

# 捕食者を加えた粒子群最適化での捕食範囲の調査

犬飼 規雄<sup>†</sup> 井上 拓也<sup>†</sup> 上手 洋子<sup>†</sup> 西尾 芳文<sup>†</sup>

<sup>†</sup> 徳島大学工学部 〒 105-0123 徳島県徳島市常三島町 2-1

E-mail: †{inukai,t-inoue,uwate,nishio}@ee.tokushima-u.ac.jp

あらまし 粒子群最適化 (Particle Swarm Optimization; PSO) は、群知能の一種であり、魚の群れの動きのシステムを用いたアルゴリズムである。PSO は魚の群れの 1 匹が最適な点を発見すると、群れの残りがその最適な点周辺に集まる性質を用いたアルゴリズムである。アルゴリズムとしては単純で、容易に解を発見できるため多くの応用ができる。ただし、極所解に陥るとそこから抜け出しにくい欠点がある。そこで今回われわれは、捕食者を加えた PSO を提案する。提案手法は、捕食者である大きな魚を入れることにより、PSO を改善するアルゴリズムである。本研究では提案手法と既存手法との効率の違いをコンピュータシミュレーションを用いて確認した。

キーワード PSO, 群知能, 関数最適化, 捕食者

## Investigation of Particle Swarm Optimization with Predator

Norio INUKAI<sup>†</sup>, Takuya INOUE<sup>†</sup>, Yoko UWATE<sup>†</sup>, and Yoshifumi NISHIO<sup>†</sup>

<sup>†</sup> Department of Electrical and Electronic Engineering, Tokushima University

2-1 Minami-Josanjima, Tokushima-shi, Tokushima, 770-8506 Japan

E-mail: †{inukai,t-inoue,uwate,nishio}@ee.tokushima-u.ac.jp

**Abstract** Particle Swarm Optimization(PSO) is known as one of Swarm Intelligence. PSO algorithm is used for the system of fish school. PSO is used nature the rest of the herd is to gather the optimum point around one of fish school is to find the optimal point. PSO algorithm is simple and it is possible to find a solution easily, so it is possible for many applications. However, there is a drawback is hard to get out of localized solution into there. So this time we propose PSO with Predator. This improve the PSO into predator that is big fish.

**Key words** PSO, Swarm Intelligence, Function optimization, Predator

### 1. ま え が き

魚や鳥は群れを作ることにより、餌や敵をすばやく発見している。これは、ベストではないかもしれないが効率は良いものと考えられる。このような群れで素早く発見する方法をモデル化したものが、粒子群最適化 (Particle Swarm Optimization; PSO) である。また同様に、蟻の群れをモデル化したものとしてアントコロニー最適化 (Ant Colony Optimization; ACO) [1] [2] や、ミツバチの採餌行動をモデル化したものとしてミツバチコロニー最適化 (Bee Colony Optimization; BCO) などがある。これらのように、生物の行動を応用した最適化アルゴリズムを総じて群知能 [3] [4] と呼ばれる。

群知能を用いてらる問題として、巡回セールスマン問題 (traveling salesman problem) やナップサック問題 (Knapsack problem), 関数の最適化問題 (function optimization problem) などがある。関数の最適化問題は、関数のこれらの問題を解く場合、全件検索などの従来法より効率がよくなる。これら群知

能の弱点として、最良でない解 (局所解) に陥りやすいことがあげられる。

PSO は粒子の位置と速度の情報だけを使い計算できるため、簡易に実装できる。しかしながら、PSO も群知能の一つであるため、局所解に陥りやすい弱点がある。そこで、PSO を元に魚の群れに特化したアルゴリズムを新たに提案する。この提案手法は、小さい魚が大きい魚に捕食することを加えた手法である。これにより、局所解に陥りやすいという弱点を改善できることが予測される。本研究ではこの手法を、捕食者を加えた PSO (PSO with Predator; PSO-P) と名づける。ここでの捕食者とは、大きな魚である。詳しい手順については、3 章に示す。提案手法では、捕食者の捕食範囲や空腹度など新たなパラメータが定義される。

今回、提案手法である PSO-P と基本的な PSO を比較した。また、提案手法については捕食者の空腹度の変化量は一定として、捕食者の捕食範囲を変化させシミュレーションした。

## 2. 基本的な PSO のアルゴリズム

PSO では、以下の 2 つの式 (1), (2) を用いる。

$$\vec{v}_{k+1} = \vec{a} \otimes \vec{v}_k + \vec{b}_1 \otimes \vec{r}_1 \otimes (\vec{p}_1 - \vec{x}_k) + \vec{b}_2 \otimes \vec{r}_2 \otimes (\vec{p}_2 - \vec{x}_k) \quad (1)$$

$$\vec{x}_{k+1} = \vec{c} \otimes \vec{x}_k + \vec{d} \otimes \vec{v}_{k+1} \quad (2)$$

