

### 衝突粒子群最適化法とその複数解問題への応用

丸山, 一紀 / MARUYAMA, Kazuki

---

(発行年 / Year)

2014-03-24

(学位授与年月日 / Date of Granted)

2014-03-24

(学位名 / Degree Name)

修士(工学)

(学位授与機関 / Degree Grantor)

法政大学 (Hosei University)

# 2013年度 修士論文

論文題名 衝突粒子群最適化法と  
その複数解問題への応用

指導教授 斎藤 利通 教授

法政大学大学院工学研究科

電気工学専攻修士課程

学生証番号：12R3115

マルヤマ カズキ

氏名 丸山 一紀

# あらまし

粒子群最適化法 (Particle Swarm Optimizers, PSO) は鳥の群れや魚群のような生物群の振る舞いを基に考案された最適化手法の一種である。生物群は粒子群にモデル化され、粒子をエージェントとした粒子間の相互作用で探索を行う。その概念とアルゴリズムの簡潔さ、動作の柔軟さ、多様な改良の可能性、幅広い応用、などから注目を集めている。PSO には種々の応用が提案されているが、本論文では“離散力学系の複数周期点探索”に的を絞る。これを複数解問題 (複数の最適値を有する目的関数) と見なし、幾つかのアルゴリズムを提案する。

まず初めに PSO に“粒子の鈍感さ”と“粒子間の衝突”を持つアルゴリズム (鈍感衝突粒子群最適化法, ICPSO) について考察する。この手法は粒子群の収束を“鈍感”により遅延させ、収束後は“衝突”により粒子を拡散することにより、複数解問題での解探索を実現することを目的としている。適度な鈍感さと衝突を与えることは効果的な複数解探索を可能とした。

次に、衝突の近傍を考慮した PSO (衝突粒子群最適化法, CPSO) を考察する。粒子に“衝突のための近傍領域”を持たせることにより、粒子の動きに更なる多様性を付与することを可能とした。

最後に、多目的最適化問題と複数解問題の両者に対して有効な PSO (多目的衝突粒子群最適化法) を考察する。複数解探索を行うと同時に、複数の目的関数を同時に最適化する。これにより複数存在する周期点を求めると同時に、周期点の周期を分類することを可能とした。

いずれの PSO もデジタル PSO であり、離散空間での探索を行う点が従来の PSO とは異なる。加えて乱数パラメータ等の確率的な要素を含まない、決定論的な PSO である。このような特長は、計算コストの削減に寄与すると共にハードウェア化に貢献する可能性が示唆される。

# Collision Particle Swarm Optimizer and its Application to Multi-solution Problems

## Abstract

The particle swarm optimizers (PSO) is a population-based paradigm for solving optimization problems inspired by flocking behavior of living beings. The particles communicate each other in the swarm. The PSO is simple in concept, is easy to implement and has been applied to optimization problems in various systems. Various applications of the PSO have been proposed. However, we focus on the "search of multiple periodic points of discrete dynamical systems". We regard this problem as multi-solution problems (MSP), and propose several algorithms.

First, we consider including the "inter-particle collision" and "insensitivity". This method delays the convergence of the particles by the insensitivity, and spreads the particle swarm by "inter-particle collision". The collision and insensitivity can be effective to make a variety of swarm diversity for the MSP, if the control parameters are selected suitably.

Second, we consider the collision particle swarm optimizer with a concept of vicinity area of particle. Particles have "vicinity" area for inter-particle collision. It is effective to enlarge the particle diversity.

Finally, we consider the new CPSO that is effective both multi-objective problem (MOP) and MSP. The CPSO can solve MSP and can optimize MOP.

These PSOs are discrete and deterministic. They search solutions on a discrete search space and do not contain stochastic parameters. It is different from the conventional PSOs. Such deterministic systems are useful from viewpoints of reproducibility and simplicity.

# 目次

第1章	まえがき	7
1.1	研究背景	7
1.2	研究目的	8
1.3	本論の構成	9
第2章	鈍感さと衝突を伴う決定論的離散粒子群最適化法	15
2.1	導入	15
2.2	PSO アルゴリズム	16
2.3	数値実験	18
2.4	まとめ	19
第3章	決定論的衝突粒子群最適化法と離散複数解問題	27
3.1	導入	27
3.2	離散複数解問題	28
3.3	CPSO アルゴリズム	30
3.4	Hénon map への応用	32
3.5	まとめ	34
第4章	衝突粒子群最適化法と Hénon map の周期解探索	45
4.1	導入	45
4.2	離散的な複数解問題	46
4.2.1	離散時間力学系	46
4.2.2	Hénon map	48

4.3	CPSO アルゴリズム . . . . .	49
4.4	数値実験 . . . . .	51
4.5	まとめ . . . . .	53
<b>第 5 章</b>	<b>むすび</b>	<b>64</b>
<b>付 録 A</b>	<b>擬似コード</b>	<b>65</b>
A.1	DDPSO 擬似コード (第 2 章) . . . . .	65
A.2	CPSO 擬似コード (第 3 章) . . . . .	66
A.3	CPSO 擬似コード (第 4 章) . . . . .	67
<b>付 録 B</b>	<b>Ikeda map の概形と DDPSO 探索過程の要約</b>	<b>68</b>
	<b>研究業績</b>	<b>70</b>
	<b>謝辞</b>	<b>73</b>

# 目次

1.1	Conceptual diagram of the position determination of particle. . . . .	11
2.1	Social network structures. . . . .	20
2.2	Collision of particles $P_m$ and $P_n$ . . . . .	21
2.3	Search process for $C_1 = 0.25$ , $k = 16$ , $\epsilon = 0.2$ and $N = 50$ . . . . .	22
2.4	Search for $C_1 = 0.25$ , $k = 16$ , $\epsilon = 0.2$ , $t_{max} = 50$ . . . . .	23
3.1	Discrete solution regions (DSRs). . . . .	35
3.2	Definition of $l$ -neighbors. . . . .	36
3.3	Discrete multi-solution problem based on the Hénon map. . . . .	37
3.4	An exploring process of LPSO for the DMSP. . . . .	38
3.5	An exploring process of CPSO for the DMSP. . . . .	39
3.6	Evolution of particles to DSRs of the DMSP. . . . .	40
4.1	Basic functions of the Hénon map for $a = 0.96875$ , $b = 0.3$ . . . . .	55
4.2	Basic concept in the CPSO algorithm. . . . .	56
4.3	Neighbor and vicinity. . . . .	57
4.4	Exploring process on the search space. . . . .	58
4.5	Personal best on the fitness functions plane in the exploring process. . . . .	59
4.6	Time evolution of the pbests of solution candidates $P_3, P_9, P_{15}$ and $P_{28}$ . . . . .	60
B.1	An objective function of Ikeda map for $u = 0.9$ . . . . .	68
B.2	Search for $C_1 = 0.25$ , $k = 0$ , $\epsilon = 0.2$ , $t_{max} = 50$ . . . . .	69
B.3	Search for $C_1 = 0.25$ , $k = 0$ , $\epsilon = 0$ , $t_{max} = 50$ . . . . .	69

# 表 目 次

2.1	Setting of parameters. . . . .	24
2.2	SR of DDPSO. . . . .	24
3.1	The average success rate (SR) of the DMSP based on the Hénon map. . . . .	41
3.2	The average number of iterations (#ITE) of the DMSP. . . . .	41
3.3	The average number of collisions (#COL) of the DMSP. . . . .	41
4.1	The average success rate (SR) in 100 trials. . . . .	60
4.2	The average number of iterations (#ITE) in successful runs. . . . .	60
4.3	The average number of collisions (#COL) in successful runs. . . . .	60



# 第1章 まえがき

## 1.1 研究背景

粒子群最適化法 (particle swarm optimizers, PSO) は最適化アルゴリズムの一種である [1][2]。鳥の群れや魚群ののような生物群の振る舞いを基に考案されたアルゴリズムであり，群知能 (swarm intelligence, SI) である。生物群は粒子群にモデル化される。個々の粒子は単純であるが，それらをエージェントとした粒子間の相互作用で粒子群は複雑な振る舞いをする。粒子群は探索空間内を移動し，最適値の発見に至る。その概念とアルゴリズムの簡潔さ，動作の柔軟さ，多様な改良の可能性，などの特徴を持つ。

- 微分可能性を要求しない：最急降下法・共役勾配法とは異なり目的関数に微分可能性を要求しない。また，連続性も必要としないため，不連続な目的関数に適用可能である。
- 柔軟な動作：PSO はアルゴリズムに乱数パラメータを含む，確率論的な探索手法である。他の集団的降下法に基づく乱数を用いたアルゴリズムとして，差分進化 (differential evolution, DE) がある。しかしながら，決定論的 PSO は乱数を排除しても，効果的な探索が可能である [17]。
- 非線形システムとの対応：決定論的 PSO は，大規模非線形力学系的一种とみなすことができる。これは非線形システムの動作解析法への応用が期待できる。

PSO はメタヒューリスティクスであり，広く応用されている：信号処理 [3]–[6]，ニューラルネットワーク [7]，電力システム [8]–[10]。

基本的な PSO [1][2] は乱数値をとるパラメータを含み，連続探索空間で定義される目的関数の探索を行う。これは単一解を持つ単峰性関数に対しては非常に強力な性能

を發揮する．しかし，必ずしも性能が發揮されない問題も存在する．

- 多峰性関数：複雑な形状をしている．大域的最適解の探索が目的だが，局所的最適解に陥り易い(トラップされる)ため，局所最適解からの脱出を必要とする．
- 複数解問題 (multi-solution problems , MSP)：最適解を複数持つ目的関数で，全ての最適解を発見することを目的とする．全ての最適解にトラップされることが必要とも言える．タブーサーチが代表的な手法である．
- 多目的最適化 (multi-objective problems , MOP)：複数の目的関数を同時に最適化する．しかしながら，完全最適解が存在しない場合も少なくなく，一方の目的関数の最適解であっても，他方の最適解であるとは限らない．このような問題に対処するために，パレート最適解の発見を目的とする．

## 1.2 研究目的

PSO には種々の応用が提案されているが，本論文では“離散力学系の周期点探索”に絞る [11]–[17]．力学系の周期点探索には種々な手法が存在する．しかしながら，その多く (例：Newton-Raphson 法) は局所的探索のために開発されている．これに対し，PSO は大域的探索と局所的探索の両方での活用が期待される．これは分岐現象解析への基礎となる．周期点は複数存在し，その全てを求めることが目的であるため，MSP である．目的関数には合成写像を用いるが，所望の周期点を探索することは複数の合成写像を目的関数とした解探索であり，MOP である．離散力学系の周期点探索問題は分岐現象解析，MSP，MOP の三つに関連している．

本論文では離散決定論的 PSO (discrete deterministic particle swarm optimizers , DDPSO) を用いる．以下に概要を述べる．離散時間を  $t$  とする．これは探索時間である．探索時間  $t$  における粒子群は  $P^t \equiv (P_1^t, \dots, P_N^t)$  で定義される．ただし， $N$ : 粒子数． $P_i^t \equiv (x_i^t, v_i^t)$  は  $i$  番目の粒子であり，位置と速度によって特徴付けられる． $x_i^t \equiv (x_{i1}^t, \dots, x_{iD}^t)$  は粒子の位置を示し， $v_i^t \equiv (v_{i1}^t, \dots, v_{iD}^t)$  は速度を示す．ただし， $D$ : 次元数， $i = 1, \dots, N$ ．粒子  $x_i^t$  は潜在解である．粒子の位置と速度の更新には，パーソナルベスト ( $pbest_i^t$ ) と

ローカルベスト ( $lbest_i^t$ ) を用いる。パーソナルベストは各々の粒子がこれまでに発見した“ベスト”な位置である。“ベスト”な位置は目的関数により与えられる。ローカルベストは  $P_i^t$  の“近傍の粒子”のパーソナルベストにより与えられる。基本的な PSO は連続探索空間を対象とするが、DDPSO は離散探索空間を対象とする。粒子は格子点から成る離散探索空間を探索し、その時間発展は乱数を含まない決定論的な差分方程式 (式 (1.1)) によって記述される [18]–[21]。

$$\begin{aligned} \mathbf{v}_i^{t+1} &\leftarrow w\mathbf{v}_i^t + c(lbest_i^t - \mathbf{x}_i^t) \\ \mathbf{x}_i^{t+1} &\leftarrow \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \end{aligned} \quad (1.1)$$

ただし、 $w$ ：慣性定数 (inertia weight)、 $c$ ：加速度係数 (acceleration coefficient)。

図 1.1 は PSO における粒子の探索位置決定の概念図である。粒子は慣性速度 (inertia velocity) とローカルベストを基とした速度 (social velocity) を基に、次の時間での自身の位置と速度を決定する (図 1.1(a))。ただし、 $lbest_i^t$  は近傍の粒子のパーソナルベストによって与えられる。なお、パーソナルベストを基とした速度 (cognitive velocity) を用いないことも本論文における DDPSO の特徴である。これらの特徴は非線形システムの動作解析法に有利であり、計算コストの削減と共にハードウェア化に貢献する可能性が示唆される。これを基本として周期点探索へと適用する。

### 1.3 本論の構成

本論文は 5 章から構成され、以下にその概要を述べる。

#### 第 2 章 鈍感さと衝突を伴う決定論的離散粒子群最適化法

複数解問題のための新しい DDPSO を考察する。システムは確率論的パラメータを含まない、決定論的な差分方程式に支配されている：これはダイナミクスの解析に適している。粒子は鈍感さと衝突を持ち、離散的な探索空間を移動する。鈍感さと衝突は群れの多様性に効果的であり、最適解の探索に有効である。基本的な数値実験を基にアルゴリズムの効率を調査する。

### 第3章 決定論的衝突粒子群最適化法と離散複数解問題

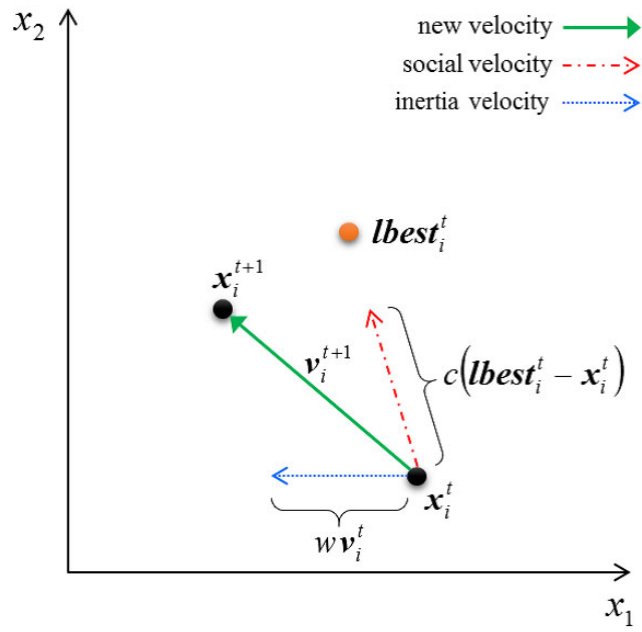
粒子間の衝突を伴う新しいDDPSOについて考察する。粒子の動きは離散探索空間において、決定論的な差分方程式により支配される。これは実験の再現性と回路実装の観点から有利である。粒子間の衝突は力学を参考にした、シンプルな法則で制御される。衝突は粒子群の多様性を拡大し、局所解、あるいは一部の最適解に粒子群が陥ることを避けるために効果的である。典型的な非線形力学系の周期点に対応した、離散複数解問題に本手法を適用する。この問題は非線形力学系解析に向けた、新たな応用に向けての第一歩に成り得る。アルゴリズムの性能は3つの特徴量にもとづいて検討される：成功率、平均探索時間、衝突回数。これらの結果は効率的なアルゴリズム開発のための基本的な情報となる。

### 第4章 衝突粒子群最適化法とHénon mapの周期解探索

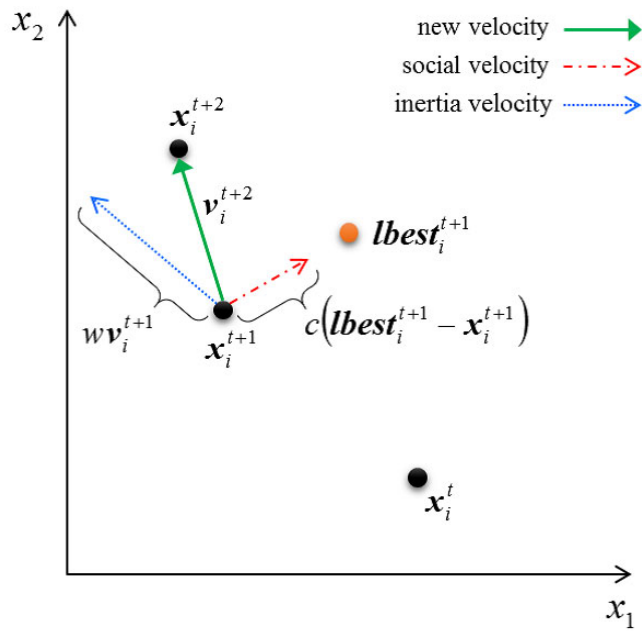
本章は本論文の中心である。離散的な複数解問題(discrete multi-solution problems, DMSP)のための衝突粒子群最適化法(collision particle swarm optimizer, CPSO)について考察する。CPSOは離散空間で定義され、決定論的な差分方程式により支配される。これは実験の再現性と回路実装の観点から有利である。CPSOは粒子間の衝突を有しており、これは局所解、あるいは一部の最適解に粒子群が陥ることを避けるために効果的である。エノン写像(Hénon map)には周期解が複数存在しており、これをDMSPと見なしCPSOを適用する。基本的な数値実験を行い、CPSOの基本的な機能を検討する。この応用はPSOに基づく非線形力学系の解析ツールを開発するための基礎となる。

### 第5章 むすび

本論文の総括をすると共に、今後の課題および展望について述べる。



(a) Time step  $t$ .



(b) Time step  $t + 1$ .

⊠ 1.1: Conceptual diagram of the position determination of particle.

## 参考文献

- [1] J. Kennedy and R. Eberhart, Particle Swarm Optimization, Proc of IEEE/ICNN, pp.1942–1948, 1995.
- [2] A. P. Engelbrecht, Fundamentals of computational swarm intelligence, Willey, 2005.
- [3] M. P. Wachowiak, R. Smolikova, Y. Zheng, J. M. Zurada and, A. S. Elmaghraby, An approach to multimodal biomedical image registration utilizing particle swarm optimization, IEEE Trans. Evol. Comput., 8, 3. pp. 289–301, 2004.
- [4] S.-T. Hsieh, T.-Y. Sun, C.-L. Lin, and C.-C. Liu, Effective learning rate adjustment of blind source separation based on an improved particle swarm optimizer, IEEE Trans. Evol. Comput., 12, 2. pp. 242–251, 2008.
- [5] A. Ghosh, A. Ghosh, A. Chowdhury, A. Konar, E. Kim and A. K. Nagar, Linear phase low pass FIR filter design using genetic particle swarm optimization with dynamically varying neighborhood technique, in Proc. IEEE Congress Evol. Comput., pp. 529–535, 2012.
- [6] R. A. Vural, T. Yildirim, T. Kadioglu and A. Basargan, Performance evaluation of evolutionary algorithms for optimal filter design, IEEE Trans. Evol. Comput., 16, 1. pp. 135–147, 2012.
- [7] A. B. van Wyk and A. P. Engelbrecht, Overfitting by PSO trained feedforward neural networks, in Proc. IEEE Congress Evol. Comput., pp. 2672–2679, 2010.

- [8] H. Qin, J. W. Kimball and G. K. Venayagamoorthy, Particle swarm optimization of high-frequency transformer, in Proc. Annual Conf. IEEE Ind. Electron. Soc., pp. 2908–2913, 2010.
- [9] K. Kawamura and T. Saito, Design of switching circuits based on particle swarm optimizer and Hybrid Fitness Function, in Proc. Annu. Conf. IEEE Ind. Electron. Soc., pp. 1099–1103, 2010.
- [10] H. Matsushita and T. Saito, Application of particle swarm optimization to parameter search in dynamical systems, NOLTA, IEICE, E94-N, 10, pp. 458–471, 2011.
- [11] K. E. Parsopoulos and M. N. Vrahatis, On the computation of all global minimizers through particle swarm optimization, IEEE Trans. Evol. Comput., 8, 3, pp. 211–224, 2004.
- [12] T. Hendtlass, WoSP: a multi-optima particle swarm algorithm, Proc. IEEE/CEC, pp. 727–734, 2005.
- [13] D. Parrott and X. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, IEEE Trans. Evol. Comput., 10, 4, pp. 440–458, 2006.
- [14] S. Yang and C. Li, A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, IEEE Trans. Evol. Comput., 14, 6, pp. 959–974, 2010.
- [15] X. Li, Niching without niching parameters: particle swarm optimization using a ring topology, IEEE Trans. Evol. Comput., 14, 1, pp. 150–169, 2010.
- [16] T. Saito and E. Miyagawa, Growing-tree particle swarm optimizer with simple tabu search function, Proc. of NOLTA, pp. 376–379, 2009.
- [17] R. Sano, T. Shindo, K. Jin’no, and T. Saito, PSO-based multiple optima search systems with switched topology, in Proc. IEEE Congress Evol. Comput., pp. 3301–3307, 2012.

