSO の最良解 (*glbest*) を更新することができるため、それらの近傍サブ PSO は局所解からの脱出やより良好な解へ移動することができる。しかし、PSON は静的な構造であるため、探索範囲が限定され、多峰性問題においては局所解に陥る可能性がある。そこで、この問題を解決するために PSON に確率的な近傍構造を導入した PSON-SC を提案した。

PSON-SC では、全てのサブ PSO が *gbest* 情報メモリを確率的に参照するもので、その参照確率 SR を小さく設定することで各サブ PSO の多様性は PSON よりも高くできることを明らかにした。しかしながら、PSON-SC においてその多様性を高くすれば、解への収束速度が低下する。多様性と収束速度はトレードオフの関係となっている。

次に、PSON-SC の有効性を明らかにするために様々な近傍構造を有する PSO および PSON との解探索性能の比較を行い、PSON-SC はこれらの手法よりも多峰性問題において優れた解探索性能を示すことを明らかにした。さらに、単峰性問題や回転問題においても、他手法と同等の解探索性能を有することも明らかにし、その計算時間も PSON よりも短いことを明らかにした。

PSON-SC は動的に近傍構造を変化させるが、各サブ PSO は *gbest* 情報メモリとのみ通信するかどうかを確率的に決めるだけでよい。このため、PSON のように予め通信するサブ PSO を決定する必要はなく、確率的に通信するため SR を小さく設定することで、通信回数も PSON よりも小さくできることが期待できる。

今後は、PSON-SC の収束速度の改善、PSON-SC を並列計算環境へ実装し、実問題の最適化に適用することを考えている。

参考文献

- [5.1] O. Olorunda and A. P. Engelbrecht, “Measuring exploration/exploitation in particle swarms using swarm diversity,” in *Proc. of IEEE Congress on Evolutionary Computation*, pp. 1128-1134, 2008.
- [5.2] Y. Shi and R. C. Eberhart, “A modified particle swarm optimizer,” in *Proc. of IEEE Int. Conf. evolutionary computation*, pp. 69-73, 1998.
- [5.3] P. N. Suganthan, “Particle swarm optimiser with neighborhood operator,” in *Proc. of IEEE CEC*, pp. 1958-1963, 1999.
- [5.4] J. Kennedy, “Small worlds and mega-minds : effects of neighborhood topology on particle swarm performance,” in *Proc. of IEEE CEC*, pp. 1931-1938, 1999.
- [5.5] J. Kennedy and R. Mendes, “Population structure and particle swarm performance,” in *Proc. of IEEE CEC*, pp. 1671-1676, 2002.
- [5.6] H. Matsushita, Y. Nishio and T. Saito, “Particle swarm optimization with novel concept of complex network,” in *Proc. of NOLTA*, pp. 197-200, 2010.
- [5.7] J. J. Liang, B. Y. Qu, P. N. Suganthan and A. G. Hernández-Díaz, “Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization,” *Comput. Intell. Lab. Zhengzhou Univ. Zhengzhou, China Nanyang Technol. Univ. Singapore, Tech. Rep.*, vol. 201212, 2013.
- [5.8] B. Welch, “The generalization of ‘Student’s’ problem when several different population variances are involved,” *Biometrika*, Vol. 34, pp. 28-35, 1947.

第六章 結論

6 結論

工学的な分野における設計や計画などにおいて最適な状態を求めることは重要であり、その需要は高まってきている。このような状況に対してその選択を経験や勘に基づいて行うのではなく、科学的に行うことが望ましい。まず、解決すべき問題を観察し、その問題を数理計画することでモデルを作成する。そして、作成したモデルの最良解を導き出し、その解が解決案となる。

数理計画により定式化された最適化問題を解く手法は多数存在するが、数理計画が困難なブラックボックス最適化問題に対しては、勾配法などの種々の手法を用いることができない場合も多い。そのため、最適性の保証はなくても実行可能解の探索が可能なメタヒューリスティクスが利用されている。

近年、それらのブラックボックス最適化問題の解空間はますます複雑化・大規模化する傾向にある。このような大規模な最適化問題は高精度な近似解を実用的な計算時間で求めることができるメタヒューリスティクスを用いたとしても膨大な計算量を要することが多い。本論文では、大規模なブラックボックス最適化問題に対して短い計算時間で高精度な近似解を求めるため、メタヒューリスティクスの解探索性能を図ることを目的としている。さらに、計算時間の低減のために並列計算環境を考慮している点も本研究の特徴である。そこで、粒子の並列計算性が高く、解探索性能が良好なメタヒューリスティクスの一つである PSO に注目して研究に取り組んだ。PSO には初期収束の問題等があり、改良の余地がある。