ここで、

$$\left\{ \begin{array}{l} \vec{v}_k : \text{粒子の速度} \\ \vec{x}_k : \text{粒子の場所} \\ \vec{p}_1 : \text{それぞれの粒子が発見した最良点} \\ \vec{p}_2 : \text{郡全体が発見した最良点} \\ \vec{a}, \vec{b}_1, \vec{b}_2, \vec{c}, \vec{d} : \text{係数} \\ \vec{r}_1, \vec{r}_2 : \text{乱数} \end{array} \right.$$

である。この式は、 $\vec{v}_0, \vec{x}_0$  を初期値とする。

基本的な PSO のアルゴリズムは、以下の手順である。

### Step 1 初期値決定

初期値として、すべての粒子に対して  $\vec{v}_0$  と  $\vec{x}_0$  をランダムに決定

### Step 2 $\vec{p}_1$ の決定

$\vec{p}_1$  を  $\vec{x}_0$  とする

### Step 3 $\vec{p}_2$ の決定

$\vec{p}_2$  をすべての粒子の  $\vec{p}_1$  の中で、もっともよいものに決定

### Step 4 終了判定

$\vec{p}_2$  が閾値以下である場合、ロープを抜けて終了。また、ループ回数が規定回数以上の場合、ロープを抜けて終了。

### Step 5 $\vec{v}_k$ と $\vec{x}_k$ の更新

式 (1), (2) にしたがって  $\vec{v}_k$  と  $\vec{x}_k$  を更新

### Step 6 $\vec{p}_1$ と $\vec{p}_2$ の更新

$\vec{p}_1$  と  $\vec{x}_k$  を比べ、 $\vec{x}_k$  の方がよいとき更新。 $\vec{p}_2$  を更新

### Step 7 ループ

Step 4 に戻る

また、今回 Step 4 の閾値として、すべての粒子の速度が一定値以下である時と考えた。

## 3. PSO-P のアルゴリズム

本研究では、捕食者を加えた PSO (PSO with Predator; PSO-P) を提案する。PSO-P では、粒子を小さい魚の群れ、解を餌場としてそれぞれ定義する。解の最適度は、餌場の大きさとして定義する。これらの条件で、捕食者である大きな魚を加える。

捕食者を入れることのメリットとして、最大でない餌場に集まっていた小さい魚をばらけさせることが可能である。すなわち、PSO の局所解から抜け出しにくいという欠点の改善が期待できる。

PSO-P を、アルゴリズムにすると以下ようになる。

### Step 1 捕食判定

捕食者の周りにいる点に対して、捕食判定を行う。

空腹度  $< (0 \sim 1 \text{ の乱数})$  の場合に捕食とする。捕食の場合は、

Step 2 へ、それ以外は、Step 4 へ進む。

### Step 2 空腹度を増加

空腹度がある程度増やす。この過程を行うことで、捕食量を制限することができる。

### Step 3 再配置

捕食された数の半数を再配置する。これは、今まで群れに属していなかった小さい魚が、新たに群れに加ることを意味する。

### Step 4 空腹度を減少

空腹度を減少させる。この過程は、捕食を再開するために行う。

### Step 5 捕食者の移動

捕食者を式 (1), (2) にしたがって移動。ただし、 $\vec{p}_1 = \vec{x}_k$  とする。

このアルゴリズムを、基本的な PSO のアルゴリズムの Step 6 と Step 7 の間に追加する。また、基本的な PSO のアルゴリズム及び PSO-P のアルゴリズムのフローチャートを図 1 に示す。

最後に、今回の測定では、一次元で考えたため式 (1), (2) を式 (3), (4) のようにする。

$$v_{k+1} = a \times v_k + b_1 \times r_1 \times (p_1 - x_k) + b_2 \times r_2 \times (p_2 - x_k) \quad (3)$$

$$x_{k+1} = c \times x_k + d \times v_{k+1} \quad (4)$$

また、 $a = 0.6, b_1 = b_2 = 1.7, c = d = 1$  とした [5]。

## 4. 結果

表 1 に示す 3 関数について検証する。

表 1 検証する関数。

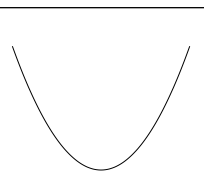
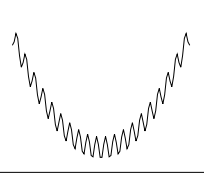
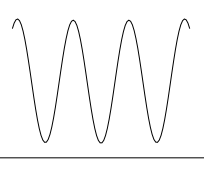
関数	式	グラフ
Sphere	$f(x) = x^2$	
Rastrigin	$f(x) = x^2 - 10 \cos(2\pi x) + 10$	
Griewank	$f(x) = \frac{x^2}{4000} - \cos(x) + 1$	