- [18] Z. Sevkli and F. E. Sevilgen, Discrete particle swarm optimization for the orienteering problem, in Proc. IEEE Congress Evol. Comput., pp. 1937–1944, 2010.
- [19] M. Kubota and T. Saito, A discrete particle swarm optimizer for multi-solution problems, IEICE Trans. Fundamentals, E95-A, 1, pp. 406–409, 2012.
- [20] K.-B. Lee and J.-H. Kim, Mass-spring-damper motion dynamics-based particle swarm optimization, in Proc. IEEE Congress Evol. Comput., pp. 2348–2353, 2008.
- [21] K. Jin’no and T. Shindo, Analysis of dynamical characteristic of canonical deterministic PSO, in Proc. IEEE Congress Evol. Comput., pp. 1105–1110, 2010.
- [22] K. Maruyama, R. Sano and T. Saito, Deterministic discrete particle swarm optimizers with collision and insensitivity, Proc. NDES, pp. 245–248, 2012.
- [23] K. Maruyama and T. Saito, Deterministic Particle Swarm Optimizers with Collision for Discrete Multi-solution Problems, Proc. IEEE/SMC, pp. 1335–1340, 2013.



# 第2章 鈍感さと衝突を伴う 決定論的離散粒子群最適化法

## 2.1 導入

粒子群最適化法 (particle swarm optimizers, PSO) は生物群の行動を基にしたアルゴリズムであり、群知能の一種である。粒子は潜在解であり、自身と群れの履歴に応じて目標の解を探索する [1]。PSO はシンプルな概念であり、様々な工学的応用が可能である：画像処理 [2]、デジタルフィルタ [3]、人工ニューラルネット [4]、電力システム [5]、再生可能エネルギー [6]、など。

本章では複数解問題 (multi-solution problems, MSP) のための新しい離散決定論的粒子群最適化法 (discrete deterministic particle swarm optimizers, DDPSO) を考察する [7] [8]。基本的な PSO は単一解を持つ最適化問題のために開発されたが、DDPSO は MSP を求めるための固有の能力と適切なパラメータの選択肢を持っている。DDPSO は以下の特徴を持っている。第一に DDPSO は決定論的な差分方程式に支配され力学系理論に基づき解析することが可能である [9] [10]。通常の PSO は確率論的なパラメータを含み、その安定性の解析が困難である [9]。第二に、粒子群はリング結合をとり、離散探索空間を移動する。離散力学系は実験の再現性とハードウェア実装に適している [11] [12]。第三に、粒子間の情報共有には鈍感さ [13] を持ち、また粒子間では衝突が発生する。この鈍感さと衝突は適切なパラメータを選択することで、MSP のために効果的な粒子群の多様性を与える。これは鈍感さと衝突を伴う PSO に関する最初の論文である。基本的な数値実験を基にアルゴリズムの効率を調査する。

## 2.2 PSO アルゴリズム

MSP のためのアルゴリズムを説明する．離散探索空間  $S_D$  における，2 次の正定の目的関数  $F$  は式 (2.1) に定義される：

$$\begin{aligned} F(\mathbf{x}), \mathbf{x} &\equiv (x_1, x_2), x_1 \in S_D, x_2 \in S_D \\ S_D &\equiv \{X_{min}, X_{min} + d, \dots, X_{max} - d, X_{max}\} \\ d &\equiv (X_{max} - X_{min})/L_p \end{aligned} \quad (2.1)$$

ただし， $X_{max} > X_{min}$ ， $S_D$  は  $(L_p + 1)^2$  個の格子点で構成される．

$F$  は複数解  $\mathbf{x}_{si}$  を含むことを前提とする：

$$F(\mathbf{x}_{si}) \approx 0, \mathbf{x}_{si} \in S_D, i = 1, 2, \dots, M \quad (2.2)$$

ただし， $M \geq 2$ ．

全ての解を検索するために，DDPSO は  $N$  個の粒子を使用する． $i$  番目の粒子  $P_i$  は位置  $\mathbf{x}_i$  と速度  $\mathbf{v}_i$  で特徴づけられる．ただし， $i = 1, \dots, N$ ， $N$ ：粒子数．粒子の位置  $\mathbf{x}_i$  は潜在解である．位置の更新にはパーソナルベスト ( $Pbest_i$ ) とローカルベスト ( $Lbest_i$ ) を基に行う． $Pbest_i$  は粒子  $P_i$  がこれまでに発見した最も優れた適合度である． $Lbest_i$  は近傍の粒子の中で最も優れた  $Pbest_i$  により与えられる．近傍の粒子は粒子群の構造によって決定される．本章ではリング結合を用いる (図 2.1)．リング結合 (図 2.1 (b)) では両側の粒子が近傍の粒子であり，全結合 (図 2.1 (a)) では全ての粒子が近傍の粒子である．アルゴリズムを定義するために，変数を定義する．離散時間を  $t$  とする．これは探索時間である．探索時間  $t$  における粒子群は  $P^t$  で示される． $P_i^t$  は  $i$  番目の粒子である． $\mathbf{x}_i^t \equiv (x_{i1}^t, x_{i2}^t)$  は粒子の位置であり， $\mathbf{v}_i^t \equiv (v_{i1}^t, v_{i2}^t)$  は粒子の速度である．ただし， $i = 1, \dots, N$ ．アルゴリズムを次のように定義する：

**Step 1:**  $t = 0$  とし，粒子群  $P^t$  を初期化する． $\mathbf{x}_i^t$  を  $S_D$  内にランダムに配置し， $\mathbf{v}_i^t$  を  $[-L_p/2, L_p/2]$  の範囲でランダムに初期化する．

**Step 2:** 各粒子の適応度  $F(\mathbf{x}_i^t)$  を計算する． $i$  番目の粒子が式 (2.3) を満たす時，それは近似解の候補として記録される．従って，いずれかの最適解に十分に近づいていることを示唆する．

$$0 < F(\mathbf{x}_i^t) < C_1 \equiv \text{criterion} \quad (2.3)$$

**Step 3:** パーソナルベストとローカルベストを更新する :

$$pbest_i^t = x_i^t \quad \text{if } F(x_i^t) < F(pbest_i^t) \quad (2.4)$$

$$lbest_i^t = pbest_i^t \quad \text{if } F(pbest_i^t) < F(lbest_i^t) - \epsilon \quad (2.5)$$

ただし,  $\epsilon$  は非負の実数である.  $\epsilon$  が増加すれば鈍感さはそれに伴って増加する.  $\epsilon = 0$  の時は最も敏感である. 敏感であればあるほど, 多くの場合, 局所解に陥る可能性がある.

**Step 4:** 粒子の速度と位置を更新する :

$$\begin{aligned} v_i^{t+1} &\leftarrow wv_i^t + c(lbest_i^t - x_i^t) \\ x_i^t &\leftarrow x_i^t + v_i^{t+1} \end{aligned} \quad (2.6)$$

更新後, 粒子は  $x_i^t$  は離散探索空間  $S_D$  内の格子点上に配置される. 試行錯誤の末, 以下のパラメータを使用する: 慣性定数  $w = 0.7$ , 加速度係数  $c = 0.7$ . 古典的な PSO では  $c$  に対し乱数を使用する. しかしながら, DDPSO ではこのような確率論的パラメータを含まないため, 決定論的であると言える.

**Step 5:** 粒子の衝突を行う.  $m$  番目と  $n$  番目の粒子が同一座標上にある ( $x_m^t = x_n^t$ ) ととき, 粒子の衝突が発生する. このとき, ローカルベスト  $lbest_i$  の値が大きい粒子が移動する (図 2.2).

$$\begin{cases} x_m^{t+1} \leftarrow x_m^t + kv_n^t \\ x_n^{t+1} \leftarrow x_n^t \end{cases} \quad \text{if } F(lbest_m^t) > F(lbest_n^t) \quad (2.7)$$

$$\begin{cases} x_m^{t+1} \leftarrow x_m^t \\ x_n^{t+1} \leftarrow x_n^t + kv_m^t \end{cases} \quad \text{if } F(lbest_n^t) > F(lbest_m^t) \quad (2.8)$$

ただし,  $k$ : 衝突パラメータ. ローカルベストが等しい場合, 粒子番号  $i$  が最小の粒子が移動する. 3 つ以上の粒子が衝突した場合, その中で最大のローカルベストの値を持つ粒子のみ移動する.

**Step 6:** 最大探索時間  $t = t_{max}$  を満たすとき探索終了とする. そうでなければ,  $t \rightarrow t+1$  に更新し, **Step 2** に戻る.

DDPSO の擬似コードを付録 A.1 に示す.

## 2.3 数値実験

アルゴリズムの性能を調査するために，2次元力学系 (2D-maps) を定義する：

$$\begin{cases} x_1(n+1) = F_1(x_1(n), x_2(n)) \\ x_2(n+1) = F_2(x_1(n), x_2(n)) \end{cases} \quad (2.9)$$

ただし， $n$ ：離散時間．

2次元力学系には様々な候補が存在するが，本章では池田写像 (Ikeda map) を採用する：

$$\begin{aligned} F_1(x_1, x_2) &= u(x_1 \cos \tau - x_2 \sin \tau) + 1 \\ F_2(x_1, x_2) &= u(x_1 \sin \tau + x_2 \cos \tau) \end{aligned} \quad (2.10)$$

$$\tau = 0.4 - \frac{6}{1 + x_1^2 + x_2^2}$$

ただし  $u = 0.9$ ， $x_i \in S_D$ . 便宜上， $X_{max} = -X_{min} = 5$ ， $L_p = 128$  とする．

目的関数はチェビシェフ距離 (Chebyshev distance) を用いる：

$$f_{\text{Ikeda2}}(x_1, x_2) = \max\{|F_1^2(x_1, x_2) - x_1|, |F_2^2(x_1, x_2) - x_2|\} \quad (2.11)$$

ただし， $F_1^m$ ： $F_1$  の  $m$  重合成写像．また，付録 B の図 B.1 に  $f_{\text{Ikeda2}}(x_1, x_2)$  の等高線図および解を示す．周期点は式 (2.11) を満たす 5 つの解に対応する．

試行錯誤の末，パラメータは以下のように定めた：

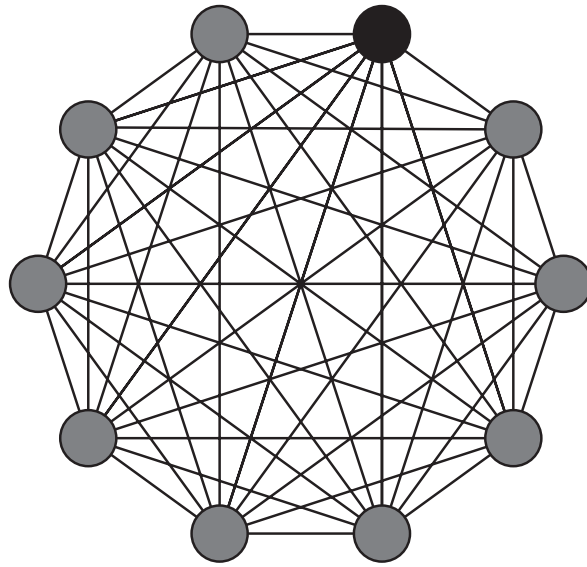
$$N = 50, C_1 = 0.25, \epsilon = 0.20, k = 16.$$

図 2.3 は探索過程の典型例である．全ての近似解を発見することを目的としている．まず， $t = 0$  で，50 個の粒子を探索領域の左側に寄せて初期配置した (図 2.3 (a))．この実験で，鈍感さと衝突はアルゴリズムに効果的に作用し，5 個の近似解 (AS1,  $\dots$ , AS5) を発見した．この探索過程を図 2.4 に要約する．また，付録 B に追加のデータを示す．

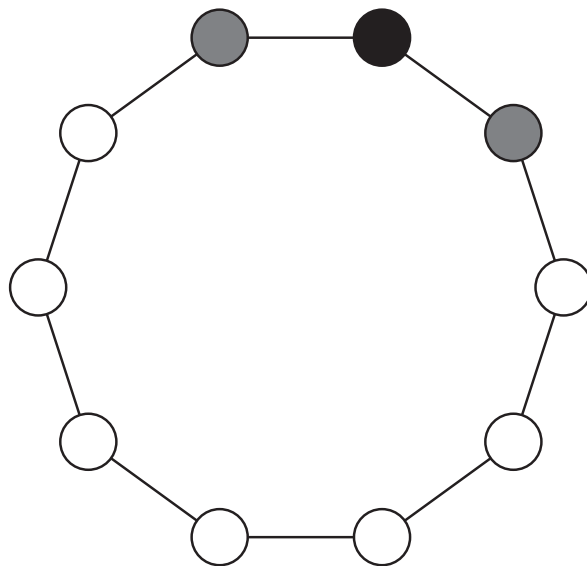
粒子の初期配置と探索性能との関係性を調べるために，3 種類のパラメータの組み合わせを用意した (表 2.1)．タイプ A では粒子を探索領域の左側に寄せて初期配置した (図 2.3 (a))．タイプ B と C は粒子をランダムに一樣に配置した．成功率 (success rate, SR) を表 2.2 に示す．DDPSO は 5 個の近似解の内，少なくとも 4 個の近似解を発見する可能である．しかしながら，粒子の一樣な初期配置，多数の粒子数は探索に有利に働くことが確認された．

## 2.4 まとめ

本章では鈍感さと衝突を有する DDPSO を提案した．典型的な MSP に DDPSO を適用することで，鈍感さと衝突が効果的に働くことを確認した．今後の課題として，探索過程の分析，最適なパラメータ設定，実用的な工学的問題への適用が挙げられる．

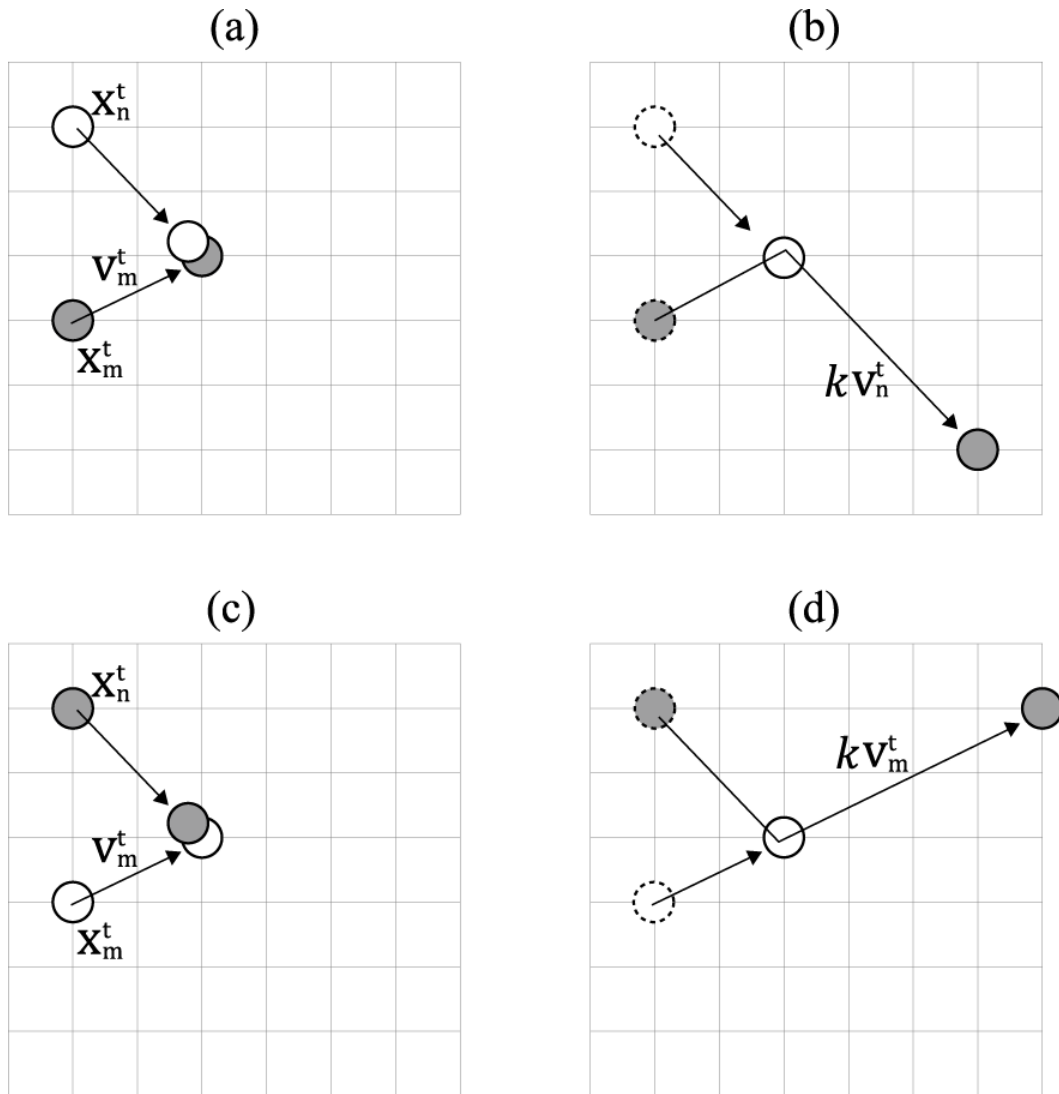


(a) Star structure



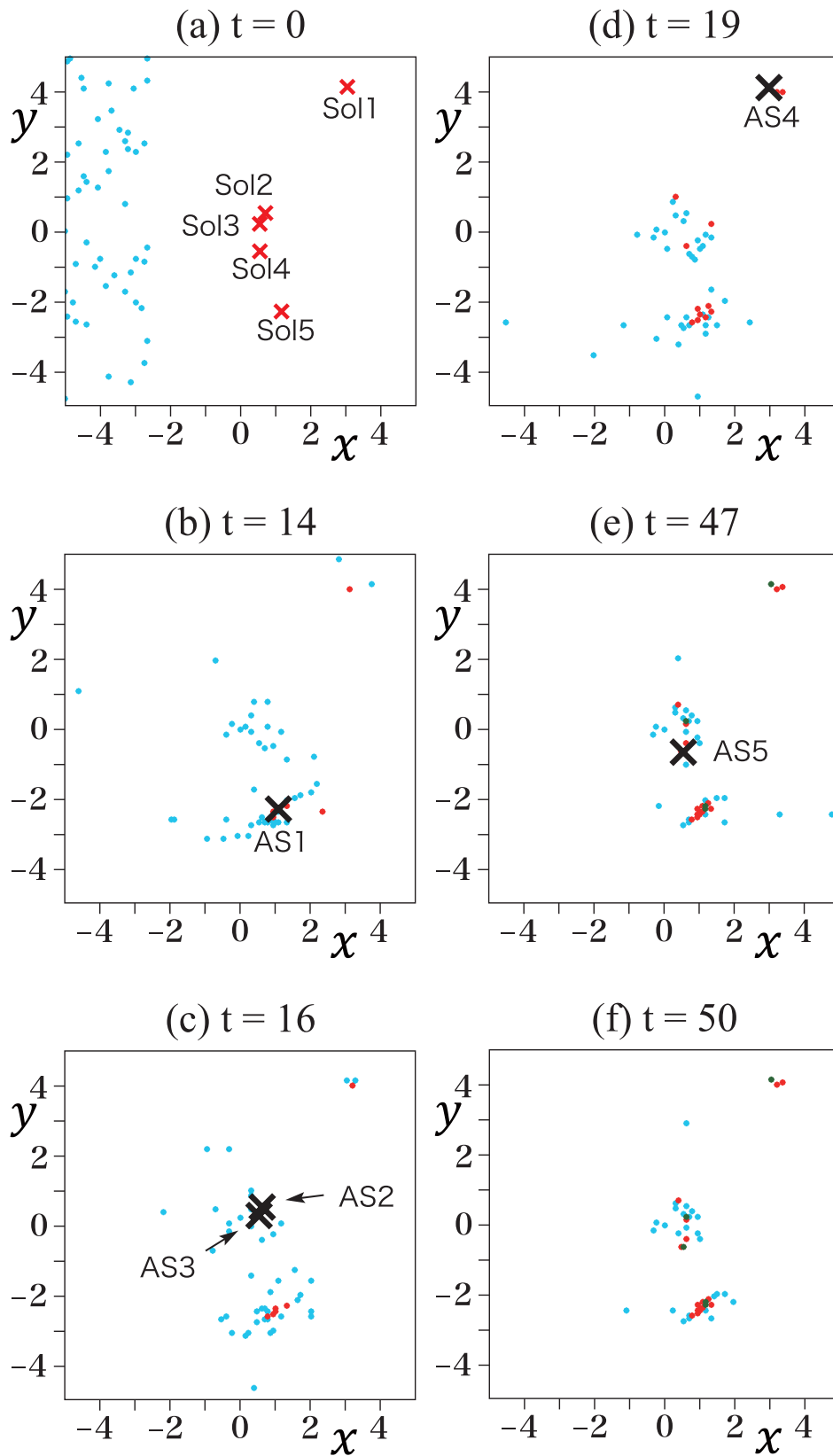
(b) Ring structure

☒ 2.1: Social network structures.



