

## RESEARCH

## Open Access



# Optimization of PID parameters with an improved simplex PSO

Ji-min Li<sup>1</sup>, Yeong-Cheng Liou<sup>2,3</sup> and Li-jun Zhu<sup>1\*</sup>

\*Correspondence:

zhulijun1995@sohu.com

<sup>1</sup>School of Mathematics and

Information Sciences, Beifang

University for Nationalities,

YinChuan, 750021, China

Full list of author information is

available at the end of the article

## Abstract

In this paper, we adopt the '0.618' method to select the compression factor and expansion factor for the simplex particle swarm optimization algorithm. We use our new method to optimize parameters of a Proportion Integral Differential (PID) controller. The experimental results show that our method can effectively solve the slow convergence problem and give a better performance than the conventional PID method does.

**MSC:** 60H35; 93C20; 65C20; 60H15

**Keywords:** simplex particle swarm optimization; '0.618' method; PID automatic regulation system

## 1 Introduction

The Particle Swarm Optimization (PSO) algorithm is an evolutionary computation method, proposed by Kennedy and Eberhart in 1995 [1, 2]. PSO has many advantages, such as simplicity, and a high practical implement ability, to be extended for multi-objective optimization [3–6]. Ono and Nakayama, proposed an algorithm using multi-objective Particle Swarm Optimization (MOPSO) for finding robust solutions against small perturbations of design variables [7]. Gong *et al.* presented a global path planning approach based on multi-objective particle swarm optimization [8]. Shang *et al.* proposed an improved multi-objective particle swarm optimization algorithm [9]. Mostaghim and Teich proposed a Sigma method that decided  $p$ -best for each particle and introduced a disturbance factor [3]. Yu introduced two kinds of improved multi-objective particle swarm optimization algorithms [10, 11]. But it is likely to be entrapped in a local extremum for some complex optimization problems. To overcome the disadvantage of a local extremum and optimize the PSO algorithm performance, expanding its application, Chen proposed a simplex particle swarm optimization algorithm (SPSO) which combines PSO with a strong local search capability method (Simplex Method, SM) [12]. In order to enhance the performance of SPSO, a new strategy for the compression factor and expand factor of SPSO is proposed in this paper. We brought our new method into the Proportion Integral Differential (PID) controller to optimize its parameters, and the results indicate the superiority of our method.

## 2 Improved simplex particle swarm optimization algorithm

### 2.1 The basic particle swarm optimization algorithm

The PSO algorithm adopts a speed- and position-based searching model, each particle representing an alternative solution; a swarm of particles fly through a  $n$ -dimensional search space with a certain velocity. The velocities are dynamically changed according to the experience of the flight. Assuming that the whole particle swarm has  $m$  particles, the position and velocity of the  $i$ th particle in the  $n$ -dimensional search space can be represented as  $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$  and  $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$ . Putting  $x_i$  into the objective function we can calculate its fitness value,  $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$  being the best position that the  $i$ th particle has experienced,  $P_{\text{best}}$  is a globally optimal position choosing from all particles,  $p_i$ . In each generation, the velocity  $v_i$  and position  $x_i$  of particle are updated by the following formulas:

$$v_{id} = \omega \cdot v_{id}(t) + c_1 \cdot \text{rand}() (p_{id} - x_{id}) + c_2 \cdot \text{rand}() (p_{gd} - x_{id}) \quad (d = 1, 2, \dots, D), \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (d = 1, 2, \dots, D), \quad (2)$$

where  $\omega$  is an inertia weight, usually the value is within  $[0.4, 1.2]$ ,  $c_1$  and  $c_2$  are acceleration constants, the so-called learning factors,  $\text{rand}()$  is for random numbers in the range of  $[0, 1]$ .

### 2.2 Improved simplex particle swarm optimization algorithm

PSO is a global optimization method, it does not need any prior information when it seeks the optimal solution in the solution space. But in some complex cases, PSO may easily fall into a local extremum. Instinctively, local search technologies were concerned to solve this problem. Instead, PSO can employ some simplex method to guide its search direction while the local search can take the PSO search result as the initial point, due to the randomness of the particles' searching movement, the potential global optimal solution nearly may be skipped. So SPSO employs a repeated searching strategy round the current optimal solution to increase the probability of finding the global optimal solution [12, 13]. In our method, we introduce a compression process when the reflection of the worst solution is inferior to the second worst solution, and an expansion process when the reflection of the worst solution is superior to the second worst solution; then a new solution instead of the worst solution involves the next simplex searching. The compression factor and the expansion factor are two random numbers, they are less and greater than 1, respectively. In our method, we exploit the '0.618' method to select the compression factor and the expansion factor.