表 1 の 3 つの関数の最小値は  $f(x) = 0$  で、 $x = 0$  の時である。

粒子数は 100, 1000 及び 2000 について検証した。また、捕食範囲は 0.01, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00 の 11 個について測定した。測定結果は、表 2 に示す。

表 2 よりまず、平均値では Sphere 関数の粒子数が 100 と

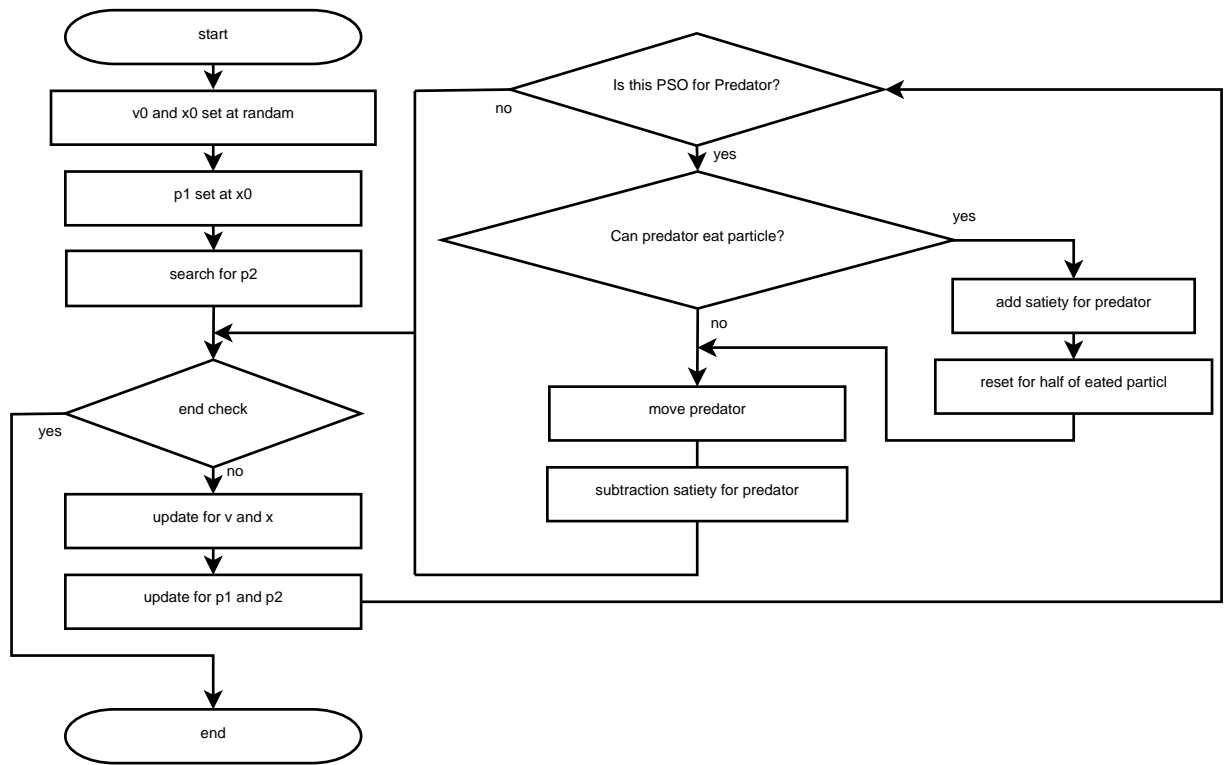


図 1 フローチャート .

2000 の時以外，PSO-P がよい結果となっている．また，最小値や最大値でも PSO-P の方がよいものがある．この結果は，局所解から早くに抜け出せたためと考えられる．また，すべての測定で最小値が発見出来ている．

次に，PSO-P での捕食範囲は，関数により良い結果が出る箇所が異なる．Sphere 関数では，粒子数が 2000 の時 0.6 ~ 0.7 での結果が良いと分かる．しかし，粒子数が 100, 1000 では範囲の影響はほとんど見られない．この結果は，Sphere 関数には局所解が存在しないためであると考えられる．また，標準的な PSO の結果と比較するとほとんど差がないことからそのような考えられる．Rastrigin 関数では，0.6 ~ 1.0 の間である．特に，0.6 周辺と 0.9 周辺の 2ヶ所に集中しているようである．Griewank 関数では，0.1 ~ 0.2 と 0.7 ~ 0.8 の 2ヶ所で良い結果が現れている．このことから，局所解があるときは，複数箇所良い結果になる場所があると考えられる．Rastrigin 関数は，関数の概形が Sphere 関数と似ているため 2ヶ所が合わさるようになったと考えられる．

## 5. ま と め

本研究では，PSO-P の性能を測定した．結果として，基本的な PSO より良い結果となった．しかし，一次元での測定であるため高次元での測定が必要である．今後の課題として多次元化への対応と空腹度の増加量を変化させた場合の測定を行う必要がある．

## 文 献

- [1] M. Dorigo and T. Stutzle, "Ant Colony Optimization," Bradford Books, 2004.
- [2] H. Koshmizu, T. Saito, "Parallel Ant Colony Optimizers

with Local and Global Ants," Proceedings of International Joint Conference on Neural Networks, 2009.