⊗ 2.2: Collision of particles  $P_m$  and  $P_n$ .

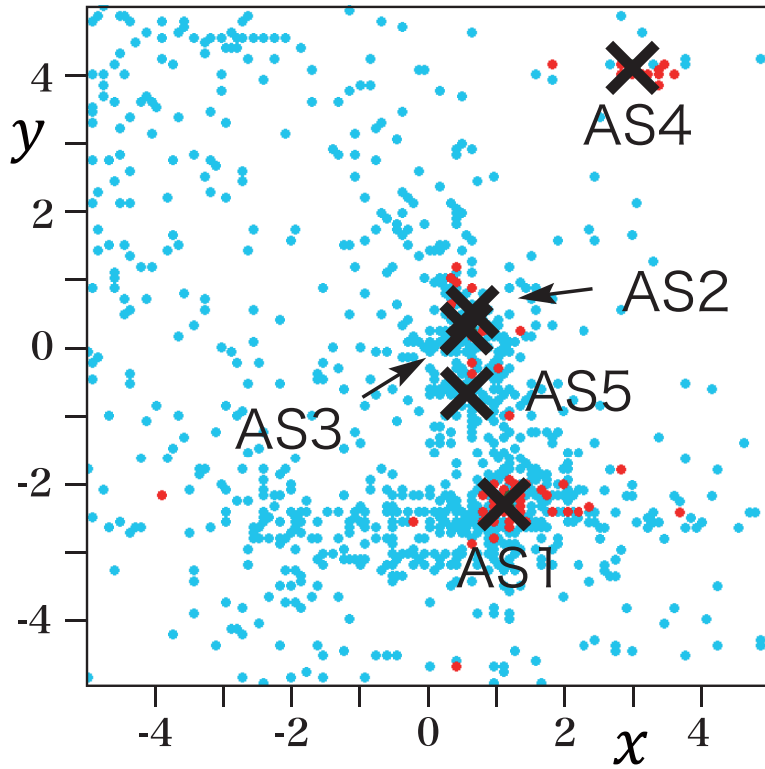
(a) and (b): If local best of  $P_m$  is larger than that of  $P_n$  ( $lbest_m > lbest_n$ ) then  $P_m$  jumps (Eq. (2.7)). (c) and (d): Inverse case ( $lbest_n > lbest_m$ ) (Eq. (2.8)).



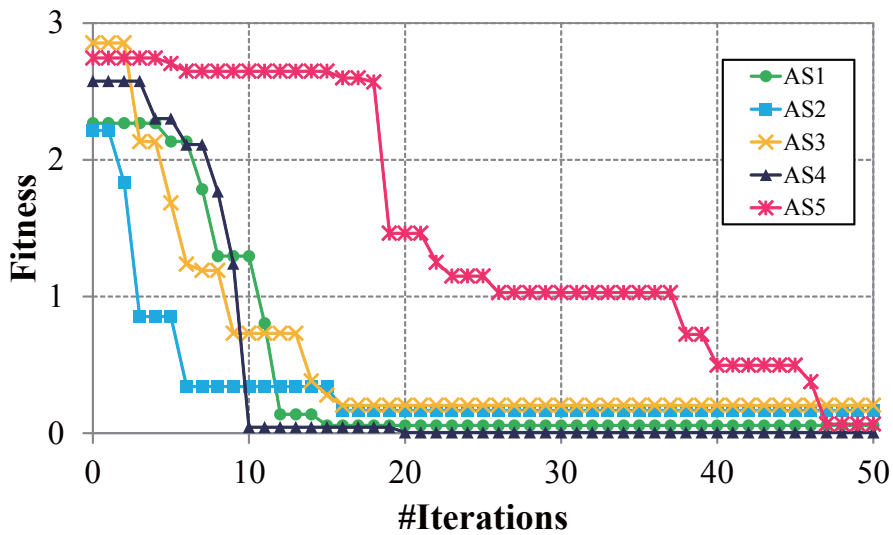
☒ 2.3: Search process for  $C_1 = 0.25$ ,  $k = 16$ ,  $\epsilon = 0.2$  and  $N = 50$ .

In (a), 5 red crosses Sol1 to Sol5 are the solutions each of which is the closest lattice point to each solution. Red points are candidates of approximate solutions. The black crosses are 5 approximate solutions (AS1 to AS5) for 5 solutions (Sol1 to Sol5): AS1 at  $t = 14$ , AS2 and AS3 at  $t = 16$ , AS4 at  $t = 19$  and AS5 at  $t = 47$ . Light blue points are particle positions at time  $t$ .





(a) Positions of  $N = 50$  particles are overwritten for  $0 \leq t \leq t_{max}$ . 5 approximate solutions are found.



(b) Search process of 5 particles for 5 approximate solutions.

☒ 2.4: Search for  $C_1 = 0.25$ ,  $k = 16$ ,  $\epsilon = 0.2$ ,  $t_{max} = 50$ .

表 2.1: Setting of parameters.

parameter	Type A	Type B	Type C
粒子の初期配置	左寄せ	一様	一様
$N$	50	50	100
$t_{max}$	50	50	50
$\epsilon$	0.2	0.2	0.2
$k$	16	16	16
trial number	4000	4000	4000

表 2.2: SR of DDPSO.

#Solutions	Type A	Type B	Type C
5 個以下	17%	20%	38%
4 個以下	100%	99%	99%
3 個以下	100%	100%	100%

## 参考文献

- [1] A. P. Engelbrecht, *Fundamentals of computational swarm intelligence*, Willey, 2005.
- [2] M. P. Wachowiak, R. Smolikova, Y. Zheng, J. M. Zurada and, A. S. Elmaghraby, An approach to multimodal biomedical image registration utilizing particle swarm optimization, *IEEE Trans. Evol. Comput.*, 8, 3. pp. 289–301, 2004.
- [3] F. Teixeira and A. Romariz, Digital filter arbitrary magnitude and phase approximations-statistical analysis applied to a stochastic-based optimization approach, in *Proc. IEEE Congress Evol. Comput.*, pp. 4089–4096, 2008.
- [4] A. B. van Wyk and A. P. Engelbrecht, Overfitting by PSO trained feedforward neural networks, in *Proc. IEEE Congress Evol. Comput.*, pp. 2672 – 2679, 2010.
- [5] H. Qin, J. W. Kimball and G. K. Venayagamoorthy, Particle swarm optimization of high-frequency transformer, in *Proc. 36th Annu. Conf. IEEE Ind. Electron. Soc.*, pp. 2908–2913, 2010.
- [6] H. Matsushita and T. Saito, Application of particle swarm optimization to parameter search in dynamical systems, *NOLTA, IEICE*, E94-N, 10, pp. 458–471, 2011.
- [7] K. E. Parsopoulos and M. N. Vrahatis, On the computation of all global minimizers through particle swarm optimization, *IEEE Trans. Evol. Comput.*, 8, 3, pp. 211–224, 2004.

- [8] S. Yang and C. Li, A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, *IEEE Trans. Evol. Comput.*, 14, 6. pp. 959–974, 2010.
- [9] K. Jin’no and T. Shindo Analysis of dynamical characteristic of canonical deterministic PSO, in *Proc. IEEE Csongress Evol. Comput.*, pp. 1105–1110, 2010
- [10] R. Sano, T. Shindo, K. Jin’no, and T. Saito, PSO-based multiple optima search systems with switched topology, in *Proc. IEEE Csongress Evol. Comput.*, to appear, 2012.
- [11] Z. Sevkli and F. E. Sevilgen, Discrete particle swarm optimization for the orienteering problem, in *Proc. IEEE Congress Evol. Comput.*, pp. 1937–1944, 2010.
- [12] M. Kubota and T. Saito, A discrete particle swarm optimizer for multi-solution problems, *IEICE Trans. Fundamentals*, E95-A, 1, pp. 406–409, 2012.
- [13] I. Handa and T. Saito, Insensitive differential evolution and multi-solution problems, in *Proc. ECTA*, pp. 292–295, 2011.

# 第3章 決定論的衝突粒子群最適化法と 離散複数解問題

## 3.1 導入

粒子群最適化法 (particle swarm optimizers, PSO) は生物群の行動を基にしたアルゴリズムであり、群知能の一種である [1]。粒子は潜在解に対応し、群れを構築する。群れの中で粒子は互いに情報交換し、目的関数の最適解を探索する。PSO は、その概念とアルゴリズムの簡潔さ、動作の柔軟さ、などから様々なシステムの最適化に応用されている。例として、信号処理 [2]–[5]、人工ニューラルネットワーク [6]、パワーコンバータのスイッチング制御 [7]–[9]。

本章では、粒子間の衝突を有する新しい粒子群最適化法 (particle swarm optimizer with inter-particle collision, CPSO) を提案し、その基本的な機能を調査する。

CPSO には3つの重要なポイントがある。第一に、粒子の動きは離散探索空間において、決定論的な差分方程式により支配される [10] [11]。CPSO は決定論的力学系の一種とみなすことが可能である。このような決定論的なシステムは再現性と実装の観点から有利である [12]–[15]。CPSO の研究は群知能と非線形力学系を結び、新たな架け橋と成り得る。第二に、粒子間の衝突は力学を参考にした、シンプルな法則で制御される。衝突は探索空間内における粒子の位置に依存して発生するので、離散探索空間は衝突の判定を行う面で有効である。衝突は粒子群の多様性を拡大し、局所解、あるいは一部の最適解に粒子群が陥ること (トラップ) を避けるために効果的である。通常の PSO では、これらを要因とする探索の停滞が解の発見を妨げる。第三に、離散複数解問題 (discrete multi-solution problems, DMSP) へ CPSO を適用する。DMSP の目的は、解として基準を満たす離散解領域 (discrete solution regions, DSR) を発見することであ

る．簡単のため，DMSP の例として Hénon map の周期点を使用し，周期点と DSR を対応づけて考察する．衝突は局所解，あるいは一部の DSR へのトラップを回避し，すべての DSR を見つけるために効果的に作用する．Hénon map は典型的な離散時間力学系であり，様々な周期現象 / カオス現象を呈する [16]．周期点の探索は分岐現象の解析の基礎である．この DMSP は非線形力学系解析に向けた，新たなアプリケーションの第一歩に成り得る．アルゴリズムの性能は 3 つの特徴量にもとづいて検討される：成功率，平均探索時間，衝突回数．これらの結果は DMSP への効率的なアルゴリズム開発のための基本的な情報となる．

なお，DMSP はアナログ複数解問題 (analog multi-solution problems, AMSP) とは異なる点に留意すべきである [17]–[19]．AMSP は連続探索空間内の全ての解を発見し，様々な実用的 / 潜在的な応用に対応付けられる．AMSP を解決するために，興味深いアルゴリズムが研究されている．例として，タブー探索法 [17]，群知能 [18]，多様なトポロジーの PSO [19]–[21]．これら既存のアルゴリズムは DMSP への適用は不可能であり，CPSO との比較は困難である．しかしながら，CPSO の基本的な結果は AMSP のためのアルゴリズムの改善に寄与することが可能である．著者らは従来研究において CPSO を研究してきた [22]．その CPSO は人工的な衝突アルゴリズムを使用しているが，本章は異なる衝突アルゴリズムを適用した新規の CPSO であることを強調する．

## 3.2 離散複数解問題

CPSO の目的関数を定義する．

$$\begin{aligned}
 F_D(\mathbf{x}) &: S_D \rightarrow \mathbf{R}_+ \\
 S_D &= \{\mathbf{x} \mid (x_i - X_{Li}) \in \{d, \dots, Md\}, i = 1, \dots, D\} \\
 \mathbf{x} &\equiv (x_1, \dots, x_D)
 \end{aligned} \tag{3.1}$$

ただし， $\mathbf{R}_+$  は非負実数， $S_D$  は  $D$  次元の離散探索空間である． $X_{Li}$  は  $S_D$  の  $i$  次元目の左端， $M$  は次元あたりの格子点数， $d$  格子点の間隔である． $S_D$  は  $M^D$  個の格子点で分割される．

次に，DSR を定義する．

$$A_s^i \equiv \{\mathbf{x} \mid F_D(\mathbf{x}) < C_A, \|\mathbf{x} - \mathbf{c}_s^i\| < \rho, \mathbf{x} \in S_D\} \quad (3.2)$$

ただし， $i = 1, \dots, N_A$ ， $N_A$ ：DSR の数， $C_A$ ：近似解基準， $\|\cdot\|$ ：ユークリッド距離， $\mathbf{c}_s^i$ ：DSR の中心座標， $\rho$ ： $\mathbf{c}_s^i$  からの近似解半径． $A_s^i$ ： $i$  番目の DSR であり，DSOL に含まれる．DSOL は近似解基準  $C_A$  を満たし，各 DSR の中心  $\mathbf{c}_s^i$  から半径  $\rho$  の内側である条件を満たす必要がある (図 3.1)．DMSP の目的は全ての DSOL から，少なくともひとつの DSR を発見することである．これが CPSO の目的である．

DMSP は4つのパラメータ  $M$ (格子点数)， $D$ (次元数)， $C_A$ (近似解基準)， $\rho$ (DSR 半径) から特徴づけられる．格子点数  $M$ ，次元数  $D$  が増加するにつれて総当たり計算 (brute force) での計算は次第に不可能になってゆく．特に， $D$  が増加するに連れて格子点の数が指数関数的に増加し，現実的な時間内での総当たり計算は不可能になる．これは我々が CPSO を研究する理由である．

ここで，DMSP とは異なる，AMSP の簡単な説明を行う．AMSP では目的関数は次のように定義される：

$$\begin{aligned} F_A : S_A &\rightarrow \mathbf{R}_+, \\ S_A &= \{\mathbf{x} \mid X_{Li} \leq x_i \leq X_{Ri}, i = 1, \dots, D\} \end{aligned} \quad (3.3)$$

ただし， $S_A$ ：連続探索空間．正定の関数  $F_A$  は次のように定義される：

$$F_A(\mathbf{x}_s^i) = 0, \mathbf{x}_s^i \equiv (x_{s1}^i, \dots, x_{sD}^i) \in S_A, i = 1, \dots, N_s \quad (3.4)$$

ただし， $\mathbf{x}_s^i$ ： $i$  番目の解， $N_s$ ：解の総数．解は最適点に対応する．AMSP の目的は  $S_A$  内の全ての解を発見することである．AMSP は CPSO の解析対象にならないことに注意したい．また，DMSP は AMSP の近似ではなく，上下関係も存在しない．AMSP を離散化すれば DMSP を得ることが可能であり，DMSP を補間すれば AMSP を得ることが可能である．我々は AMSP に基づく DMSP の例題を構築することが可能であり，最適点は DSR の中心  $\mathbf{x}_s^i$  である．

### 3.3 CPSO アルゴリズム

CPSO を定義するために、変数を以下に定義する。離散時間を  $t$  とする。これは探索時間である。探索時間  $t$  における粒子群は  $P^t \equiv (P_1^t, \dots, P_N^t)$  で定義される。ただし、 $N$  は粒子数。  $i$  番目の粒子  $P_i^t$  は位置  $\mathbf{x}_i^t \equiv (x_{i1}^t, \dots, x_{iD}^t)$  と速度  $\mathbf{v}_i^t \equiv (v_{i1}^t, \dots, v_{iD}^t)$  により特徴づけられる。ただし、 $i = 1, \dots, N$ 。位置  $\mathbf{x}_i^t$  は潜在解である。粒子の更新はパーソナルベスト ( $Pbest_i$ ) とローカルベスト ( $Lbest_i$ ) を基に行う。パーソナルベスト  $Pbest_i = F_D(\mathbf{pbest}_i^t)$  は目的関数  $F_D(\mathbf{x}_i^t)$  において、その粒子がこれまでに発見した中で最良の適応度である。ローカルベスト  $Lbest_i = F_D(\mathbf{lbest}_i^t)$  は  $P_i^t$  の近傍の粒子の中で最良のパーソナルベストにより与えられる。近傍の粒子は粒子群のトポロジーに応じて決定される [20] [21]。簡単のため、両隣の粒子が近接する、リングトポロジーを利用する。CPSO は以下の 6 ステップで定義される。

**Step 1:**  $t = 0$  とし、粒子  $P^t$  を初期化する。  $\mathbf{x}_i^t$  を  $S_D$  内にランダムに配置し、  $\mathbf{v}_i^t$  を  $\{\mathbf{v}_i^t \mid -\frac{1}{4}Md \leq v_{ij}^t \leq \frac{1}{4}Md, d = 1, \dots, N\}$  の範囲でランダムに初期化する。

**Step 2:** 各粒子の適応度を目的関数  $F_D(\mathbf{x}_i^t)$  により評価する。  $j$  番目の粒子の座標が  $i$  番目の DSR に存在するとき：

$$\mathbf{x}_j^t \in A_s^i \tag{3.5}$$

このとき、  $\mathbf{x}_j^t$  は DSOL として記録され、  $i$  番目の DSR は発見されたと宣言される。全ての DSRs を発見したとき、探索は終了する。

**Step 3:**  $\mathbf{pbest}$  と  $\mathbf{lbest}$  を更新する。

$$\mathbf{pbest}_i^t \leftarrow \mathbf{x}_i^t \quad \text{if } F_D(\mathbf{x}_i^t) < F_D(\mathbf{pbest}_i^t) \tag{3.6}$$

$$\mathbf{lbest}_i^t \leftarrow \mathbf{pbest}_j^t \quad \text{if } F_D(\mathbf{pbest}_j^t) < F_D(\mathbf{lbest}_i^t) \tag{3.7}$$

ただし、  $\mathbf{pbest}_j^t$  は  $i$  番目とその近傍の粒子のパーソナルベストである： $j \in \{i-1, i, i+1\}$  であり、粒子群はリングトポロジーを用いる。



**Step 4:** 粒子の速度と位置を更新する .

$$\begin{aligned} \mathbf{v}_i^t &\leftarrow w\mathbf{v}_i^t + c(\mathbf{lbest}_i^t - \mathbf{x}_i^t) \\ \mathbf{x}_i^t &\leftarrow \mathbf{x}_i^t + \mathbf{v}_i^t \end{aligned} \quad (3.8)$$

ただし ,  $w$  と  $c$  は決定論的なパラメータである . 従来の PSO は確率論的パラメータを含む [1] . 式 (3.8) の適用後 , 粒子の新しい位置  $\mathbf{x}_i^t$  は離散探索空間  $S_D$  内の格子点上に離散化される .

**Step 5:** 粒子間の衝突 .  $N_i(l)$  は  $i$  番目の粒子の  $l$  近傍の領域である (図 3.2) .  $m$  番目と  $n$  番目の粒子の近傍に共通部分が存在するとき :

$$N_m(l) \cap N_n(l) \neq \emptyset \quad (3.9)$$

その後 , 粒子の衝突処理が行われ , 速度を更新する :

$$\begin{aligned} \mathbf{v}_m^t &\leftarrow ((\mathbf{v}_n^t - \mathbf{v}_m^t) \cdot \hat{\mathbf{c}}) \hat{\mathbf{c}} + \mathbf{v}_m^t \\ \mathbf{v}_n^t &\leftarrow ((\mathbf{v}_m^t - \mathbf{v}_n^t) \cdot \hat{\mathbf{c}}) \hat{\mathbf{c}} + \mathbf{v}_n^t \end{aligned} \quad (3.10)$$

ただし , “ $\cdot$ ” は内積を示し ,  $\hat{\mathbf{c}}$  は 2 つの粒子の位置から導出された単位ベクトルである :

$$\hat{\mathbf{c}} = \frac{\mathbf{x}_n^t - \mathbf{x}_m^t}{|\mathbf{x}_n^t - \mathbf{x}_m^t|}$$

速度を基に , 位置が更新される :

$$\begin{aligned} \mathbf{x}_m^t &\leftarrow \mathbf{x}_m^t + \mathbf{v}_m^t \\ \mathbf{x}_n^t &\leftarrow \mathbf{x}_n^t + \mathbf{v}_n^t \end{aligned} \quad (3.11)$$

式 (3.9) において ,  $l = 0$  は  $m$  番目と  $n$  番目の粒子の位置が同じであることを意味する . 探索空間  $S_D$  の境界に粒子  $P_i^t$  が到達したとき , 粒子は境界面に反射する . 離散的な探索空間は , 衝突を定義する観点から便利である . 粒子が剛体である場合はより複雑な定義が必要となる .