### 2.3 New algorithm

- (1) Initialize each parameter values, evaluate the initial fitness value of each particle according to the objective function;
- (2) calculate each particle's velocity and position by (1) and (2);
- (3) update and store every particle's historical optimal location and fitness value, update and store the historical global optimal position and optimum;
- (4) if the current global optimum is not better than the historical one, go to (7), otherwise, go to (5);

- (5) call the improved simplex method (SM), if the SM end conditions are satisfied, substitute the results of SM into PSO;
- (6) if it is necessary, update and store the historical optimal location and optimal fitness of the each back substitution particle; update and store the historical global optimal position and optimum;
- (7) if the termination conditions (deviations or iteration times) have been reached, output the global optimal position and optimum, end the iteration. Otherwise, go to (2).

### 3 Optimize PID parameters with the improved SPSO

#### 3.1 The optimization problem of PID controller parameters

In an industrial control system, the output of some control object under the action of a step signal appears in an S-shaped rising curve; in this case, one can use a model of second-order inertia and delay to describe it. Its transfer function is

$$W(s) = \frac{Ke^{-T_3s}}{(T_1s + 1)(T_2s + 1)}.$$

In order to maintain for the control output  $y$  a constant value under the effect of a disturbance, usually, the PID controller is used to form a constant value control system.

When the production process is stable, namely, the object properties are stable,  $K$ ,  $T_1$ ,  $T_2$ ,  $T_3$  are constant, respectively, at this time, the adjusted PID parameters can stay the same, while, when changes appear frequently in the production process, such as chemical reaction in chemical engineering or a load change in a power plant, the constant PID parameters usually cannot achieve an optimal control result.

Since the computer has a good capability of calculation and control flexibility and can bring about an automatic control of DDC, it is to be possible to adjust the PID controller parameters by some computer system [14, 15].

Assume that the control  $u$  and deviation  $e$  satisfy the following difference equation:

$$u(n) = K_p \left[ e(n) + \frac{1}{T_I} \sum_{K=0}^n e(K)T + T_d \frac{e(n) - e(n-1)}{T} \right],$$

or

$$u(n) = u(n-1) + K_p \left\{ [e(n) - e(n-1)] + \frac{T}{T_I} e(n) + \frac{T_d}{T} [e(n) - 2e(n-1) + e(n-2)] \right\},$$

where  $u(n)$  is the  $n$ th control action,  $e(n)$  is the  $n$ th deviation,  $T$  is the sampling period,  $K_p$  is the proportional gain coefficient,  $T_I$  is a constant integral time, and  $T_d$  is a constant differential time.

Our optimization task is searching for proper  $K_p$ ,  $T_I$ ,  $T_d$  of PID, which allow the objective function  $Q = \int_0^{+\infty} |e|t dt$  to be minimum. It belongs to the multivariate function optimization problems of nonlinear programming, and up to now it cannot use a mathematical expression to describe the relationship between the objective function and  $K_p$ ,  $T_I$ ,  $T_d$ . In this paper, we exploit the improved particle swarm optimization algorithm to deal with this case.

**Table 1** The output results of system simulation

$T$	$X_1$	$X_5$	$T$	$X_1$	$X_5$
0.1	0.1101215	9.8898785	1.6	9.9883500	0.0116500
0.2	0.8970125	9.1029875	1.7	9.9981210	0.001879
0.3	3.8164750	6.1835250	1.8	10.0118	-0.011816
0.4	5.1986732	4.8013268	1.9	10.0103	-0.010275
0.5	7.9087764	2.0912236	2.0	10.0124	-0.012357
0.6	8.8758920	1.1241080	2.1	10.0176	-0.017586
0.7	9.8284680	0.1715320	2.2	10.0183	-0.018272
0.8	10.0175180	-0.0175180	2.3	10.0145	-0.014451
0.9	10.1256500	-0.1256500	2.4	10.0133	-0.013263
1.0	10.0345000	-0.0345000	2.5	10.0131	-0.013063
1.1	10.0186500	-0.0186500	2.6	10.0126	-0.012556
1.2	10.0100411	-0.0100411	2.7	10.0115	-0.011496
1.3	9.9952210	0.0047790	2.8	10.0084	-0.008376
1.4	9.9598610	0.0401390	2.9	10.0072	-0.007192
1.5	9.9496476	0.0503524	3.0	10.0061	-0.006071

### 3.2 The simulation results

We input the following data:

- variable number  $N = 3$ ; calculation accuracy  $E = 0.01$ ;
- compression factor of 0.618; expansion factor of 1.618;
- the parameters of the controlled object  $T_1 = 0.44$  s,  $T_2 = 0.44$  s,  $T_3 = 0.12$  s;
- total number of print  $L_3 = 30$ ; the control system input value  $R = 10$ ;
- the PID parameters  $K_p, T_I, T_d$ ; the initial values  $X(1, 0) = 1.5, X(2, 0) = 0.88, X(3, 0) = 0.11$ ;
- the group size of SPSO is 30, the termination number is 50,  $v_d^{\max}$  is set to 15% up and down,  $c_1 = c_2 = 2$ , the inertia weight  $\omega = 0.8$ .

