

2匹の戦略的捕食者を加えた粒子群最適化の調査

犬飼 規雄[†] 上手 洋子[†] 西尾 芳文[†]

[†] 徳島大学工学部 〒 770-8506 徳島県徳島市常三島町 2-1

E-mail: †{inukai,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, 魚や鳥など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. しかし, 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, 2匹の戦略的捕食者を加えた PSO を提案する.

キーワード PSO, 群知能, 関数最適化, 捕食者

Investigation of Particle Swarm Optimization with Two Strategic Predators

Norio INUKAI[†], Yoko UWATE[†], and Yoshifumi NISHIO[†]

[†] Department of Electrical and Electronic Engineering, Tokushima University

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

E-mail: †{inukai,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, bird flock and so on. 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, the standard PSO is hard to get out from local minimum. Accordingly, this time we propose PSO with Two Strategic Predators.

Key words PSO, Swarm Intelligence, Function optimization, Predator

1. ま え が き

群知能アルゴリズムとして, アントコロニー最適化 (Ant Colony Optimization; ACO) [1] [2] や, ミツバチコロニー最適化 (Bee Colony Optimization; BCO), 粒子群最適化 (Particle Swarm Optimization; PSO) などがよく知られている. これらの群知能アルゴリズムは, 最適化問題の解法としてよく使われている. 最適化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) などが知られている.

この内, 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, 簡易に実装できました, 多くの応用がある. しかしながら, 多くの群知能アルゴリズムと同様に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.

今回, 捕食者数を2匹として, それぞれの捕食者の行動戦略を変えることによる変化について調査した. 調査法として, 関数の最適化問題を用いて評価を行った.

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.

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

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

ここでの変数の意味は, 以下のとおりである.

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

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 をランダムに決定する。

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、最もよいものに決定する。

Step 4 \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 を更新する。

Step 5 \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する。同様に、 \vec{p}_2 を更新する。

Step 6 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する。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する。ここで、続ける場合 Step 4 に戻る。

また、今回 Step 6 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、解を餌場と仮定する。解の最適度は、餌場の大きさとする。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。ここでは、簡潔にするため捕食者 1 匹の行動についてのみ示す。

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。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

空腹度をある程度増やす。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。

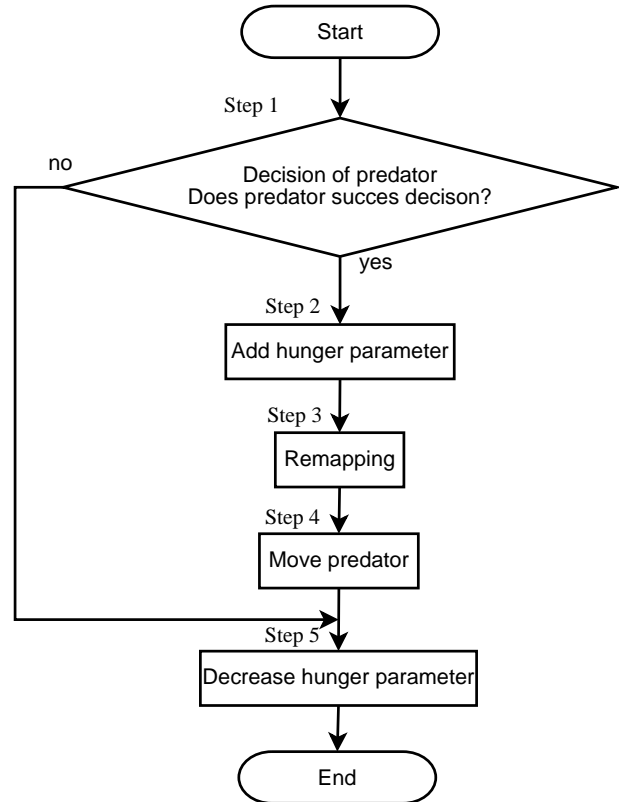


図 2 PSO-P の詳細なフローチャート。

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) にしたがって移動させる。ここで、 \vec{p}_1 と \vec{p}_2 を変化させることで、捕食者の行動戦略を変化させている。行動戦力 0 共に、群全体の最良値を用いいる。行動戦力 1 共に、群全体の一つ前の最良値を用いいる。行動戦力 2 \vec{p}_1 に群全体の一つ前の最良値、 \vec{p}_1 に群全体の最良値を用いいる。