**Step 6:** 最大探索時間  $t = t_{max}$  を満たすまで ,  $t \rightarrow t + 1$  に更新し , **Step 2** に戻る .

CPSO の擬似コードを付録 A.2 に示す . 更に , ローカルベスト型 PSO (local-best PSO , LPSO) を定義する . LPSO は CPSO アルゴリズムにおける **Step 5** の衝突を削除することによって得られる . LPSO は CPSO の能力を検討する上で基準となる .

### 3.4 Hénon map への応用

提案したアルゴリズムの基本的な性能を検討するために，DMSP の例題を考える．本章では Hénon map を採用した：

$$\begin{cases} x_1(n+1) = F_1(x_1(n), x_2(n)) \\ x_2(n+1) = F_2(x_1(n), x_2(n)) \end{cases}$$

$$\begin{cases} F_1(x_1, x_2) \equiv 1 - ax_1^2 + x_2 \\ F_2(x_1, x_2) \equiv bx_1 \end{cases} \quad (3.12)$$

ab.  $\mathbf{x}(n+1) = \mathbf{F}(\mathbf{x}(n))$

ただし  $n$  : 離散時間．この写像は典型的な離散力学系として知られており，様々な周期現象 / カオス現象を呈し，様々な分岐現象を示す [16]．

Hénon map の周期点を定義する．点  $\mathbf{p}$  は周期  $k$  の周期点である：

$$\mathbf{p} = \mathbf{F}^k(\mathbf{p}) \text{ and } \mathbf{p} \neq \mathbf{F}^l(\mathbf{p}) \text{ for } 0 \leq l < k$$

ただし， $\mathbf{F}^k$  :  $k$  重の合成写像  $\mathbf{F}$ ．周期点の探索は力学系解析の基礎である．

正定の目的関数を定義する．目的関数の最小値は周期点に対応する．簡単のため， $a = 1.4$ ， $b = 0.3$  の場合を考える．このパラメータの値では，Hénon map は 2 つの 2 周期点と 2 つの不動点を持つ．それらは  $\mathbf{F}^2(\mathbf{x}) - \mathbf{x} = 0$  を満たす 4 つの解として求められる．この式を使用して，我々は AMSP を構築する．

$$\begin{aligned} F_A(\mathbf{x}) &= \|\mathbf{F}^2(\mathbf{x}) - \mathbf{x}\| \\ &= \sqrt{(F_1^2(x_1, x_2) - x_1)^2 + (F_2^2(x_1, x_2) - x_2)^2} \end{aligned} \quad (3.13)$$

$$S_A = \{\mathbf{x} \mid -1.5 \leq x_i \leq 1.5, i = 1, \dots, 4\}$$

ただし， $\|\cdot\|$  : ユークリッド距離．図 3.3 (a) は 4 つの解を示している．アナログ探索空間  $S_A$  を  $M = 128$  として離散化することで我々は離散探索空間  $S_D$  を得る：

$$\begin{aligned} F_D(\mathbf{x}) &= \|\mathbf{F}^2(\mathbf{x}) - \mathbf{x}\| \\ \mathbf{x} \in S_D &= \{\mathbf{x} \mid x_i \in \{-1.5 + d, \dots, 1.5 - d, 1.5\}\} \end{aligned} \quad (3.14)$$

ただし  $i = 1, 2$ ， $d = 3/128$ ．離散探索空間  $S_D$  は  $128^2$  の格子点で構成される．式 (3.14)(3.15) を基として DMSP の最適解を求める．

$$C_A = 0.1, \rho = 0.15 \quad (3.15)$$

この問題は図 3.3 (b) に示す，4 つの DSR を発見することである．アルゴリズムに用いるパラメータは試行錯誤の末に式 3.16 を選択した：

$$N = 70, D = 2, t_{max} = 50, w = 0.7, c = 1.4 \quad (3.16)$$

この場合， $M^D$  個の格子点全てを求めるには総当たり計算 (brute force) が必要である． $t_{max} = M^D/N > 234$  であるため，CPSO の能力を考慮する上でこれらのパラメータ選択は適切である．格子点数  $M$ ，次元数  $D$  が増加するにつれて総当たり計算での計算は次第に不可能になっていく． $M = 128$  と  $D = 2$  は小さい値ではあるが，これらの数値実験の結果は  $M$  と  $D$  が大きい場合でも基本的な情報を提供することが可能である．

初めに，LPSO を DMSP に適用する．図 3.4 は探索過程の概要である．図 3.4 の各図は図 3.3 (b) に対応している．LPSO は 3 つの DSR を  $t = 11$  までに発見したが，残り 1 つの DSR は  $t = t_{max}$  まで発見することが出来なかった．探索時間  $t$  が進むにつれて，粒子は 3 つの DSR に集中する傾向がある (図 3.4 (d))．この粒子の集中は残り 1 つの DSR の発見を妨げる要因である．

次に，CPSO( $l = 8$ ) を DMSP に適用する．図 3.5 は探索過程の概要である．これは LPSO と同様の初期配置，初期速度から探索が行われている．図 3.5 (d) において，粒子が各 DSR の周囲に分布している (同じ座標に重なっているわけではない) ことから，衝突が示されている．CPSO の粒子間の衝突は効果的に作用しており，最後の DSR は  $t = 20$  に発見されている．図 3.6 (c) は粒子の濃度と分散を示唆している．これは粒子の多様性を維持することに対して有効である．

アルゴリズムの特性を考察するため，LPSO(衝突無し)，CPSO( $l = 0, 4, 8, 12$ ) を対象に，それぞれ 1000 回ずつ数値実験を行った．CPSO において， $l$  が増加すると，粒子間衝突の可能性が上昇する．各試行は乱数に応じて，異なる初期配置と初期速度が与えられる．表 3.1，3.2，3.3 は以下の 3 つの特徴量に基づく結果である：

- SR: 平均成功率．全ての DSR を発見した時，探索成功とする．
- #ITE: 探索成功時の平均探索時間．
- #COL: 探索成功時の平均衝突回数．

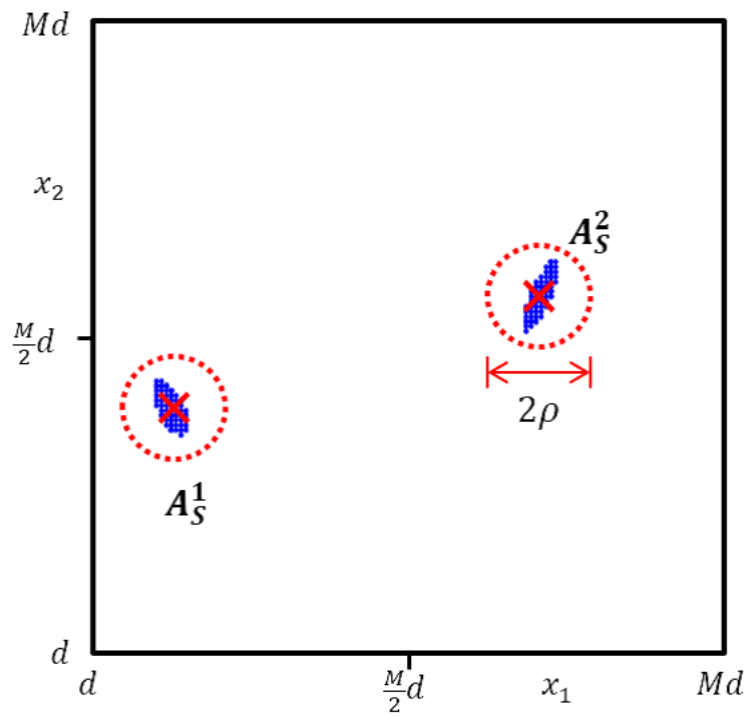
表 3.1 は SR(平均成功率)に関する表であり， $N$ (粒子数)と $l$ (近傍数)の関係性を示している． $N$ の増加に従い SR が向上する．しかしながら， $l$ の増加につれて SR が単調に増加するわけではない．例として， $N = 80$ では $l = 12$ よりも $l = 8$ が SR の最大値を記録する．これは過度に大きい $l$ は必要以上の衝突を引き起こし，DSRs へ粒子が移動することを妨げることを示唆している．従って，問題に応じて適度な $l$ や $N$ を与えることが，高い SR を実現する．

表 3.2, 3.3 は $l$ の上昇につれて#ITE と#COL は向上する．これは粒子の動きが複雑になることを示唆し，粒子は他の DSRs に移動することが可能である． $N$ の増加につれて#COL は増加する．しかしながら，#ITE は著しく変化しない．

### 3.5 まとめ

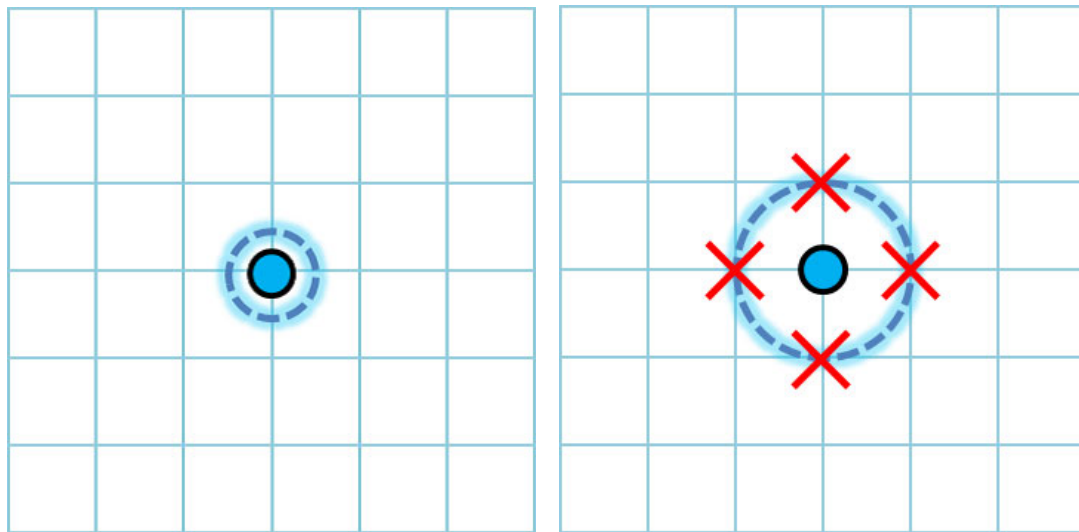
CPSO を提案し，Hénon map を例題とする DMSP に適用した．粒子間の衝突は粒子の多様性に寄与し，全 DSR の発見に効果的である．アルゴリズムの性能は 3 つの特徴量にもとづいて検討を行った：SR，#ITE，#COL．この結果は DMSP のための効果的なアルゴリズム開発実現への基本的な情報である．特に，非線形力学系における周期点探索は，分岐現象 / 安定性解析への応用を今後考察してゆく上で非常に重要である．これは，その応用に向けての第一歩と成り得る．

CPSO には，以下を含む多くの課題がある：探索過程の詳細な分析，パラメータの適切な設定，粒子間の衝突の様々な応用，衝突の効果の評価，大規模問題への適用，実用的な問題への適用．



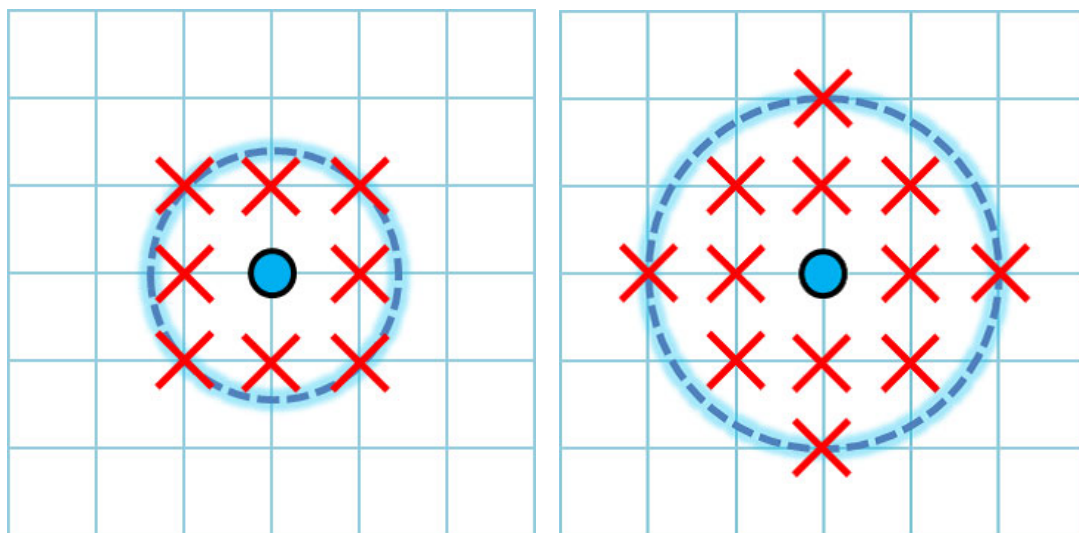
⊠ 3.1: Discrete solution regions (DSRs).

DSR consisting of discrete solutions (blue dots). Dotted circles denote hyper spheres centered at  $c_s^i$  (red crosses) with radius  $\rho$ .



(a) 0-neighbor

(b) 4-neighbor

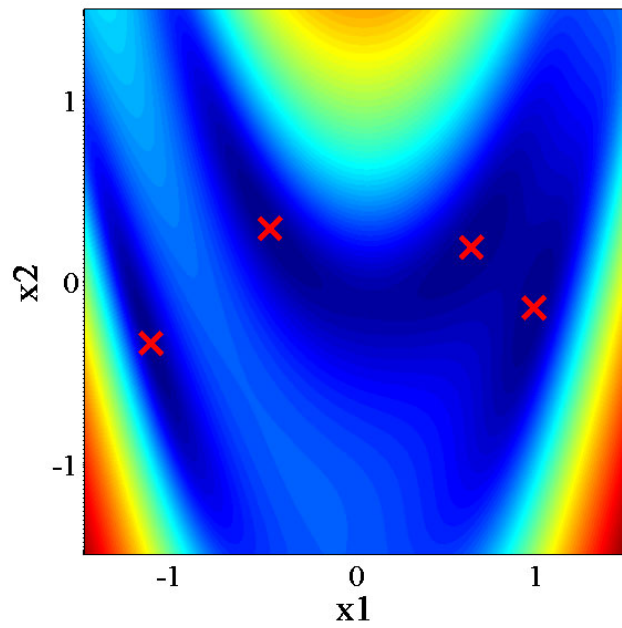


(c) 8-neighbor

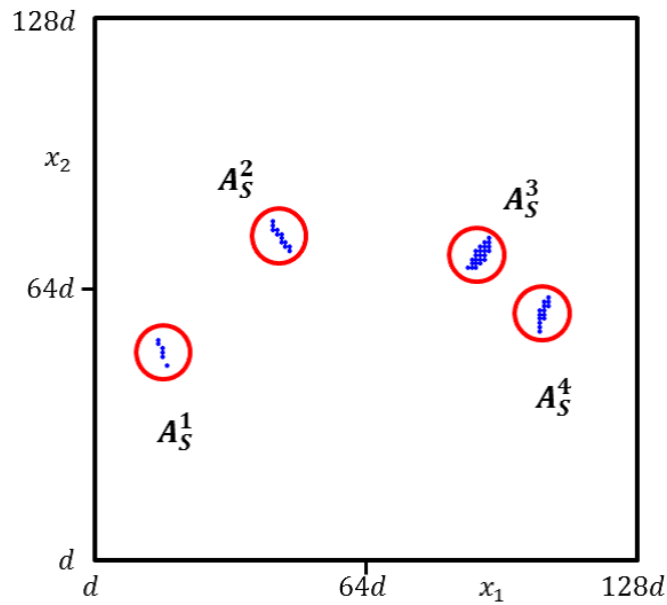
(d) 12-neighbor

☒ 3.2: Definition of  $l$ -neighbors.

The blue circle denotes position of a particle and the red crosses construct its neighbor.

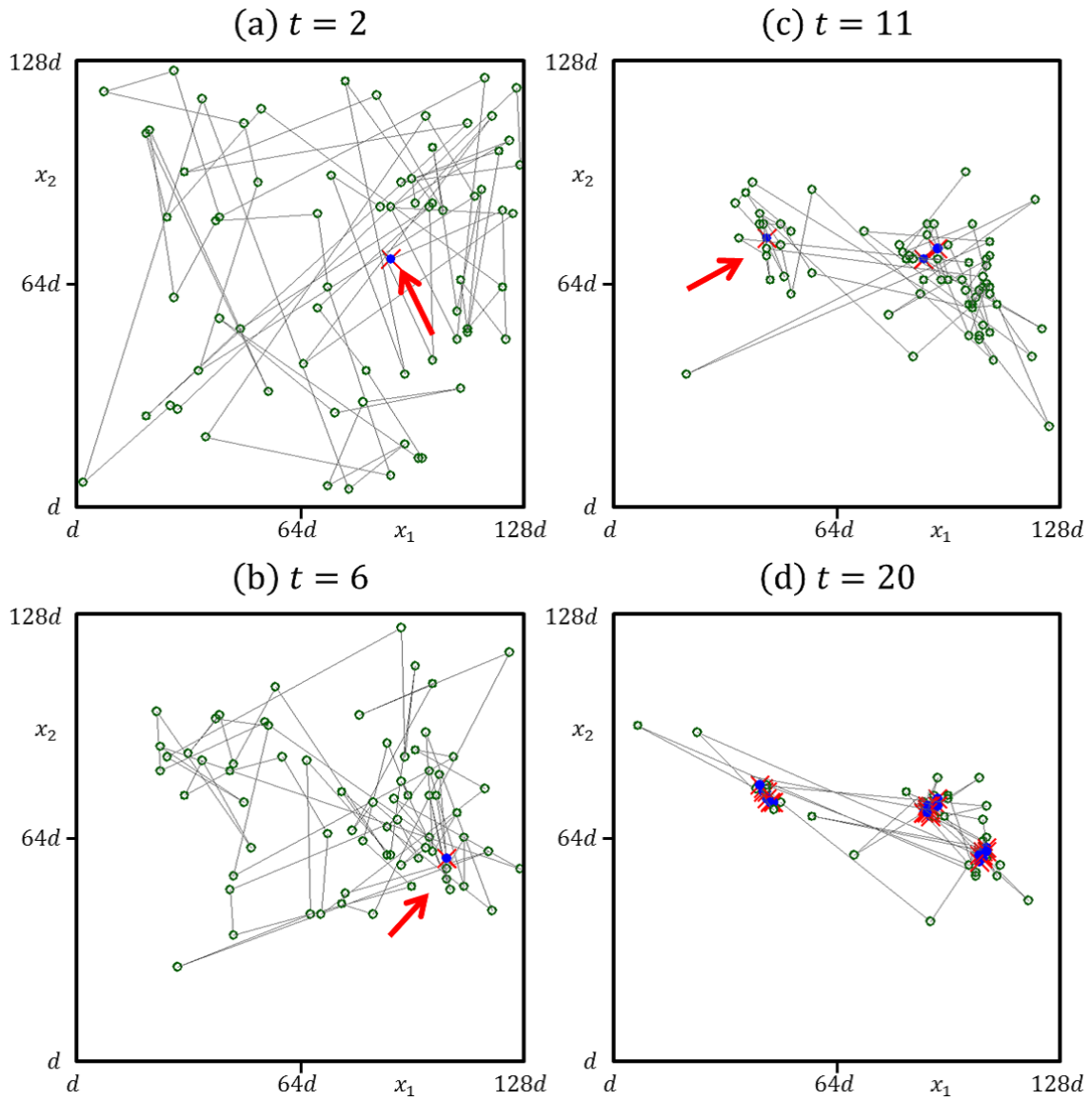


(a) The objective function  $F_A$  for fixed points and 2-periodic points of the Hénon map. In this contour map, colder color denotes lower height of  $F_A$ . The red crosses denote the multi-solution.



(b) Discrete solution regions of  $F_D$ .  $A_S^1$ ,  $A_S^2$ ,  $A_S^3$  and  $A_S^4$  contain 6, 14, 23 and 14 DSOLs, respectively.  $M = 128$ ,  $C_A = 0.1$ .

⊠ 3.3: Discrete multi-solution problem based on the Hénon map.



⊠ 3.4: An exploring process of LPSO for the DMSP.

Snapshots of an exploring process of LPSO for the DMSP based on the Hénon map. The second to fourth DSRs are found as shown in arrows at  $t = 11, 2,$  and  $6,$  respectively. The 1st DSR cannot be found until  $t = t_{max}$ .