We calculated the optimal value of the PID controller parameters:

$$K_p = 1.69762, \quad T_I = 0.772664 \text{ s}, \quad T_d = 0.229206 \text{ s}.$$

Correspondingly, we also found some interesting information between the output  $X_1$  and deviation value  $X_5$  (shown in Table 1).

The system output overshoot volume  $E = 1.26\%$ , transient time  $T_p = 1.0$  s.

At this point, according to the conventional simplex method we calculate the results as follows:

$$K_p = 1.69768, \quad T_I = 0.772671 \text{ s}, \quad T_d = 0.229212 \text{ s}.$$

The system output overshoot volume  $E = 1.85\%$ , the transient time  $T_p = 1.2$  s.

According to the engineering design method of the standard  $I$  system, the system output overshoot volume  $E = 4.3\%$ , transient time  $T_p = 1.76$  s.

Comparison results show our optimization method can attain a better result than the conventional PID method does, and it also effectively solves the problem of slow convergence.

### 4 Conclusion

In this paper, we developed an improved SPSO method to solve the optimization of PID parameters. Simulation results show that the proposed method has a good convergence efficiency and a good accuracy. It can effectively improve the quality of a dynamic system.

**Competing interests**

The authors declare that they have no competing interests.

**Authors' contributions**

All authors contributed equally and significantly in writing this paper. All authors read and approved the final manuscript.

**Author details**

<sup>1</sup>School of Mathematics and Information Sciences, Beifang University for Nationalities, YinChuan, 750021, China.

<sup>2</sup>Department of Information Management, Cheng Shiu University, Kaohsiung, 833, Taiwan. <sup>3</sup>Center for General Education, Kaohsiung Medical University, Kaohsiung, 807, Taiwan.

**Acknowledgements**

This work is supported by Natural Science Foundation of Ningxia (No. NZ14101); National Natural Science Foundation of China (61362033).

Received: 30 July 2015 Accepted: 13 August 2015 Published online: 09 October 2015

**References**

1. Kennedy, J, Eberhart, RC: Particle swarm optimization. In: Proceedings IEEE International Conference on Neural Networks, IEEE Service Center, Perth, Australia, pp. 1942-1948 (1995)
2. Eberhart, RC, Kennedy, J: A new optimizer using particle swarm theory. In: Proceedings the Sixth International Symposium on MicroMachine and Human Science, IEEE Service Center, Nagoy, Japan, pp. 39-43 (1995)
3. Hu, X, Eberhart, R, Shi, Y: Particle swarm with extended memory for multiobjective optimization. In: Proceedings of 2003 IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, April, pp. 193-197 (2003)
4. Mostaghim, S, Teich, J: Strategies for finding good local guides in multiobjective particle swarm optimization. In: Proceedings of 2003 IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, April, pp. 26-33 (2003)
5. Mostaghim, S, Teich, J: The role of e-dominance in multiobjective particle swarm optimization methods. In: Proceedings of IEEE Congress on Evolutionary Computation CEC'2003, Canberra, Australia, pp. 1764-1771 (2003)
6. Huang, VL, Suganthan, PN, Liang, JJ: Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems. *Int. J. Intell. Syst.* **21**(2), 209-226 (2006)
7. Ono, S, Nakayama, S: Multi-objective particle swarm optimization for robust optimization and its hybridization with gradient search. In: IEEE International Conference on Evolutionary Computations, pp. 1629-1636 (2009)
8. Gong, DW, Zhang, JH, Zhang, Y: Multi-objective particle swarm optimization for robot path planning in environment with danger sources. *J. Comput.* **6**(8), 1554-1561 (2011)
9. Shang, JT, Sun, TY, Liu, CC: An improved multi-objective particle swarm optimizer for multi-objective problems. *Expert Syst. Appl.* **37**(8), 5872-5886 (2010)
10. Guolin, Y: Multi-objective estimation of distribution algorithm based on the simulated binary crossover. *J. Conver. Inf. Technol.* **7**(3), 110-116 (2012)
11. Yu, G: An improved differential evolution algorithm for multi-objective optimization problems. *Int. J. Adv. Comput. Technol.* **3**(9), 1106-1113 (2011)
12. Chen, G-c, Yu, J-S: Simplex particle swarm optimization algorithm and its application. *J. Syst. Simul.* **18**(4), 862-865 (2006)
13. Xiong, W-l, Xu, B-g, Zhou, Q-m: Study on optimization of PID parameter based on improved PSO. *Comput. Eng.* **24**, 41-43 (2005)
14. Xiong, G-l: Digital Simulation Control System, pp. 159-162. Tsinghua University Press, Beijing (1984)
15. Yuan, Y-X, Sun, W-Y: Optimization Theory and Method, pp. 69-75. Science Press, Beijing (1999)

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)