このアルゴリズムを、基本的な PSO のアルゴリズムの Step 5 と Step 6 の間に追加する。また、基本的な PSO 及び PSO-P のフローチャートを図 1 に、PSO-P の詳細なフローチャートを図 2 に示す。

また、PSO のパラメータを、 $a = 0.6, b_1 = b_2 = 1.7, c = d = 1.0$ と定めた [5]。

4. 結果

表 1 に示す 3 関数について検証する。この 3 つの関数の最小値は $f(x) = 0$ で、 $x = 0$ の時である。また、次元は 30 次元で行った。

粒子数は 15, 30 及び 60 について検証し、捕食者の行動戦略は 3 通りで捕食者 2 匹なので、6 通りそれぞれ調べた。捕食範囲は 0.1~1.0 を 0.1 ステップで測定した。測定は、100 回行った。評価方法として、平均値を求め、そこから Cost を求めた。

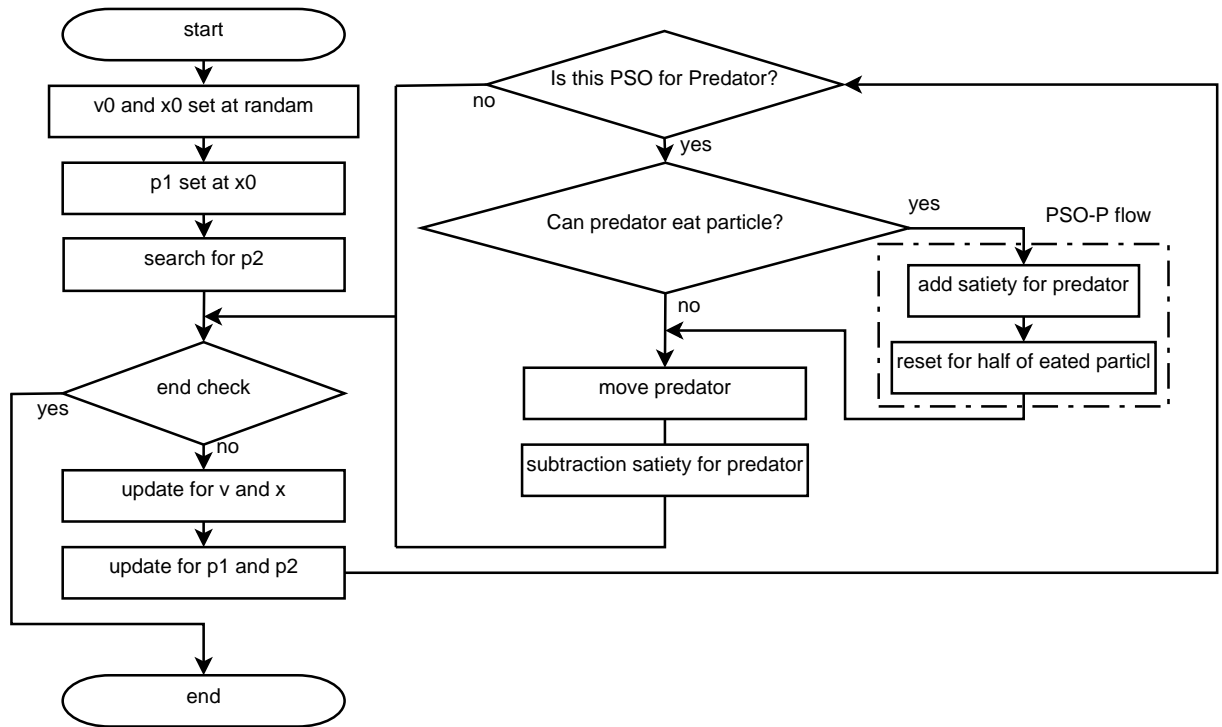


図1 基本的な PSO と PSO-P のフローチャート。

平均値は、最終の結果が表 1 の “Goal for f ” 以下の平均値を用いた。また、 $Cost$ は式 (3) と定義した。

$$Cost = \frac{Particle\ of\ number \times Average}{Success\ rate} \quad (3)$$

測定結果は、表 2, 3, 4 に示す。結果よりも、Rastrigin 関数では、30 粒子・行動戦略 1-1 の捕食半径 0.9 が最も良い結果となっている。また、Griewank 関数では、30 粒子・行動戦略 0-0 の捕食半径 0.1 が最も良い結果となっている。最後に、Rosenbrock 関数では、15 粒子・行動戦略 2-2 の捕食半径 0.2 である。このことから、それぞれの関数での最良値は提案手法となっている。また、関数ごとの特徴として、Rastrigin 関数と Rosenbrock 関数は提案手法が、Griewank 関数では、従来法が良い結果を示している。