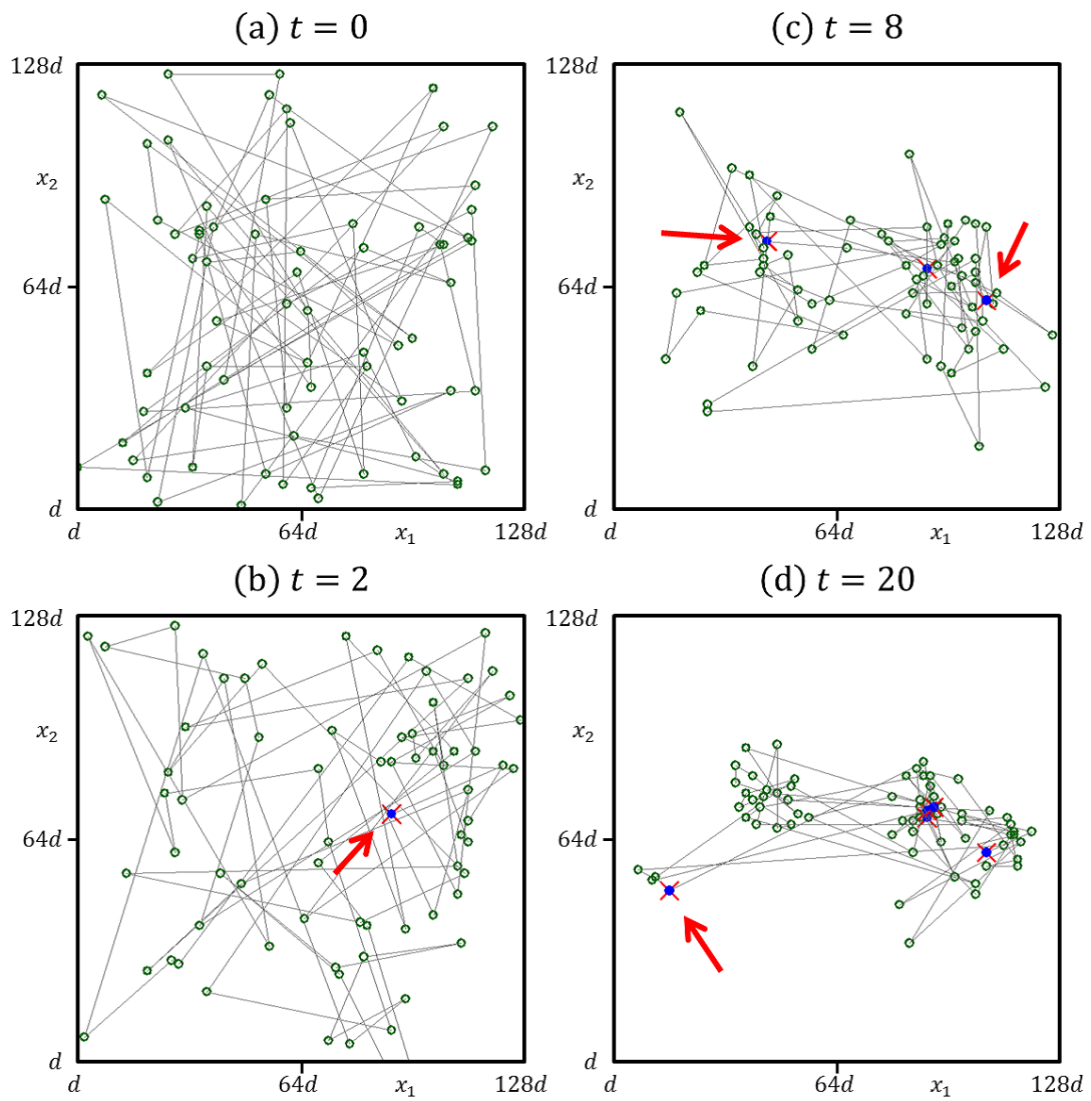
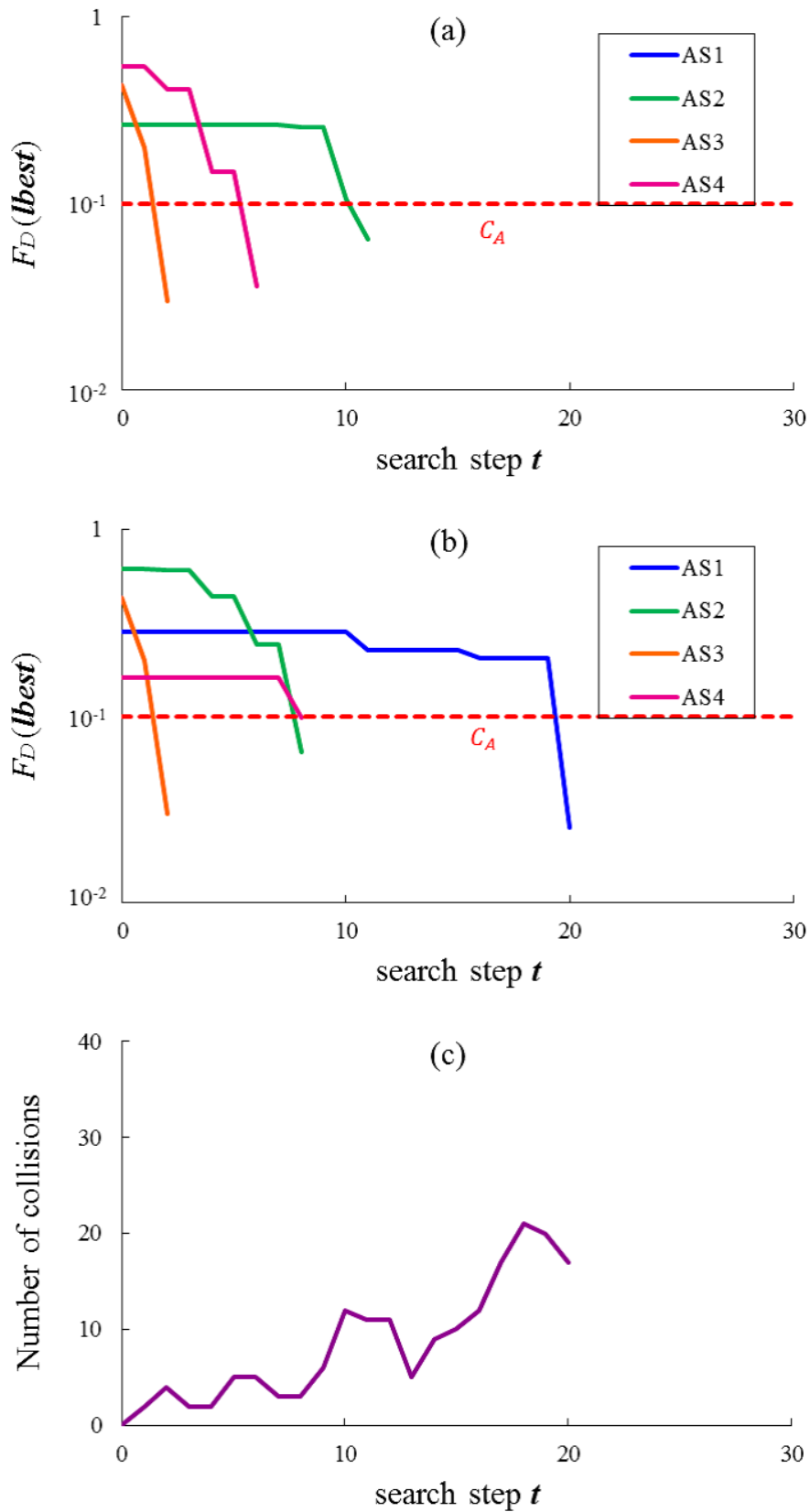


图 3.5: An exploring process of CPSO for the DMSP.

The first to fourth DSRs are found at  $t = 20, 8, 2$  and  $8$ , respectively.



☒ 3.6: Evolution of particles to DSRs of the DMSP.

Graphs of AS1 to AS4 denote  $F_D(\mathit{lbest})$  of particles to the first to fourth DSRs, respectively. (a) LPSO in failure case. The first DSR (AS1) cannot be found. (b) CPSO in successful case. (c) The number of collisions.

表 3.1: The average success rate (SR) of the DMSP based on the Hénon map.

$N$	LPSO	CPSO			
	w/o col.	$l = 0$	$l = 4$	$l = 8$	$l = 12$
40	38.1%	39.3%	42.5%	45.4%	<b>46.2%</b>
50	51.0%	52.2%	54.5%	58.8%	<b>59.0%</b>
60	60.3%	61.8%	65.1%	68.4%	<b>68.7%</b>
70	67.5%	69.3%	72.4%	<b>74.9%</b>	74.4%
80	74.3%	75.4%	78.6%	<b>81.0%</b>	79.2%
90	79.4%	80.9%	83.6%	<b>84.9%</b>	83.7%
100	82.3%	84.8%	86.9%	<b>89.5%</b>	85.8%

表 3.2: The average number of iterations (#ITE) of the DMSP.

$N$	LPSO	CPSO			
	w/o col.	$l = 0$	$l = 4$	$l = 8$	$l = 12$
40	17.8	18.2	19.9	22.9	25.3
50	17.2	17.6	19.4	22.5	24.8
60	16.7	17.1	18.9	22.2	25.6
70	16.3	16.8	18.7	22.2	25.5
80	16.0	16.3	18.6	22.5	26.3
90	15.6	16.1	18.3	22.1	25.8
100	15.2	15.8	17.9	22.5	26.0

表 3.3: The average number of collisions (#COL) of the DMSP.

$N$	LPSO	CPSO			
	w/o col.	$l = 0$	$l = 4$	$l = 8$	$l = 12$
40	N/A	16	58	116	163
50	N/A	22	79	155	216
60	N/A	27	98	193	291
70	N/A	34	121	241	362
80	N/A	39	151	300	459
90	N/A	47	173	342	517
100	N/A	54	196	417	617

## 参考文献

- [1] J. Kennedy and R. Eberhart, Particle Swarm Optimization, Proc of IEEE/ICNN, pp. 1942–1948, 1995.
- [2] M. P. Wachowiak, R. Smolikova, Y. Zheng, J. M. Zurada and, A. S. Elmaghraby, An approach to multimodal biomedical image registration utilizing particle swarm optimization, IEEE Trans. Evol. Comput., 8, 3. pp. 289–301, 2004.
- [3] S.-T. Hsieh, T.-Y. Sun, C.-L. Lin, and C.-C. Liu, Effective learning rate adjustment of blind source separation based on an improved particle swarm optimizer, IEEE Trans. Evol. Comput., 12, 2. pp. 242–251, 2008.
- [4] A. Ghosh, A. Ghosh, A. Chowdhury, A. Konar, E. Kim and A. K. Nagar, Linear phase low pass FIR filter design using genetic particle swarm optimization with dynamically varying neighborhood technique, Proc of IEEE Congress Evol. Comput., pp. 529–535, 2012.
- [5] R. A. Vural, T. Yildirim, T. Kadioglu and A. Basargan, Performance evaluation of evolutionary algorithms for optimal filter design, IEEE Trans. Evol. Comput., 16, 1. pp. 135–147, 2012.
- [6] A. B. van Wyk and A. P. Engelbrecht, Overfitting by PSO trained feedforward neural networks, Proc of IEEE Congress Evol. Comput., pp. 2672–2679, 2010.
- [7] H. Qin, J. W. Kimball and G. K. Venayagamoorthy, Particle swarm optimization of high-frequency transformer, Proc of Annual Conf. IEEE Ind. Electron. Soc., pp. 2908–2913, 2010.

- [8] K. Kawamura and T. Saito, Design of switching circuits based on particle swarm optimizer and hybrid fitness function, Proc of Annual Conf. IEEE Ind. Electron. Soc., pp. 1099–1103, 2010.
- [9] H. Matsushita and T. Saito, Application of particle swarm optimization to parameter search in dynamical systems, NOLTA, IEICE, E94-N, 10, pp. 458–471, 2011.
- [10] Z. Sevkli and F. E. Sevilgen, Discrete particle swarm optimization for the orienteering problem, Proc of IEEE Congress Evol. Comput., pp. 1937–1944, 2010.
- [11] M. Kubota and T. Saito, A discrete particle swarm optimizer for multi-solution problems, IEICE Trans. Fundamentals, E95-A, 1, pp. 406–409, 2012.
- [12] K.-B. Lee and J.-H. Kim, Mass-spring-damper motion dynamics-based particle swarm optimization, Proc of IEEE Congress Evol. Comput., pp. 2348–2353, 2008.
- [13] K. Jin’no and T. Shindo, Analysis of dynamical characteristic of canonical deterministic PSO, Proc of IEEE Congress Evol. Comput., pp. 1105–1110, 2010.
- [14] K. Tamura and K. Yasuda, Quantitative analysis based tuning law for convergence rate of spiral optimization, Proc of IEEE Int’l Conf. Systems, Man, and Cybernetics, pp. 767–772, 2012.
- [15] K. Tatsumi, T. Ibuki and T. Tanino, Chaotic particle swarm optimization with an intensive search around the personal and global bests, Proc of IEEE Int’l Conf. Systems, Man, and Cybernetics, pp. 1333–1338, 2012.
- [16] E. Ott, Chaos in Dynamical Systems, Cambridge Univ. Press, 1993.
- [17] K. E. Parsopoulos and M. N. Vrahatis, On the computation of all global minimizers through particle swarm optimization, IEEE Trans. Evol. Comput., 8, 3, pp. 211–224, 2004.

- [18] S. Yang and C. Li, A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, *IEEE Trans. Evol. Comput.*, 14, 6. pp. 959–974, 2010.
- [19] X. Li, Niching without niching parameters: particle swarm optimization using a ring topology, *IEEE Trans. Evol. Comput.*, 14, 1. pp. 150–169, 2010.
- [20] T. Saito and E. Miyagawa, Growing-tree particle swarm optimizer with simple tabu search function, *Proc of NOLTA*, pp. 376–379, 2009.
- [21] R. Sano, K. Jin'no and T. Saito, Particle swarm optimization with switched topology and deterministic parameters, *Proc of IEEE Int'l Conf. Systems, Man, and Cybernetics*, pp. 530–535, 2012.
- [22] K. Maruyama, R. Sano and T. Saito, Deterministic discrete particle swarm optimizers with collision and insensitivity, *Proc of NDES*, pp. 245–248, 2012.

# 第4章 衝突粒子群最適化法と Hénon mapの周期解探索

## 4.1 導入

粒子群最適化法 (particle swarm optimizers, PSO) は生物群の行動を基にしたアルゴリズムであり, 群知能の一種である [1] [2]. 生物は粒子にモデル化され, 粒子は潜在解に対応する. 対象とする問題の最適値を得るために, 粒子は自身の履歴を参照し, その情報を他の粒子と相互に通信することで客観的な評価を試みる. PSO は, その概念とアルゴリズムの簡潔さ, 動作の柔軟さ, などから様々なシステムの最適化に応用されている: 信号処理 [3]–[6], ニューラルネットワーク [7], 電力システム [8]–[10].

本章では, エノン写像 (Hénon map) の所望の周期点を探索する, 衝突粒子群最適化法 (collision particle swarm optimizer, CPSO) について考察する. 第一に, CPSO は格子点から成る離散的な探索空間で定義され, 決定論的な差分方程式により支配される [11]–[14]. このような決定論的なシステムは, 実験の再現性と回路実装の観点から有利である.

第二に, CPSO は粒子間に人工的な衝突を有しており, これは局所解, あるいは一部の最適解に粒子群が陥ることを避けるために効果的である. 通常の PSO では, これらを要因とする探索の停滞が解の発見を妨げる. 離散的な探索空間は, 衝突を定義する観点から便利である.

第三に, CPSO は Hénon map の周期点探索に適用される. この写像は様々な周期現象 / カオス現象を呈する非線形力学系の典型的な例である [15]. 所望の周期点を探索することは, 分岐現象解析の基礎である. 周期点の探索問題は, 離散的な複数解問題 (discrete multi-solution problems, DMSP) と等価である. これは, 複数の離散解領域

(discrete solution regions , DSR) を目的とする . DSR は連続探索空間における解 (analog multi-solution problems , AMSP) に対応する . DMSP は複数の適応度関数と論理演算からなる多目的問題 (multi-objective problems , MOP) で評価される . 基本的な数値実験を行うことで , 一部の DSR に粒子群が陥ることを避けるために人工的な衝突が効果的であり , CPSO は DMSP を解決可能であることが確認された .

本章が新規性と成り得る幾らかのポイントを提案する . Hénon map を含む力学系の周期点探索には種々の手法が存在する . しかしながら , その多く (例 : Newton-Raphson 法) は局所探索のために開発されている . これに対し , CPSO は大域的探索と局所探索の両方で利用可能であり , 目的関数に微分可能性を要求しないことも大きな特徴である . 従って , DMSP と MOP の実用的な例題として周期点探索問題は重要である .

CPSO は DMSP を対象とする手法であり , AMSP に従来の PSO を適用することとは全く異なる手法である .

AMSP を解決するために , 改良された PSO が提案されている : タブーサーチ法 , niching PSO , 成長木 PSO , etc. [16]–[22] . しかしながら , これらの既存のアルゴリズムは DMSP に適用することは困難である . DMSP のためのアルゴリズムを AMSP に適用することもまた同様である . ここで述べる基本的な結果は , DMSP のための効率的なアルゴリズムを開発するための基本的な情報として提供可能である . また , 群知能と非線形力学系の連携の糸口とも成り得る . 本章は , 我々の研究 [23] [24] において不充分であった CPSO と DMSP の定義を明確にし , 更に開発を進めた成果である .

## 4.2 離散的な複数解問題

### 4.2.1 離散時間力学系

CPSO の目的である DMSP を定義する . DMSP は様々な問題で定義することが可能であるが , ここでは離散時間力学系の解析として定義する ;

$$\mathbf{x}(n+1) = \mathbf{F}(\mathbf{x}(n)), \quad \mathbf{x} \equiv (x_1, x_2, \dots, x_D) \in \mathbb{R}^D$$



ただし,  $n$ : 離散時間,  $x$ :  $D$ 次元を示す状態変数. このシステムのために, 周期点を定義する. 点  $p$  は周期  $m$  の周期点である:

$$p = F^m(p) \text{ and } p \neq F^l(p) \text{ for } 0 \leq l < m$$

ただし,  $F^m$ :  $m$ 重の合成写像  $F$ . 周期 1 の周期点は不動点と呼ばれる.

所望の周期点を探索することは, ダイナミクスを解析するための基本的な問題である.

ここで, 探索問題に対する DMSP を定義する. まず, 基本関数  $G_m$  は正定である.

$$\begin{aligned} \text{基本関数 : } G_m(x) &\equiv \|F^m(x) - x\| \geq 0, x \in S_A, \\ S_A &\equiv \{x \mid x_j \in [X_{Lj}, X_{Rj}]\} \end{aligned} \quad (4.1)$$

ただし,  $j = 1, \dots, D$ ,  $S_A$ : アナログ探索空間,  $\|\cdot\|$ : ユークリッド距離.

$m = 1$  の場合, 不動点は関数の最小値で与えられる:

$$G_1(x) = 0, x \in S_A \quad (4.2)$$

つまり, 不動点はこの方程式の解により与えられる.

$m \geq 2$  を満たす素数の場合, 周期  $m$  の周期点は式 (4.3) の解で与えられる:

$$G_m(x) = 0, G_1 \neq 0, x \in S_A \quad (4.3)$$

$m \geq 2$  が素数で無い場合は, 素因数分解が可能である:  $m = m_1^{n_1} \times \dots \times m_K^{n_K}$ . ただし,  $m_k \geq 2$ :  $k$  番目の約数,  $k = 1, \dots, K$ .

周期  $m$  の周期点は式 (4.4) の解で与えられる:

$$G_m(x) = 0, G_{m/m_k}(x) \neq 0, x \in S_A \quad (4.4)$$

$x_s$  は式 (4.4) の解である. 一般に解の個数は少なくとも  $m$  あるいは  $m$  の倍数である.

式 (4.4) は複数次解  $x_{s1} \dots x_{sN_s}$  を持つ. ただし,  $N_s$  は解の個数.

例:

周期 12 の周期点の場合:  $12 = 2^2 \times 3$ ,  $m = 12$ ,  $m_1 = 2$ ,  $n_1 = 2$ ,  $m_2 = 3$ ,  $n_2 = 1$ .