PSO の解探索性能を向上させるために、これまで、粒子の近傍構造に関する研究やマルチスウォーム化の研究など様々に行われてきており、これらの研究は多峰性問題に対して初期収束の問題を改善し、解探索性能を向上している。特に、PSO の近傍構造については、探索の経過によらず固定的な近傍構造を持つ静的な手法よりも探索の経過につれて近傍構造が変化する動的な手法の方が解探索性能が向上することが示されている。また、PSO の並列計算環境への実装に関する研究も様々に行われてきた。しかしながら、解探索性能を向上させるための PSO の従来研究の多くは並列計算環境への実装を考慮した手法ではない。並列計算環境への実装手法を考慮して解探索性能を向上させる手法の開発が期待されている。

本論文では、マルチスウォーム PSO 手法に対してサブ PSO 間を近傍構造によりネットワーク化するモデルを 2 つ提案した。(1) 静的な近傍構造をサブ PSO 間に持つ PSO-N, (2) 動的な近傍構造をサブ PSO 間に持つ PSO-SC, である。この 2 つの手法では、分割した複数のサブ PSO を近傍構造により情報共有を行うことができ、各サブ PSO を並列計算環境における 1 つの計算機に実装することで解探索性能の向上および計算時間の低減を実現するものである。各サブ PSO は解空間を独立して探索し、近傍サブ PSO との情報共有により局所解からの脱出や良好な近似解の探索を行うこともできる。また、情報共有するデータは小さく、情報交換の仕組みも簡単に実現できるため、計算機間の通信により発生するボトルネックを小さくし、実装難易度も低い。したがって、大規模な

最適化問題を解く際に多数のサブ PSO を用意しても、並列計算により計算時間を低減することが可能となる。ただし、提案手法の並列計算環境への実装に関しては本論文の検討項目ではなく、本文中において言及されていない。

最後に、本論文における成果を総括する。2 章では、最適化問題およびメタヒューリスティクスの分類について説明し、本論文で取り扱う最適化問題および PSO の位置付けについて明らかにした。3 章では、標準的な PSO の説明および本論文に関する諸研究について説明し、提案手法がどのような研究の分類に属しているかについて明らかにした。4 章では、サブ PSO 間に静的な近傍構造を導入した PSON を提案した。慣性速度定数および加速度係数が PSON の粒子軌道に与える影響を解析し、粒子が安定して解に収束する条件を明らかにした。また、各サブ PSO の近傍サブ PSO 数と最適化問題の景観の関係性を明らかにし、様々な PSO 手法との解探索性能の比較を行い、PSON の有効性を明らかにした。5 章では、サブ PSO 間に動的な近傍構造を導入した PSON-SC を提案した。PSON では、その近傍構造が固定的であることから各サブ PSO の探索範囲が限定され、難易度の高い多峰性問題を解くために必要な多様性を持つことができない。PSON-SC はサブ PSO 間の近傍構造を動的にすることで PSON よりも多様性を向上させることができ、難易度の高い多峰性問題を解く場合において有効である。PSON-SC と PSON の多様性を比較し、PSON-SC は PSON よりも多様性が高いことを実験的に明らかにした。また、PSON を含む様々な近傍構造を持つ PSO 手法との解探索性能と計算時間の比較を行った。解探索性能が全比較対象の手法よりも優れていて計算時間においても優れていることを明らかにした。

本論文で与えた成果は、多数の粒子を必要とする大規模な最適化問題を高速に解くために有効な手法であり、今後、提案手法を並列計算環境へ実装することで、実世界の大規模なブラックボックス最適化問題を解くために貢献できるものと確信している。

謝辞

本研究を遂行し，論文を執筆するにあたり，終始，御指導して頂きました東京都市大学知識工学部情報科学科教授・田口 亮先生，宮内 新先生，同学科准教授・中野 秀洋先生に深謝の意を表する。

同学科教授・向井 信彦先生，大屋 英稔先生，日本工業大学工学部電気電子工学科教授・神野 健哉先生には審査委員としてご助言を頂くとともに本論文の細部にわたり御指導して頂きました。ここに深謝の意を表する。

本研究の一部は日本学術振興会科学研究費 (特別研究員 No. JP16J11745) によった。