ここで、関数の特徴を考える。Rosenbrock 関数は平らな面が広がる関数である。このことから、Rosenbrock 関数の従来法の結果が非常に大きい理由として広がっている面によって粒子の収束が阻害されているからと考えられる。これを、捕食者を用いることで、収束早める働きがあると考えられる。また、Rastrigin 関数及び Griewank 関数は局所解を多く持つ関数である。特性が似ている関数であるが、結果に大きな違いが見られる。このことは、Griewank 関数のほうが局所解が多いことが理由として考えられる。

次に、PSO-P での捕食範囲について考える。先ほどの関数の特性も考慮して、局所解を多く持つ Rastrigin 関数及び Griewank 関数ではものは捕食範囲での変化が見られにくく、平らな面が広がる Rosenbrock 関数では、変化が見られた。このことから、捕食範囲は平らな面が大きいほど有効であると考えられる。

最後に、捕食者の行動戦略について考える。こちらも、捕食

範囲と同様に Rastrigin 関数及び Griewank 関数では大きな変化が見られないが、Rosenbrock 関数では、変化が見られた。変化の方向に違いはあるが、行動戦力 2 の要素が大きな変化に起因しているようである。

5. まとめ

本研究では、PSO-P の性能を測定した。結果として、従来法より良い結果を概ね示した。しかし、関数により違いがあるため、提案手法のさらなる解析が必要である。

謝辞

本研究の一部は、JSPS 科研費 26540127 の助成を受けたものである。

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 1 Optimization Test Functions

name	Formula	Range n	Goal for f [x_{\min}, x_{\max}]
Rastrigin	$f(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^n$	100
Griewank	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^n$	0.1
Rosenbrock	$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-30, 30]^n$	100

表 2 検証結果 (Rastrigin 関数) .