この周期点は式 (4.5) の解である.

$$G_{12}(x) = 0, G_6(x) \neq 0, G_4(x) \neq 0, x \in S_A \quad (4.5)$$

アナログ探索空間  $S_A$  を  $M^D$  の格子点上に離散化する．このとき，式 (4.4) は DMSP に変換される：

$$G_m(\mathbf{x}) \leq T_m, G_{m/m_k}(\mathbf{x}) \geq T_{m/m_k}, \mathbf{x} \in S_D, \quad (4.6)$$

$$S_D = \{\mathbf{x} \mid x_j \in \{d_{0j}, \dots, d_{Mj}\}\}, d_{nj} = X_{Lj} + n(X_{Rj} - X_{Lj})/M$$

ただし， $n = 0, \dots, M$ ， $M$ ：次元あたりの格子点数， $T_m$ ：DMSP の近似解基準．互いに素な  $DSR_i \in S_D$  の集合を発見することは DMSP となる．これは式 (4.6) を満たす：

$$G_m(\mathbf{x}_i) \leq T_m, G_{m/m_k}(\mathbf{x}_i) \geq T_{m/m_k}, \mathbf{x}_i \in DSR_i$$

ただし， $i = 1, \dots, N_s$ ， $k = 1, \dots, K$ ．

## 4.2.2 Hénon map

本章では典型的な離散時間力学系の一種である Hénon map を考察する [15]．

$$\begin{cases} x_1(n+1) = F_1(x_1(n), x_2(n)) \equiv 1 - ax_1^2 + x_2 \\ x_2(n+1) = F_2(x_1(n), x_2(n)) \equiv bx_1 \end{cases} \quad (4.7)$$

パラメータ  $(a, b)$  を変化させることで，この2次元写像は様々な周期現象 / カオス現象を呈する [15]．基本関数  $G_m$  は次式で与えられる：

$$G_m(\mathbf{x}) = \sqrt{(F_1^m(x_1, x_2) - x_1)^2 + (F_2^m(x_1, x_2) - x_2)^2}$$

$$S_A = \{\mathbf{x} \mid x_1 \in [X_{L1}, X_{R1}], x_2 \in [X_{L2}, X_{R2}]\}$$

本章では基本的な問題を考え，周期4の周期点を探索する ( $m = 4$ )．

$$G_4(\mathbf{x}) = 0, G_2(\mathbf{x}) \neq 0, \mathbf{x} \in S_A \quad (4.8)$$

図 4.1 に  $G_2$  と  $G_4$  の等高線図および解を示す．周期点は式 (4.8) を満たす4つの解に対応する．離散化を行うことで，DMSP を得る．

$$G_4(\mathbf{x}_i) \leq T_4, G_2(\mathbf{x}_i) \geq T_2, \mathbf{x} \in S_D \quad (4.9)$$

$$S_D \equiv \{\mathbf{x} \mid x_1 \in \{d_{01}, \dots, d_{M1}\}, x_2 \in \{d_{02}, \dots, d_{M2}\}\}$$

ただし， $d_{nj} = X_{Lj} + n(X_{Rj} - X_{Lj})/M$ ， $j = 1, 2$ ．

DMSP は式 (4.9) を満たす，互いに素な部分集合の  $DSR_1, \dots, DSR_4$  を発見することを目的とする．この例題は第 4.4 節で説明する．

### 4.3 CPSO アルゴリズム

CPSO を定義するために、変数を以下に定義する。離散時間を  $t$  とする。これは探索時間である。探索時間  $t$  における粒子群は  $P^t \equiv (P_1^t, \dots, P_N^t)$  で示される。  $N$  は粒子数である。  $P_i^t \equiv (\mathbf{x}_i^t, \mathbf{v}_i^t)$  は  $i$  番目の粒子であり、位置と速度によって特徴付けられる。  $\mathbf{x}_i^t \equiv (x_{i1}^t, \dots, x_{iD}^t)$  はその粒子の位置を示し、  $\mathbf{v}_i^t \equiv (v_{i1}^t, \dots, v_{iD}^t)$  は速度を示す。ただし、  $i = 1, \dots, N$ 。粒子  $\mathbf{x}_i^t$  は潜在解である。粒子の位置と速度の更新には、パーソナルベスト ( $\mathbf{pbest}_i^t$ ) とローカルベスト ( $\mathbf{lbest}_i^t$ ) を用いる。パーソナルベストは各々の粒子がこれまでに発見した“ベスト”な位置である。“ベスト”な位置はCPSO アルゴリズムと適応度関数  $G_m$  により与えられる。ローカルベストは  $P_i^t$  の“近傍の粒子”のパーソナルベストにより与えられる。“近傍の粒子”は粒子群のトポロジーにより決定される [21]。便宜上、図 4.3(a) に示すリングトポロジーを用いる。粒子  $P_i$  の両隣の粒子が“近傍の粒子”である。これらは論理トポロジー (ネットワークの構造的な距離) であり、物理トポロジー (探索空間における粒子間の物理的な距離) ではない。

**Step 1 (初期化):** 探索時間および粒子群  $P^t$  を初期化する。  $t = 0$  に初期化、  $\mathbf{x}_i^0$  を探索空間  $S_D$  内にランダムに初期化し、  $\mathbf{v}_i^0$  を  $S_V = \{\mathbf{v} \mid -d_{Mj}/2 \leq v_j \leq d_{Mj}/2\}$  の範囲でランダムに初期化する。

**Step 2 (適合度判定):** 粒子が式 (4.10) を満たすとき、解候補として記録する。

$$G_m(\mathbf{x}_i^t) \leq T_m \text{ AND } G_{m/m_k}(\mathbf{x}_i^t) \geq T_{m/m_k}, k = 1, \dots, K. \quad (4.10)$$

**Step 3 ( $\mathbf{pbest}$  の更新):** パーソナルベストを更新する。  $\mathbf{pbest}_i^t$  に応じて、Case1 と Case2 のいずれかを適用する：

Case 1:  $G_m(\mathbf{pbest}_i^t) \leq T_m$  AND (  $G_{m/m_k}(\mathbf{pbest}_i^t) < T_{m/m_k}$  for some  $k$  )

$$\mathbf{pbest}_i^t = \begin{cases} \mathbf{x}_i^t & \text{if } G_{m/m_k}(\mathbf{x}_i^t) > G_{m/m_k}(\mathbf{pbest}_i^t) \\ \mathbf{pbest}_i^t & \text{otherwise} \end{cases} \quad (4.11)$$

Case 2:  $G_m(\mathbf{pbest}_i^t) > T_m$  OR (  $G_{m/m_k}(\mathbf{pbest}_i^t) \geq T_{m/m_k}$  for all  $k$  )

$$\mathbf{pbest}_i^t = \begin{cases} \mathbf{x}_i^t & \text{if } G_m(\mathbf{x}_i^t) < G_m(\mathbf{pbest}_i^t) \\ \mathbf{pbest}_i^t & \text{otherwise} \end{cases} \quad (4.12)$$

図 4.2(a) は  $G_m - G_{m/m_k}$  平面を示している．Case 1 は領域 I に対応し，Case 2 はそれ以外の領域 II  $\cup$  III  $\cup$  IV に対応する．Case 1 では  $G_m$  を減少， $G_{m/m_k}$  を増加させることを目的とし，Case 2 では  $G_m$ ， $G_{m/m_k}$  共に減少させることを目的としている．これは，複数の関数  $G_m$ ， $G_{m/m_k}$  から式 (4.10) に示す解候補を選出するためのひとつの方法である．第 4.4 節では，我々は基本となる， $m = 4$  と  $m/m_k = 2$  を用いる．

**Step 4 (*lbest* の更新):** ローカルベストを更新する．

Case 1:  $G_m(\mathbf{pbest}_i^t) \leq T_m$  AND ( $G_{m/m_k}(\mathbf{pbest}_i^t) < T_{m/m_k}$  for some  $k$ )

$$\mathbf{lbest}_i^t = \begin{cases} \mathbf{pbest}_c^t & \text{if } G_{m/m_k}(\mathbf{pbest}_c^t) \text{ is the maximum value in } N_i \\ & \text{and } G_{m/m_k}(\mathbf{pbest}_j^t) > G_{m/m_k}(\mathbf{lbest}_i^t) \\ \mathbf{lbest}_i^t & \text{otherwise} \end{cases} \quad (4.13)$$

ただし， $N_i \equiv \{i-1, i, i+1\}$  は  $i$  番目の粒子とその近傍である (図 4.3)．

Case 2:  $G_m(\mathbf{pbest}_i^t) > T_m$  OR ( $G_{m/m_k}(\mathbf{pbest}_i^t) \geq T_{m/m_k}$  for all  $k$ )

$$\mathbf{lbest}_i^t = \begin{cases} \mathbf{pbest}_c^t & \text{if } G_m(\mathbf{pbest}_c^t) \text{ is the minimum value in } N_i \\ & \text{and } G_m(\mathbf{pbest}_j^t) < G_m(\mathbf{lbest}_i^t) \\ \mathbf{lbest}_i^t & \text{otherwise} \end{cases} \quad (4.14)$$

**Step 5 (位置と速度の更新):** 粒子の速度と位置を更新する．

$$\mathbf{v}_i^t \leftarrow w\mathbf{v}_i^t + c(\mathbf{lbest}_i^t - \mathbf{x}_i^t) \quad (4.15)$$

$$\mathbf{x}_i^t \leftarrow \mathbf{x}_i^t + \mathbf{v}_i^t \quad (4.16)$$

ただし， $w$  と  $c$  は共に乱数を含まない，決定論的なパラメータである．従来の PSO は確率論的パラメータを含む [1]．式 (4.15)(4.16) を適用後，粒子の新しい位置  $\mathbf{x}_i^t$  は離散探索空間  $S_D$  内の格子点上に配置される．

**Step 6 (衝突判定):** 粒子の衝突を行う． $P_p^t$  の近傍領域に  $P_q^t$  が侵入した時，人工的な粒子衝突が適用される．

$$\begin{aligned} \mathbf{v}_p^t &\leftarrow ((\mathbf{v}_q^t - \mathbf{v}_p^t) \cdot \hat{\mathbf{c}})\hat{\mathbf{c}} + \mathbf{v}_p^t \\ \mathbf{v}_q^t &\leftarrow ((\mathbf{v}_p^t - \mathbf{v}_q^t) \cdot \hat{\mathbf{c}})\hat{\mathbf{c}} + \mathbf{v}_q^t, \quad \hat{\mathbf{c}} = \frac{\mathbf{x}_q^t - \mathbf{x}_p^t}{|\mathbf{x}_q^t - \mathbf{x}_p^t|} \end{aligned} \quad (4.17)$$

ただし，“ $\cdot$ ” は内積を示し， $\hat{\mathbf{c}}$  は  $P_p^t$  と  $P_q^t$  の位置から導出された単位ベクトルである．簡単のため，図 4.3(b) に示すように，各粒子に対しその粒子を取り巻く 8 つの格子点を

近傍領域として設定した．この速度を用いて，位置を更新する：

$$\mathbf{x}_p^t \leftarrow \mathbf{x}_p^t + \mathbf{v}_p^t, \mathbf{x}_q^t \leftarrow \mathbf{x}_q^t + \mathbf{v}_q^t \quad (4.18)$$

簡単のため，粒子の衝突は2つの粒子間でのみ発生するものとする．衝突の検出は不公平さを避けるために， $i$ 番目の粒子から実行する．ただし， $i \equiv t + N \pmod{N}$  とする．粒子が剛体である場合はより複雑な定義が必要となる [17]．探索空間  $S_D$  の境界に粒子  $P_i^t$  が到達したとき，粒子は境界面に反射する：

$$\begin{cases} v_{ij}^t \leftarrow v_{ij}^t - 2(x_{ij}^t - d_{0j}), & x_{ij}^t \leftarrow x_{ij}^t - 2(x_{ij}^t - d_{0j}) & \text{if } x_{ij} \text{ hits } X_{Lj} \\ v_{ij}^t \leftarrow v_{ij}^t - 2(x_{ij}^t - d_{Mj}), & x_{ij}^t \leftarrow x_{ij}^t - 2(x_{ij}^t - d_{Mj}) & \text{if } x_{ij} \text{ hits } X_{Rj} \end{cases} \quad (4.19)$$

人工的な衝突と反射の適用後，粒子は離散探索空間  $S_D$  内の格子点上に配置される．

**Step 7 (探索時間の更新):** 最大探索時間  $t = t_{max}$  を満たすとき探索終了とする．そうでなければ， $t \rightarrow t + 1$  に更新し，**Step 2** に戻る．

$m = 4$ ， $m/m_k = 2$  とする CPSO の疑似コードを **Algorithm 1** に示す． $m$  と  $m/m_k$  を一般化した疑似コードは付録 A.3 を参照のこと．このアルゴリズム終了後，**Step 2** から解候補を取得する．これらの解候補から目的の DSR を自由に構築することが可能となれば，DMSP は解決される．

## 4.4 数値実験

CPSO を DSM に適用するために，式 (4.9) を定義した：4つの DSR は Hénon map の4周期点に対応する．全てのパラメータ  $(a, b)$ (式 (4.7)) を調査することは困難であるため，試行錯誤の結果，以下のパラメータを選択した：

$$N = 30, t_{max} = 50, M = 128, T_4 = T_2 = 0.05, w = 0.7, c = 1.4. \quad (4.20)$$

この場合，総当たり計算 (brute force) では  $M^D = 128^2 = 2^{14}$  回の反復回数を必要とするが，CPSO では最多で  $N \times t_{max} = 1500$  回で実行されることに注意したい． $2^{14}$  回の回数があれば全ての DSR を発見することは可能であるが，全 DSR を発見することは本

質ではない．CPSOの基本的な能力を調査し，Hénon mapを含む種々のシステムへの効率的な探索アルゴリズム開発のための基本的な情報を提供することが目的である．格子点数  $M$ ，次元数  $D$ が増加するにつれて総当たり計算での計算は次第に不可能になっていく．

図 4.4 は探索過程の成功例の概要である．解候補が良好に発見されたことが確認された． $t = t_{max}$  では，4つの解候補の集合 ( $DSR_1, \dots, DSR_4$ ) に分類する (図 4.5 (f))． $DSR_1, \dots, DSR_4$  はそれぞれが Hénon map の 4 周期点を含む．DSR の自動的な分類分けの方法として，動径基底適応共鳴理論写像 (Radial Basis Adaptive Resonance Theory Map, RBART-Map) [25] などのいくつかの分類アルゴリズムを適用することが考えられる．しかしながら，このような問題を体系的に議論することは容易ではなく，今後の課題である．

図 4.5 はパーソナルベストを  $G_2 - G_4$  平面に図示したものである．図 4.4 と同様の成功例を用いている．4つの DSR に分類し，色で塗り分けた (図 4.5 (f))． $DSR_1$  (茶色)， $DSR_3$  (青色)， $DSR_4$  (マゼンタ) は  $t = 12$  までに発見され， $DSR_2$  (緑色) は  $t = 19$  までに発見されている．

図 4.6 に，各 DSR に収束した粒子のパーソナルベストの時間発展を示す．図中の \* 印はパーソナルベストが増加しており，これは  $G_4(\mathit{pbest}) \leq T_4$  and  $G_2(\mathit{pbest}) < T_2$  (領域 I) を満たす領域にパーソナルベストが記録されたことを示唆している．このような増加は DMSP を解析する上で有効であり得る．これらの結果は，粒子の位置とパーソナルベストは特定の DSR の近くに集中し，粒子群が  $t = t_{max}$  までに十分な多様性を発揮出来ないことを示唆している．通常の PSO では，一部の DSR に陥った粒子がそこから脱出することは困難な傾向にある [24]．

$N \in \{10, 20, 30, 40, 50, 60, 70\}$  として，初期値を変えてそれぞれ 100 回の数値実験を行った．粒子数以外のパラメータは式 (4.20) と同様である．これらの結果を表 4.1 から表 4.3 に示す．表では以下の 3 種類の特徴量を用いる．

SR: 平均成功率 . 各 DSR で少なくともひとつの解候補を発見した時 , 探索成功とする .

#ITE: 探索成功時の平均探索時間 .

#COL: 探索成功時の 1 粒子あたりの平均衝突回数 .

衝突の影響を調べるためにローカルベスト型 PSO(local-best PSO , LPSO) の結果を示す . LPSO は CPSO アルゴリズムにおける Step 6 の人工的な衝突を削除することによって得られる . 粒子数  $N$  が増加するに連れて SR の増加と #ITE の減少が見られる . CPSO はより良いパフォーマンスを実現している : (LPSO と比較して)SR はより多く , #ITE はより少ない . これは粒子の衝突がパフォーマンスの向上に寄与していることを示している . 表 4.3 では  $N$  が増加するに連れて #COL が減少する .  $N$  の増加につれて , CPSO の探索速度が向上する (#ITE が少なくなる) . 従って探索成功につれて衝突の累積数は減少する .

## 4.5 まとめ

CPSO を提案し , Hénon map の所望の周期点探索問題に適用した . 探索問題は DMSP と MOP と捉えることが可能である . 基本的な数値実験を行い , 粒子の衝突が全ての DMSP の発見へ効果的に寄与することを確認した . 今後の課題として , 検索処理の詳細な分析 , パラメータ値の適当な設定と離散力学系の分岐解析への応用が挙げられる .

---

**Algorithm 1** The pseudo code of the generalized CPSO algorithm ( $m = 4$ ,  $m/m_k = 2$ ).

---

Create and initialize a particle swarm  $P^t$

**while** ( $t < t_{max}$ ) **do**

**for each particle**  $i = 1, \dots, N$  **do**

    Evaluate fitness function  $G_4(x_i^t)$

**end for**

  // set the personal best position  $pbest_i^t$

**for each particle**  $i = 1, \dots, N$  **do**

**if**  $G_4(pbest_i^t) \leq T_4$  AND  $G_2(pbest_i^t) < T_2$  ( $k = 1, \dots, K$ ) **then**

      update personal best using Eq. (4.11)

**else**

      update personal best using Eq. (4.12)

**end if**

**end for**

  // set the neighborhood best position  $lbest_i^t$

**for each particle**  $i = 1, \dots, N$  **do**

**if**  $G_4(pbest_i^t) \leq T_4$  AND  $G_2(pbest_i^t) < T_2$  ( $k = 1, \dots, K$ ) **then**

      update local best using Eq. (4.13)

**else**

      update local best using Eq. (4.14)

**end if**

**end for**

**for each particle**  $i = 1, \dots, N$  **do**

    update the velocity using Eq. (4.15)

    update the position using Eq. (4.16)

**end for**

  // collision detection

**for each particle**  $i = t, \dots, t + N \bmod N$  **do**

    update the velocity using Eq. (4.17)

    update the position using Eq. (4.18)

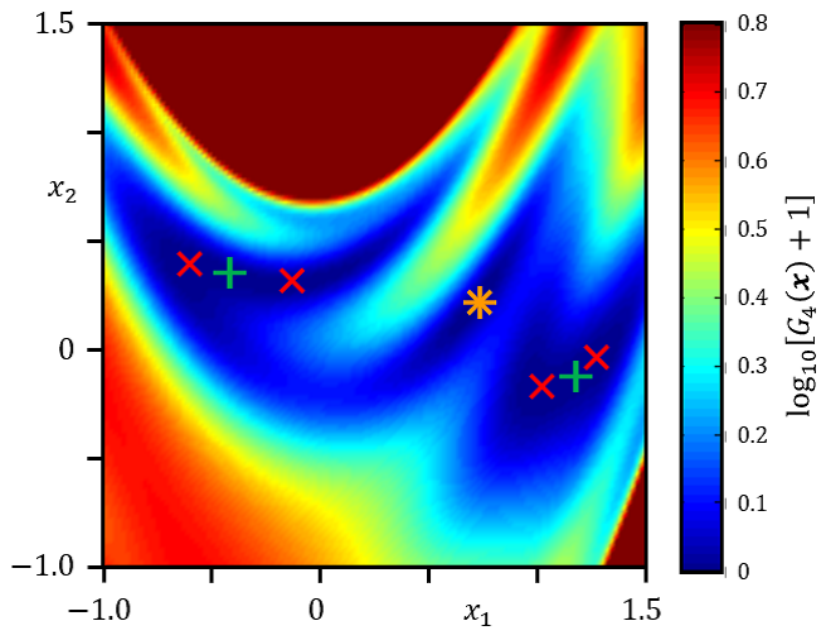
**end for**

  update the time step  $t \rightarrow t + 1$

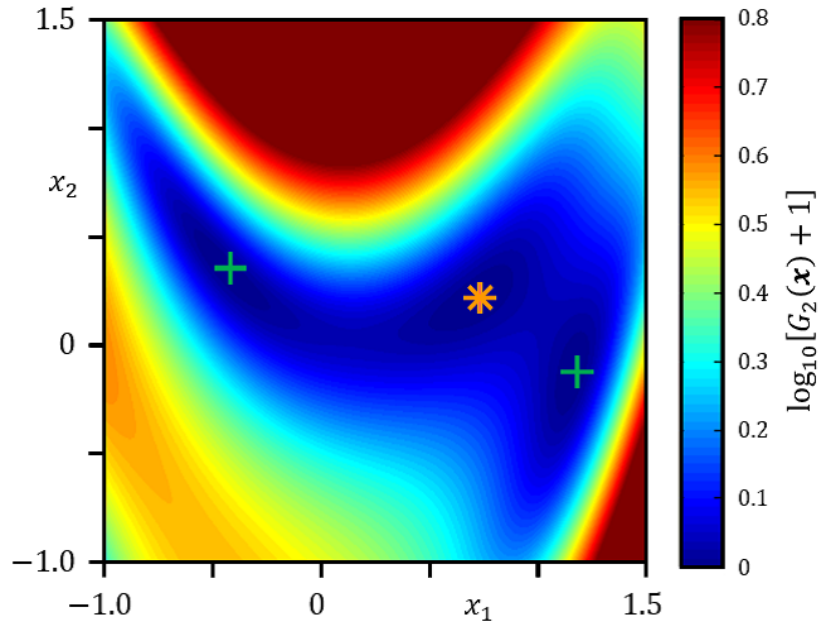
**end while**

---



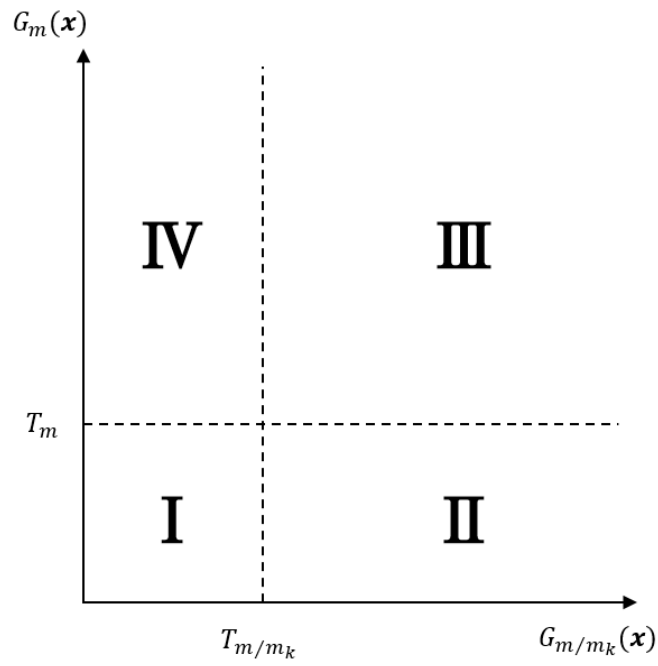


(a) Contour map of  $G_4$ . Red x-mark, green +-mark and orange \*-mark denote periodic point with period 4, period 2 and period 1, respectively.