- [3] Blum and Merkle (eds), "Swarm Intelligence," Springer, 2008.
- [4] E. Bounabeau, M. Dorigo and T. Stutzle, "Inspiration for optimization from social insect behavior,"
- [5] Trelea, I. C. "The particle swarm optimization algorithm: convergence analysis and parameter selection", Information processing letters, 85(6), pp. 317325, 2003.

表 2 検証結果 (測定回数は 100 回, 単位はループ回数, 太字はそれぞれの行での最良値) .

関数	粒子数		PSO	PSO with Predator											
				捕食範囲											
				0.01	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00	
Sphere	100	Ave	<b>141.41</b>	142.68	142.68	142.75	142.64	142.64	142.80	143.04	143.04	142.72	142.72	142.94	
		Min	<b>124</b>	125	125	125	125	125	125	125	125	125	125	125	125
		Max	198	<b>173</b>	<b>173</b>	<b>173</b>	<b>173</b>	<b>173</b>	<b>173</b>	<b>173</b>	<b>173</b>	<b>173</b>	<b>173</b>	<b>173</b>	<b>173</b>
	1000	Ave	164.63	164.85	164.73	165.23	165.22	164.46	164.02	163.50	164.17	163.65	163.11	<b>162.88</b>	
		Min	149	<b>147</b>	<b>147</b>	151	151	149	149	149	149	149	149	149	
		Max	193	189	189	189	189	187	187	187	187	<b>181</b>	<b>181</b>	187	
	2000	Ave	<b>170.68</b>	170.75	172.17	171.87	172.19	171.11	171.01	170.98	170.85	173.07	172.44	172.94	
		Min	155	157	158	159	160	159	<b>153</b>	<b>153</b>	<b>153</b>	156	156	156	
		Max	213	195	195	195	197	197	197	<b>192</b>	<b>192</b>	204	204	212	
Rastrigin	100	Ave	165.43	163.69	163.64	163.74	163.26	163.39	163.36	<b>162.69</b>	162.74	163.03	163.26	163.35	
		Min	137	<b>127</b>	<b>127</b>	<b>127</b>	<b>127</b>	<b>127</b>	<b>127</b>	<b>127</b>	<b>127</b>	<b>127</b>	<b>127</b>	<b>127</b>	
		Max	235	219	219	219	219	219	219	<b>211</b>	<b>211</b>	<b>211</b>	<b>211</b>	<b>211</b>	
	1000	Ave	198.73	196.97	197.78	197.69	197.68	197.08	197.88	195.60	196.11	196.02	<b>195.14</b>	194.99	
		Min	169	171	171	171	171	170	170	170	170	170	170	<b>163</b>	
		Max	265	278	278	278	236	236	236	<b>233</b>	236	253	253	253	
	2000	Ave	208.00	208.99	207.76	207.61	207.48	207.26	205.7	<b>203.79</b>	205.20	207.66	205.60	206.62	
		Min	179	181	181	181	183	180	<b>177</b>	<b>177</b>	<b>177</b>	182	181	181	
		Max	256	264	265	265	260	249	255	255	280	280	<b>249</b>	262	
Griewank	100	Ave	301.57	303.09	302.6	303.03	302.27	302.42	300.56	298.85	295.14	293.95	293.62	<b>293.47</b>	
		Min	<b>189</b>	198	198	198	198	198	198	198	198	198	198	198	
		Max	<b>479</b>	549	549	549	549	549	549	496	496	496	496	496	
	1000	Ave	417.91	413.51	397.41	<b>390.60</b>	399.80	406.54	405.83	411.67	398.72	395.12	398.32	407.12	
		Min	308	321	321	316	316	308	308	308	<b>299</b>	<b>299</b>	<b>299</b>	307	
		Max	558	724	507	<b>472</b>	628	628	612	564	529	554	613	613	
	2000	Ave	456.17	451.67	<b>429.83</b>	446.05	438.25	439.16	440.55	444.69	438.88	458.66	450.96	446.07	
		Min	351	343	350	364	356	347	344	350	<b>325</b>	<b>325</b>	337	337	
		Max	694	655	<b>632</b>	730	654	580	691	618	742	820	664	682	