	Standard PSO	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
15 粒子, 0-0	14598.0	13130.4	10606.6	13281.5	13861.6	13921.5	13566.0	11328.3	11611.6	14030.1	12558.8
15 粒子, 0-1	14598.0	12779.9	10461.4	12277.4	13433.1	14516.5	13570.5	11975.5	11743.0	13464.5	12179.8
15 粒子, 0-2	14598.0	15086.4	11441.1	11134.3	12766.7	12033.0	11573.0	13786.0	13320.4	12776.3	10458.0
15 粒子, 1-1	14598.0	12942.4	10534.5	12790.5	13530.0	14218.0	13854.1	11404.4	11986.5	13619.6	11981.9
15 粒子, 1-2	14598.0	14058.1	11429.3	11534.5	12438.2	12652.3	11602.7	13300.1	11936.6	12343.9	11059.2
15 粒子, 2-2	14598.0	11407.0	11175.0	13736.4	13180.3	12466.3	11368.6	12848.9	12815.0	12397.0	14135.8
30 粒子, 0-0	19585.2	15415.6	15770.7	15080.2	16732.5	15790.2	16328.2	15671.7	15262.2	14773.5	15173.2
30 粒子, 0-1	19585.2	15856.1	16308.2	14888.7	16956.6	15780.8	16136.2	14973.1	15144.3	14903.8	14722.6
30 粒子, 0-2	19585.2	15351.3	15093.4	15296.9	16236.3	15909.4	14011.4	14929.9	15535.8	14589.2	15387.6
30 粒子, 1-1	19585.2	15904.2	15612.9	15987.7	16687.0	16377.4	15532.2	15511.5	15120.4	14504.8	14998.5
30 粒子, 1-2	19585.2	15403.6	14859.1	15428.5	15693.8	15853.7	13939.3	14991.7	15204.9	15858.8	15567.1
30 粒子, 2-2	19585.2	15309.1	14885.9	16094.2	16475.5	14697.4	14989.3	16036.6	15291.7	16173.8	15900.1
60 粒子, 0-0	33871.2	25343.6	25904.2	25812.0	25758.2	25011.9	26234.7	25280.1	25277.0	25742.2	26872.0
60 粒子, 0-1	33871.2	26828.2	25940.4	24183.0	25300.3	24826.8	26699.4	24514.2	25283.0	26565.4	26817.2
60 粒子, 0-2	33871.2	25126.2	25631.2	25319.9	25406.3	25014.0	27025.0	25511.9	25040.1	24786.0	24912.8
60 粒子, 1-1	33871.2	26243.4	26954.4	24487.8	25694.4	24495.0	24821.4	25278.8	25216.7	25999.8	26935.5
60 粒子, 1-2	33871.2	26030.6	25316.2	26942.3	25714.3	24942.8	28543.5	26564.5	25293.5	24983.8	24773.4
60 粒子, 2-2	33871.2	25966.8	25749.9	25714.8	26904.9	24921.9	24691.8	26856.3	26067.3	27022.3	26308.4

表 3 検証結果 (Griewank 関数) .

	Standard PSO	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
15 粒子, 0-0	30491.4	45071.3	41835.0	51715.2	39868.7	76339.1	60489.7	59377.1	53853.6	76084.8	56587.8
15 粒子, 0-1	30491.4	43861.7	42686.7	49322.5	40937.0	86173.3	69515.2	59377.1	56010.4	76692.0	63598.3
15 粒子, 0-2	30491.4	45548.3	41595.7	58326.4	47575.3	56947.9	64950.0	63670.1	56387.5	66120.0	39421.5
15 粒子, 1-1	30491.4	44957.2	41145.0	44258.0	41497.4	92318.9	59928.7	59739.7	50734.6	80531.3	58676.9
15 粒子, 1-2	30491.4	43735.7	39759.0	53119.8	47548.5	62783.1	62459.2	49637.9	59144.6	66378.8	39931.1
15 粒子, 2-2	30491.4	41776.8	50782.5	46172.8	60646.3	49772.9	50021.4	55568.1	43236.2	82957.0	34949.1
30 粒子, 0-0	25535.5	22764.2	26316.4	26880.7	27679.0	27284.5	28467.4	32738.5	30685.1	31256.4	30922.4
30 粒子, 0-1	25535.5	23922.9	26404.7	29443.0	27046.4	27502.0	29336.5	31579.1	31109.0	31355.9	30524.2
30 粒子, 0-2	25535.5	29465.9	26316.6	31492.7	30593.4	30369.8	33060.5	37767.3	33744.2	37384.6	40703.6
30 粒子, 1-1	25535.5	23445.7	25353.5	26697.7	28849.1	27809.6	29898.6	32171.0	32492.5	30843.3	29524.9
30 粒子, 1-2	25535.5	33141.4	27043.1	27981.6	28231.5	29964.0	35359.5	35677.2	36084.3	37100.6	39547.9
30 粒子, 2-2	25535.5	25793.3	25014.7	28097.9	24699.0	27626.2	28439.3	27573.0	28297.4	29475.2	30422.5
60 粒子, 0-0	36552.3	33036.0	35049.3	36366.5	37580.8	37370.7	38332.4	39766.9	40692.1	45852.2	43788.7
60 粒子, 0-1	36552.3	33096.8	35884.1	37958.3	36824.7	38097.0	42800.1	40959.2	41202.9	44929.4	45364.0
60 粒子, 0-2	36552.3	38883.2	41920.9	42121.2	43394.4	43823.1	44822.8	44971.9	48847.0	47464.8	47494.6
60 粒子, 1-1	36552.3	33735.9	34655.8	37636.2	38598.6	39319.9	38588.9	43178.1	41295.2	43841.7	42791.1
60 粒子, 1-2	36552.3	38890.7	39391.2	43218.0	43386.6	48439.1	42770.2	44444.2	48837.8	48844.4	46409.0
60 粒子, 2-2	36552.3	33491.4	34573.7	39236.3	38082.2	37662.1	38431.0	44218.8	39656.0	43085.7	41968.8