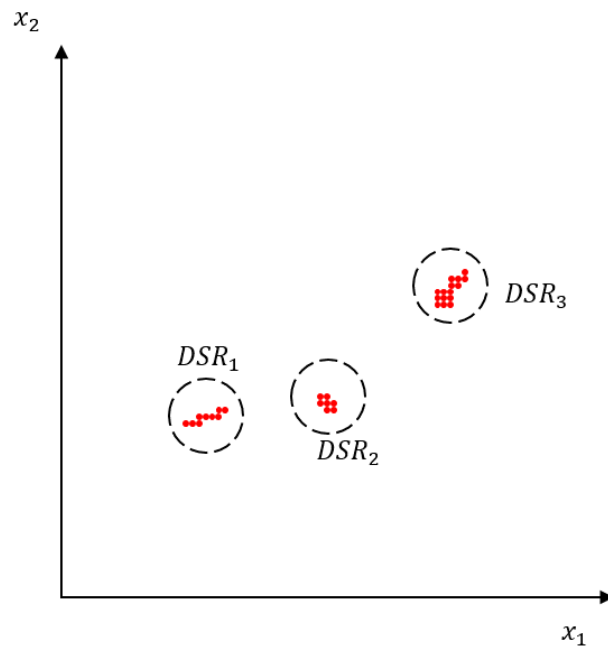


(b) Contour map of  $G_2$ . Green +-mark and orange \*-mark denote periodic point with period 2 and period 1, respectively.

⊠ 4.1: Basic functions of the Hénon map for  $a = 0.96875$ ,  $b = 0.3$ .

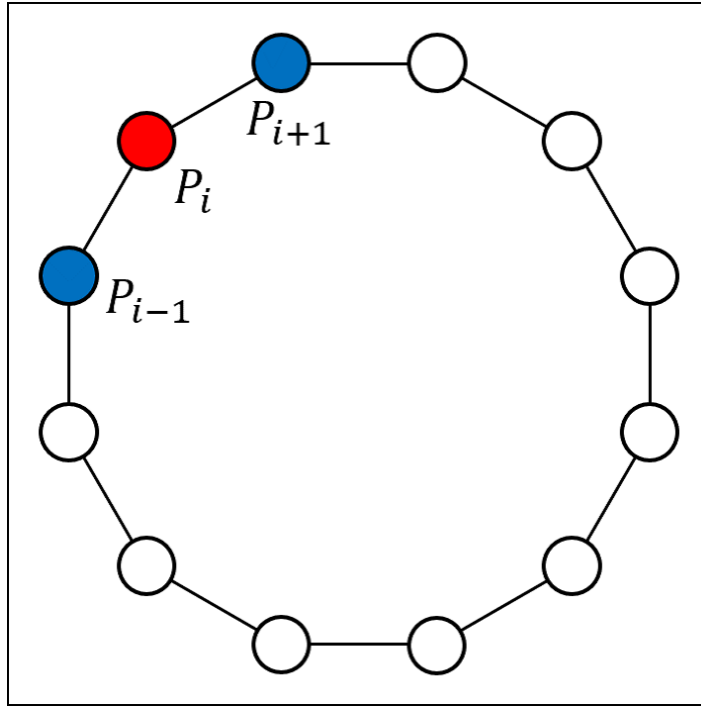


(a) Evaluation by two fitness functions.

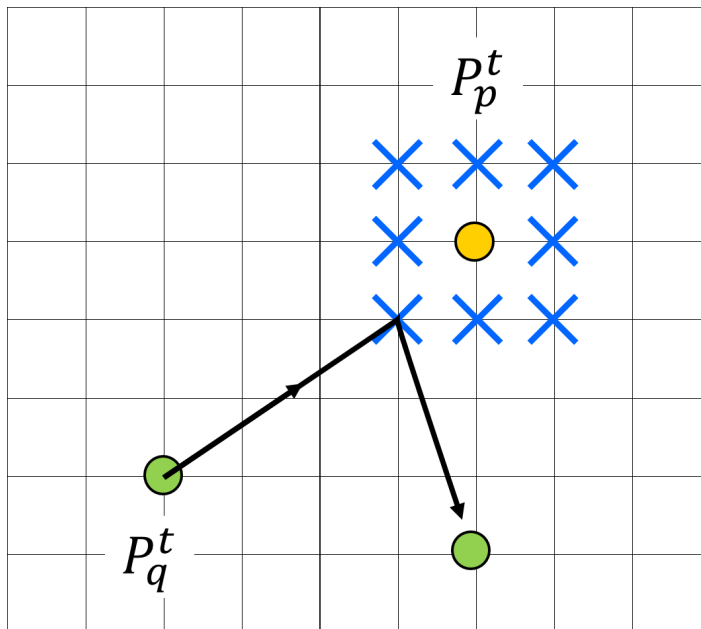


(b) Discrete Solution Regions.

☒ 4.2: Basic concept in the CPSO algorithm.



(a) Illustration of ring topology. Both sides particles ( $P_{i-1}$  and  $P_{i+1}$ ) of the particle  $P_i$  (red) are the neighbors of  $P_i$ .



(b) Illustration of artificial collision. The yellow circle denotes position of a particle  $P_p^t$  and the blue x-mark construct vicinity area of particle  $P_p^t$ . The green circle is particle  $P_q^t$ , moves on a vicinity area of particle  $P_p^t$  and judged to collision with  $P_p^t$ .

☒ 4.3: Neighbor and vicinity.

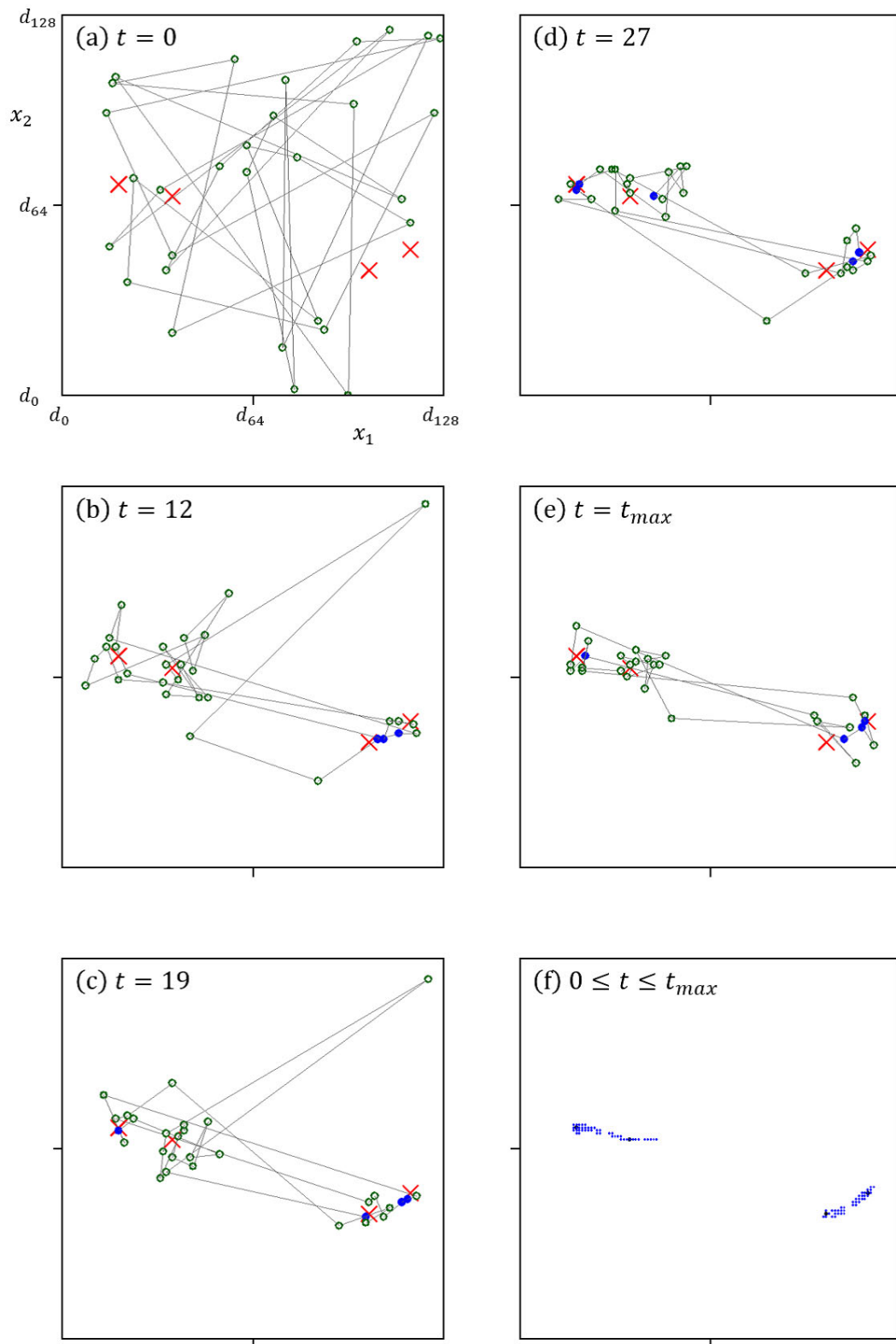
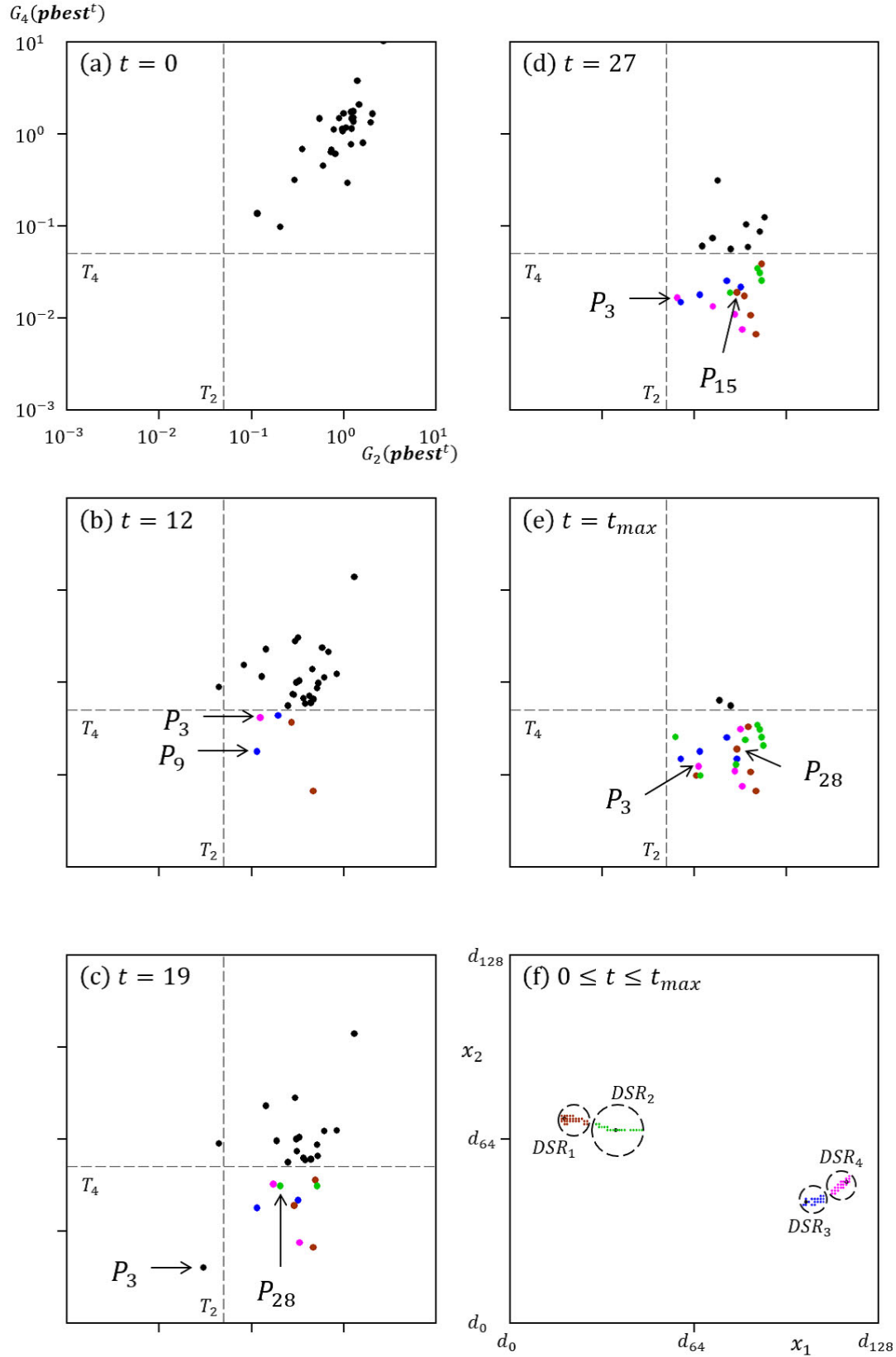


图 4.4: Exploring process on the search space.

(a) to (e): Snapshots. Blue circuits denote particles of the solution candidate and green circuits denote other particles. (f) Blue dots denote solution candidates (overlapped drawing).



⊠ 4.5: Personal best on the fitness functions plane in the exploring process.

(a) to (e): snapshots at the same timing as Fig. 4.4. Colored particles denote solution candidates in each DSRs. The indicated candidates,  $P_3, P_9, P_{15}$  and  $P_{28}$ , represent the four DSRs. (f) The color classification of DSRs.

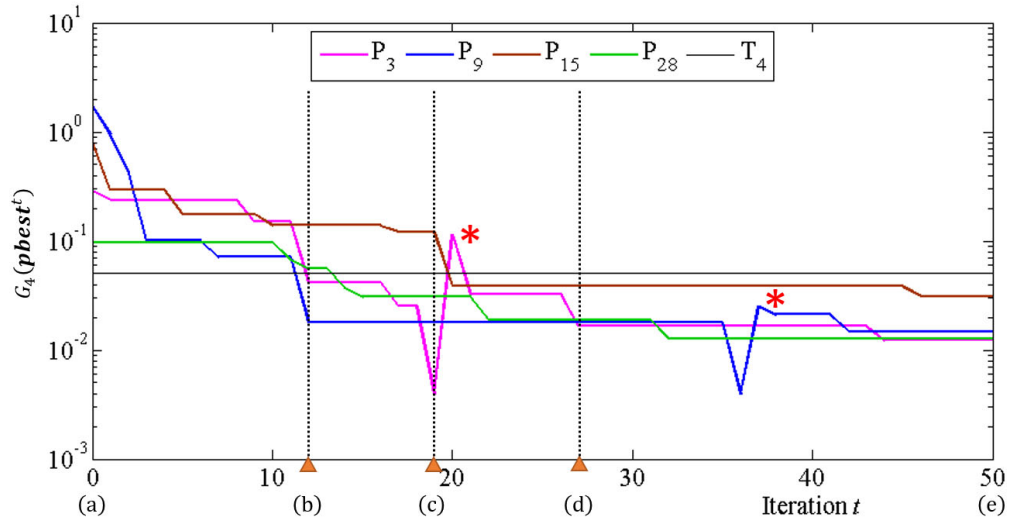


图 4.6: Time evolution of the pbests of solution candidates  $P_3, P_9, P_{15}$  and  $P_{28}$ . The four candidates represent the four DSRs and are indicated in Fig.4.5.

表 4.1: The average success rate (SR) in 100 trials.

$N$	10	20	30	40	50	60	70
LPSO	26.0%	62.0%	80.0%	87.0%	92.0%	95.0%	98.0%
CPSO	33.0%	71.0%	86.0%	94.0%	98.0%	98.0%	100%

表 4.2: The average number of iterations (#ITE) in successful runs.

$N$	10	20	30	40	50	60	70
LPSO	18.8	12.0	10.6	7.6	7.0	6.5	6.0
CPSO	18.3	13.9	10.6	8.4	8.3	6.3	5.4

表 4.3: The average number of collisions (#COL) in successful runs.

$N$	10	20	30	40	50	60	70
LPSO	N/A	N/A	N/A	N/A	N/A	N/A	N/A
CPSO	3.02	2.14	1.57	1.08	1.61	0.80	0.60

## 参考文献

- [1] J. Kennedy and R. Eberhart, Particle Swarm Optimization, Proc of IEEE/ICNN, pp.1942–1948, 1995.
- [2] A. P. Engelbrecht, Fundamentals of computational swarm intelligence, Willey, 2005.
- [3] M. P. Wachowiak, R. Smolikova, Y. Zheng, J. M. Zurada and, A. S. Elmaghraby, An approach to multimodal biomedical image registration utilizing particle swarm optimization, IEEE Trans. Evol. Comput., 8, 3. pp. 289–301, 2004.
- [4] S.-T. Hsieh, T.-Y. Sun, C.-L. Lin, and C.-C. Liu, Effective learning rate adjustment of blind source separation based on an improved particle swarm optimizer, IEEE Trans. Evol. Comput., 12, 2. pp. 242–251, 2008.
- [5] A. Ghosh, A. Ghosh, A. Chowdhury, A. Konar, E. Kim and A. K. Nagar, Linear phase low pass FIR filter design using genetic particle swarm optimization with dynamically varying neighborhood technique, in Proc. IEEE Congress Evol. Comput., pp. 529–535, 2012.
- [6] R. A. Vural, T. Yildirim, T. Kadioglu and A. Basargan, Performance evaluation of evolutionary algorithms for optimal filter design, IEEE Trans. Evol. Comput., 16, 1. pp. 135–147, 2012.
- [7] A. B. van Wyk and A. P. Engelbrecht, Overfitting by PSO trained feedforward neural networks, in Proc. IEEE Congress Evol. Comput., pp. 2672 – 2679, 2010.

- [8] H. Qin, J. W. Kimball and G. K. Venayagamoorthy, Particle swarm optimization of high-frequency transformer, in Proc. Annual Conf. IEEE Ind. Electron. Soc., pp. 2908–2913, 2010.
- [9] K. Kawamura and T. Saito, Design of switching circuits based on particle swarm optimizer and Hybrid Fitness Function, in Proc. Annu. Conf. IEEE Ind. Electron. Soc., pp. 1099–1103, 2010.
- [10] H. Matsushita and T. Saito, Application of particle swarm optimization to parameter search in dynamical systems, NOLTA, IEICE, E94-N, 10, pp. 458–471, 2011.
- [11] Z. Sevkli and F. E. Sevilgen, Discrete particle swarm optimization for the orienteering problem, in Proc. IEEE Congress Evol. Comput., pp. 1937–1944, 2010.
- [12] M. Kubota and T. Saito, A discrete particle swarm optimizer for multi-solution problems, IEICE Trans. Fundamentals, E95-A, 1, pp. 406–409, 2012.
- [13] K.-B. Lee and J.-H. Kim, Mass-spring-damper motion dynamics-based particle swarm optimization, in Proc. IEEE Congress Evol. Comput., pp. 2348–2353, 2008.
- [14] K. Jin’no and T. Shindo, Analysis of dynamical characteristic of canonical deterministic PSO, in Proc. IEEE Congress Evol. Comput., pp. 1105–1110, 2010.
- [15] E. Ott, Chaos in Dynamical Systems, Cambridge Univ. Press, 1993.
- [16] K. E. Parsopoulos and M. N. Vrahatis, On the computation of all global minimizers through particle swarm optimization, IEEE Trans. Evol. Comput., 8, 3, pp. 211–224, 2004.
- [17] T. Hendtlass, WoSP: a multi-optima particle swarm algorithm, Proc. IEEE/CEC, pp. 727–734, 2005.



- [18] D. Parrott and X. Li, Locating and tracking multiple dynamic optima by a particle swarm model using speciation, *IEEE Trans. Evol. Comput.*, 10, 4. pp. 440–458, 2006
- [19] S. Yang and C. Li, A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments, *IEEE Trans. Evol. Comput.*, 14, 6. pp. 959–974, 2010.
- [20] X. Li, Niching without niching parameters: particle swarm optimization using a ring topology, *IEEE Trans. Evol. Comput.*, 14, 1. pp. 150–169, 2010.
- [21] T. Saito and E. Miyagawa, Growing-tree particle swarm optimizer with simple tabu search function, *Proc. of NOLTA*, pp. 376–379, 2009.
- [22] R. Sano, T. Shindo, K. Jin’no, and T. Saito, PSO-based multiple optima search systems with switched topology, in *Proc. IEEE Congress Evol. Comput.*, pp. 3301–3307, 2012.
- [23] K. Maruyama, R. Sano and T. Saito, Deterministic discrete particle swarm optimizers with collision and insensitivity, *Proc. NDES*, pp. 245–248, 2012.
- [24] K. Maruyama and T. Saito, Deterministic Particle Swarm Optimizers with Collision for Discrete Multi-solution Problems, *Proc. IEEE/SMC*, pp. 1335–1340, 2013.
- [25] M. Takanashi, H. Torikai and T. Saito, An approach to collaboration of growing self-organizing maps and adaptive resonance theory maps, *IEICE Trans. Fundamentals*, E90-A, 9, pp. 2047–2050, 2007.

## 第5章 むすび

本論文では複数最適解問題を対象とした新しい粒子群最適化法を考察した。探索空間は離散化され、粒子の動きを支配する更新式は決定論的である。粒子群はシンプルなリングトポロジーネットワークで構成されており、粒子間の相互作用には適度な鈍感さを、粒子間の物理的な距離には人工的な粒子間の衝突を与えた。これにより論理トポロジー・物理トポロジーの両者から粒子群に多様性を与え、探索性能の向上へ寄与した。

解析対象には二次元離散力学系の周期点探索を例題とし、Ikeda map と Hénon map の周期点探索を適用した。特に Hénon map では所望の周期点探索問題として捉え、複数解問題と多目的問題の融合として扱った。この試みは粒子群最適化法に基づく非線形力学系の新しい解析手法の第一歩である。

今後の課題として、以下に示すような課題が挙げられる：

1. 探索過程の詳細な分析
2. パラメータ値の適当な設定
3. 粒子間の衝突の様々な応用
4. 衝突の効果の評価
5. 離散力学系の分岐解析への応用

# 付録A 擬似コード

## A.1 DDPSO 擬似コード (第2章)

---

**Algorithm 2** The pseudo code of the DDPSO algorithm (chapter 2).

---

```
Create and initialize a particle swarm  $P^t$ 
while ( $t < t_{max}$ ) do
  for each particle  $i = 1, \dots, N$  do
    Evaluate fitness function  $F(x_i^t)$ 
  end for
  // set the personal best position  $pbest_i^t$ 
  for each particle  $i = 1, \dots, N$  do
    if  $F(x_i^t) < F(pbest_i^t)$  then
      update personal best using Eq. (2.4)
    end if
  end for
  // set the neighborhood best position  $lbest_i^t$ 
  for each particle  $i = 1, \dots, N$  do
    if  $F(pbest_j^t) < F(lbest_i^t) - \epsilon$  then
      update local best using Eq. (2.5)
    end if
  end for
  for each particle  $i = 1, \dots, N$  do
    update the velocity and position using Eq. (2.6)
  end for
  // inter-particle collision
  for each particle  $i = 1, \dots, N$  do
    if  $F(lbest_m^t) > F(lbest_n^t)$  then
      update the velocity and position using Eq. (2.7)
    else
      update the velocity and position using Eq. (2.8)
    end if
  end for
  update the time step  $t \rightarrow t + 1$ 
end while
```

---

## A.2 CPSO 擬似コード (第3章)

---

**Algorithm 3** The pseudo code of the CPSO algorithm (chapter 3).

---

```
Create and initialize a particle swarm  $P^t$ 
while ( $t < t_{max}$ ) do
  for each particle  $i = 1, \dots, N$  do
    Evaluate fitness function  $F_D(\mathbf{x}_i^t)$ 
  end for
  // set the personal best position  $\mathbf{pbest}_i^t$ 
  for each particle  $i = 1, \dots, N$  do
    if  $F_D(\mathbf{x}_i^t) < F_D(\mathbf{pbest}_i^t)$  then
      update personal best using Eq. (3.6)
    end if
  end for
  // set the neighborhood best position  $\mathbf{lbest}_i^t$ 
  for each particle  $i = 1, \dots, N$  do
    if  $F_D(\mathbf{pbest}_j^t) < F_D(\mathbf{lbest}_i^t)$  then
      update local best using Eq. (3.7)
    end if
  end for
  for each particle  $i = 1, \dots, N$  do
    update the velocity and position using Eq. (3.8)
  end for
  // collision detection
  for each particle  $i = 1, \dots, N$  do
    update the velocity using Eq. (3.10)
    update the position using Eq. (3.11)
  end for
  update the time step  $t \rightarrow t + 1$ 
end while
```

---

### A.3 CPSO 擬似コード (第4章)

---

**Algorithm 4** The pseudo code of the generalized CPSO algorithm (chapter 4).

---

```
Create and initialize a particle swarm  $P^t$ 
while ( $t < t_{max}$ ) do
  for each particle  $i = 1, \dots, N$  do
    Evaluate fitness function  $G_m(\mathbf{x}_i^t)$ 
  end for
  // set the personal best position  $\mathbf{pbest}_i^t$ 
  for each particle  $i = 1, \dots, N$  do
    if  $G_m(\mathbf{pbest}_i^t) \leq T_m$  AND  $G_{m/m_k}(\mathbf{pbest}_i^t) < T_{m/m_k}$  ( $k = 1, \dots, K$ ) then
      update personal best using Eq. (4.11)
    else
      update personal best using Eq. (4.12)
    end if
  end for
  // set the neighborhood best position  $\mathbf{lbest}_i^t$ 
  for each particle  $i = 1, \dots, N$  do
    if  $G_m(\mathbf{pbest}_i^t) \leq T_m$  AND  $G_{m/m_k}(\mathbf{pbest}_i^t) < T_{m/m_k}$  ( $k = 1, \dots, K$ ) then
      update local best using Eq. (4.13)
    else
      update local best using Eq. (4.14)
    end if
  end for
  for each particle  $i = 1, \dots, N$  do
    update the velocity using Eq. (4.15)
    update the position using Eq. (4.16)
  end for
  // collision detection
  for each particle  $i = t, \dots, t + N \bmod N$  do
    update the velocity using Eq. (4.17)
    update the position using Eq. (4.18)
  end for
  update the time step  $t \rightarrow t + 1$ 
end while
```

---

## 付録B Ikeda mapの概形と DDPSO探索過程の要約

第2章の追加データを示す。

図B.1にIkeda mapの等高線図および解を示す。

図B.2, B.3は、いずれも粒子数  $N = 50$  , 探索時間  $0 \leq t \leq t_{max}$  とした時の粒子の動きを一枚に要約したものである。図B.2: 衝突を除去した場合 ( $k = 0$ ) , 粒子群の多様性はわずかに減少し、全ての近似解の発見を困難にする (4つの近似解を発見)。図B.3: 衝突と鈍感さの両者を除去 ( $k = 0, \epsilon = 0$ ) した場合、アルゴリズムの性能は更に低下する (2つの近似解を発見)。

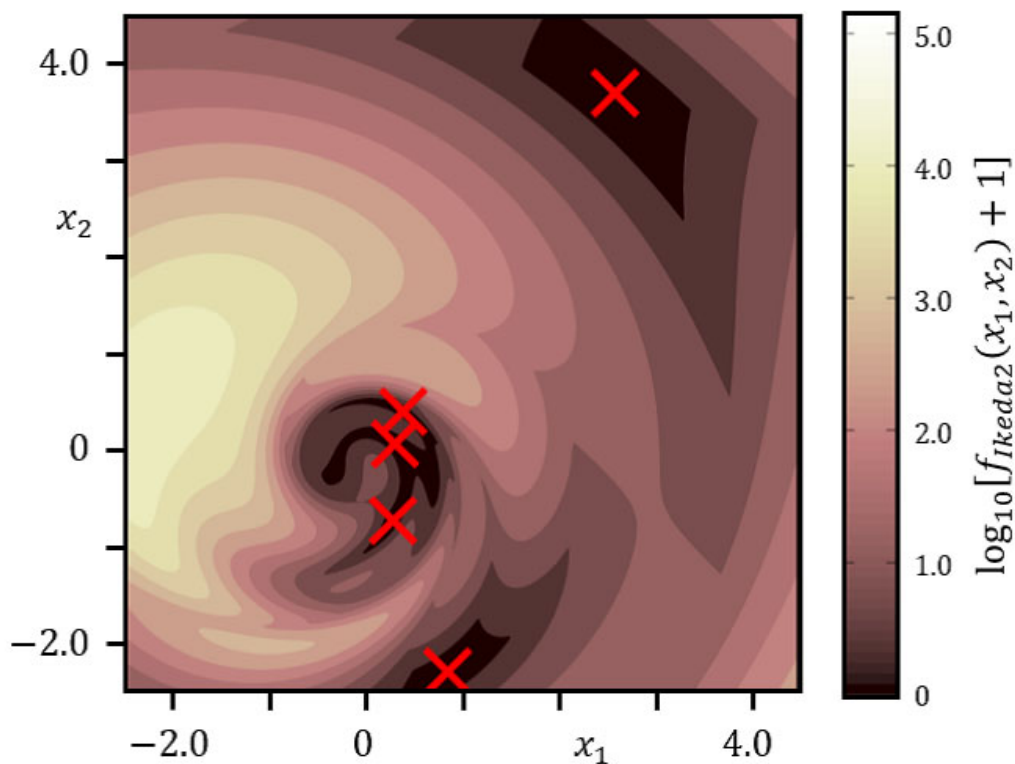
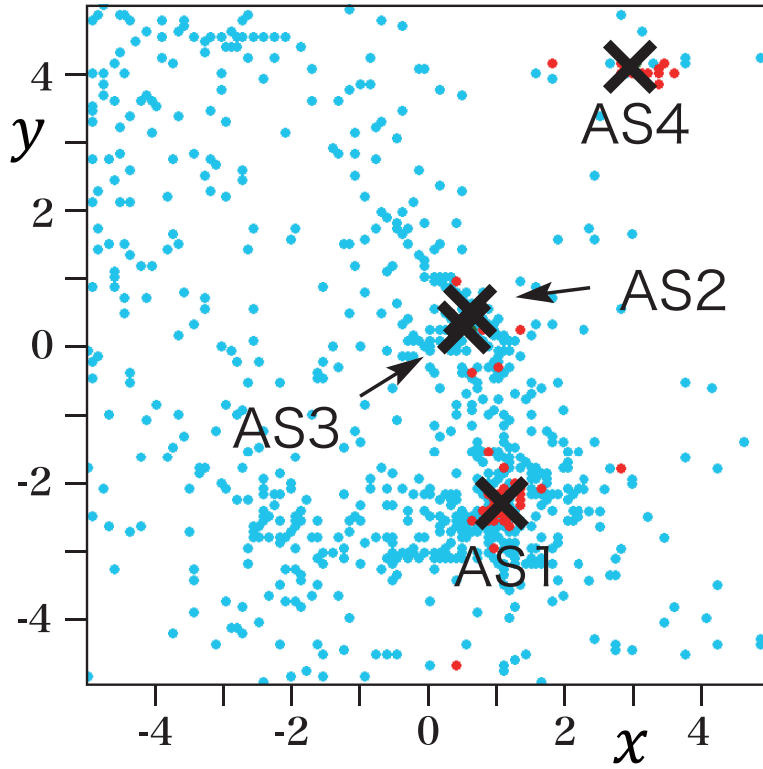


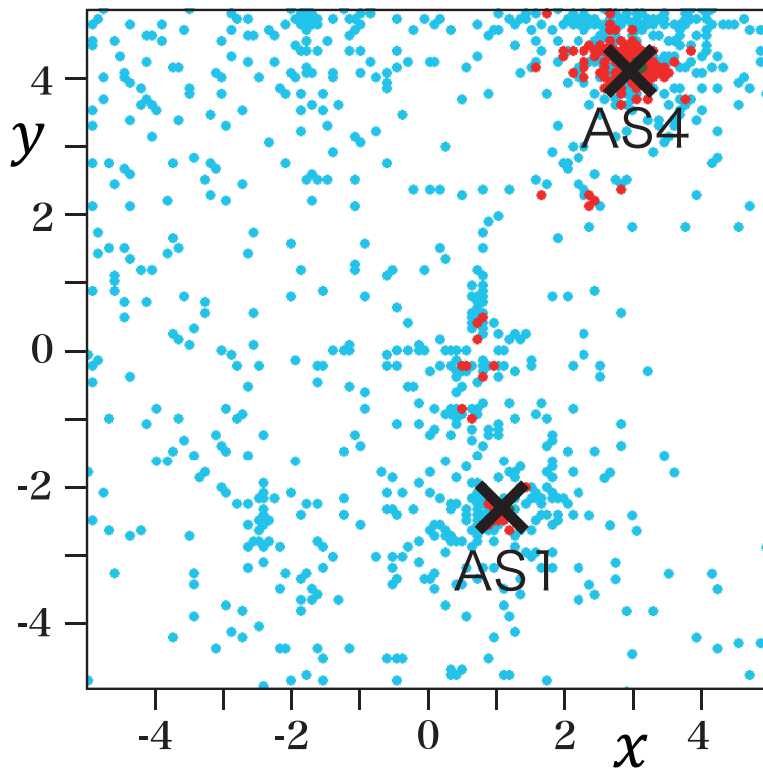
図 B.1: An objective function of Ikeda map for  $u = 0.9$ .

Contour map of  $f_{Ikeda2}$ . Red x-mark denote periodic point with period 2 and period 1.



⊠ B.2: Search for  $C_1 = 0.25$ ,  $k = 0$ ,  $\epsilon = 0.2$ ,  $t_{max} = 50$ .

Positions of  $N = 50$  particles are overwritten for  $0 \leq t \leq t_{max}$  4 approximate solutions are found.



⊠ B.3: Search for  $C_1 = 0.25$ ,  $k = 0$ ,  $\epsilon = 0$ ,  $t_{max} = 50$ .

Positions of  $N = 50$  particles are overwritten for  $0 \leq t \leq t_{max}$  2 approximate solutions are found.

## 研究業績

### (原著論文)

1. K. Maruyama and T. Saito, “Collision Particle Swarm Optimizers and Exploring Periodic Points,” NOLTA, IEICE, submitted.

### (国際会議)

1. T. Sato, K. Maruyama and T. Saito, “A Fission-and-Recombination Particle Swarm Optimizer and its Application to Multi-solution Problems,” Proc. of 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2014), San Diego, USA, in preparation.
2. T. Takemura, K. Maruyama and T. Saito, “A PSO based Periodic Points Explorer,” Proc. of 2014 International Symposium on Nonlinear Theory and its Applications (NOLTA2014), Luzern, Switzerland.
3. K. Maruyama and T. Saito, “A Collision PSO for Search of Periodic Points,” Proc. of Nonlinear Dynamics of Electronic Systems 2014 (NDES 2014), Albena, Bulgaria, submitted.
4. K. Maruyama and T. Saito, “Deterministic Particle Swarm Optimizers with Collision for Discrete Multi-solution Problems,” Proc. of 2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2013), Manchester, UK, pp. 1335–1340.
5. T. Sato, K. Maruyama and T. Saito, “A Fission-and-Recombination Particle Swarm Optimizer,” Proc. of 2013 International Symposium on Nonlinear Theory and its Applications (NOLTA2013), Santa Fe, USA, pp. 193–196.
6. K. Maruyama, R. Sano and T. Saito, “Deterministic Discrete Particle Swarm Optimizers with Collision and Insensitivity,” Proc. of Nonlinear Dynamics of Electronic Systems 2012 (NDES 2012), Wolfenbüttel, Germany, pp. 245–248.



## (国内発表)

1. 竹村卓也, 丸山一紀, 斎藤利通, “粒子群最適化法に基づく分岐解析へのアプローチ,” 電子情報通信学会 総合大会, 新潟, 2014年3月.
2. 佐藤拓海, 丸山一紀, 斎藤利通, “分裂再結合粒子群最適化法に基づく周期点探索,” 進化計算学会 第6回進化計算学会研究会, 調布, 2014年3月.
3. 安藤直人, 丸山一紀, 斎藤利通, “衝突を伴う粒子群最適化法と最大電力点探索への応用,” 進化計算学会 第6回進化計算学会研究会, 調布, 2014年3月.
4. 佐藤拓海, 丸山一紀, 斎藤利通, “分裂再結合粒子群最適化法と複数解問題,” 進化計算学会 進化計算シンポジウム 2013, 霧島, 2013年12月, 1-12.
5. 安藤直人, 丸山一紀, 斎藤利通, “衝突を伴う決定論的粒子群最適化法と最大電力点探索への応用,” 進化計算学会 進化計算シンポジウム 2013, 霧島, 2013年12月, 3-12.
6. 丸山一紀, 斎藤利通, “衝突を伴う確定的粒子群最適化法,” 電子情報通信学会情報処理学会 第12回情報科学技術フォーラム (FIT2013), 鳥取, 2013年9月, F-029.
7. 丸山一紀, 斎藤利通, “時間変化を伴う鈍感な決定論的粒子群最適化法,” 第26回回路とシステムワークショップ, 淡路, 2013年7月, pp. 484-488.
8. 佐藤拓海, 丸山一紀, 斎藤利通, “局所成長機構を有する粒子群最適化法,” 第26回回路とシステムワークショップ, 淡路, 2013年7月, pp. 60-63.
9. 丸山一紀, 斎藤利通, “鈍感粒子群最適化法とその周期点探索への応用,” 電子情報通信学会 技術研究報告 (NLP 研究会), 札幌, 2013年1月, NLP2012-119.
10. 丸山一紀, 斎藤利通, “複数解探索のための粒子群最適化アルゴリズムにおける鈍感さと衝突の効果,” 電子情報通信学会 技術研究報告 (NC 研究会), 北九州, 2012年10月, NC2012-52.

11. 丸山一紀, 佐野亮介, 斎藤利通, “鈍感な決定論的離散粒子群最適化法,” 第 25 回 回路とシステムワークショップ, 淡路, 2012 年 7 月, pp. 267–269.
12. 丸山一紀, 佐野亮介, 斎藤利通, “鈍感な粒子群最適化と複数解問題,” 電子情報通信学会 技術研究報告 (NLP 研究会), 会津若松, 2012 年 1 月, NLP2011–132.
13. 丸山一紀, 佐野亮介, 斎藤利通, “鈍感な粒子群最適化について,” 電子情報通信学会 ソサイエティ大会, 札幌, 2011 年 9 月, A-2–15.

## 謝辞

本修士論文は著者が法政大学大学院工学研究科電気工学専攻修士課程において非線形回路工学研究室にて行った研究をまとめたものである。この研究は同大学理工学部電気電子工学科 斎藤利通教授の指導下で行ったもので、全ての研究活動を遂行するにあたり同氏から大変御参考になる御指導・御鞭撻を沢山賜りました。ここに心から深く感謝致します。

同じ分野の研究者として、日頃から活発な議論にお付き合い頂いた斎藤利通研究室の佐野亮介氏、久保田将史氏、佐藤拓海君、安藤直人君、ART-map により PSO の可能性を拡げてくれた徳永章哲君に深謝致します。

法政大学理工学部電気電子工学科 斎藤利通研究室の皆様にはいろいろな有益な御討論・ご助言を戴きました。ここに感謝の意を表します。

最後に徹夜続きの研究生活の中で、深夜に帰宅する私を励まし続けてくれた両親に深く感謝します。