表 4 検証結果 (Rosenbrock 関数) .

	Standard PSO	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
15 粒子, 0-0	57975.9	54616.1	51330.0	53540.6	74444.4	102079.9	213719.0	96320.0	97645.8	135129.6	273281.3
15 粒子, 0-1	57975.9	54973.3	54570.0	61255.1	68144.0	86413.3	213186.0	82562.3	83938.8	151734.4	110500.0
15 粒子, 0-2	57975.9	49667.4	38298.8	41109.8	39351.4	57806.7	49256.5	59743.2	64908.3	62424.5	71572.0
15 粒子, 1-1	57975.9	56335.4	45720.7	55678.6	71975.1	108257.4	207012.4	76687.5	77739.8	131500.0	109870.4
15 粒子, 1-2	57975.9	44706.4	38012.1	43297.5	38943.7	54732.0	49860.3	57506.7	83182.5	53376.5	75569.4
15 粒子, 2-2	57975.9	47744.9	36081.6	34289.1	43440.0	42453.1	58318.5	58537.2	78051.7	109673.3	99621.3
30 粒子, 0-0	283909.4	222310.1	157402.2	165501.5	92999.6	74131.6	71118.6	51180.7	62172.8	64104.0	58284.4
30 粒子, 0-1	283909.4	213333.7	137120.9	155471.8	101875.4	97326.4	75413.4	51671.3	54792.0	69778.2	49478.9
30 粒子, 0-2	283909.4	280848.6	260777.8	204472.5	245783.0	217173.0	213503.5	243026.4	201524.9	223968.6	191609.4
30 粒子, 1-1	283909.4	205777.2	128827.6	147815.1	105325.8	91838.1	73527.1	52633.7	51016.4	61693.1	51597.4
30 粒子, 1-2	283909.4	277878.0	248572.2	232452.7	256392.6	227658.4	225085.0	260969.0	221570.8	258769.1	171821.0
30 粒子, 2-2	283909.4	294600.6	228590.8	247252.8	208277.4	213167.2	202074.2	157799.2	158413.9	130612.8	157445.8
60 粒子, 0-0	600060.0	584858.9	580916.1	559794.6	557091.9	564825.7	575940.1	597325.0	558023.6	568772.7	541095.8
60 粒子, 0-1	600060.0	581424.6	581612.0	549000.3	560414.0	537830.0	543638.6	558516.0	614518.6	580759.9	536317.4
60 粒子, 0-2	600060.0	601552.5	586399.8	586134.8	571293.0	581738.5	557453.7	556195.9	529420.2	489617.8	508754.7
60 粒子, 1-1	600060.0	588756.6	590347.3	547935.1	558256.8	541844.2	574222.8	547332.1	582581.1	596098.5	505339.5
60 粒子, 1-2	600060.0	606121.2	579802.8	597040.4	542540.2	520534.4	565961.5	515412.0	534063.0	482114.8	516897.2
60 粒子, 2-2	600060.0	594324.0	584268.7	566668.9	545831.3	568215.7	547376.4	557416.0	505179.1	569144.4	555433.6