zenghui wang <wangzengh@gmail.com>

CINC 2010 Acceptance Letter

CINC 2010 <cinc2010@easychair.org> Tue, Jun 1, 2010 at 6:30 PM
To: wangzengh@gmail.com

Dear Authors

Thank you for your submission to CINC 2010. We are pleased to inform you that, according to the reports from anonymous reviewers, the following distinguished work from you has been accepted for CINC 2010, with the publisher of IEEE, which will be indexed by EI Compendex and ISTP.

You are kindly reminded with the following important notes:
1. Open the link and find a lot of information of paper submission and registration.
http://www.ieee-ett.cn/cinc2010/registration/

2. In order to make high quality of Proceedings, the camera-ready version should follow format. Kindly download from here
http://www.ieee-ett.cn/cinc2010/registration/

3. After Finish the final Paper, you can prepare a Copyright Release Form. The copyright should download, print, write author names, paper title, sign a name and date, and scanned it to PDF format
http://www.ieee-ett.cn/cinc2010/registration/IEEE%20COPYRIGHT%20FORM.pdf

4. Kindly download the registration form and pay for it,
http://www.ieee-ett.cn/cinc2010/registration/

and send both registration form and a scanned receipt from your bank to above Email CINC2010@163.com before June 20, 2010. If you have not paid for your paper in that time, your paper will not be published.

5. The e-copy official acceptance Letter could be download though
http://www.ieee-ett.cn/cinc2010/registration/Acceptance_letter_ei.doc
Write you write author names, paper title and print it

Kindly send Final paper (doc format), copyright, registration form and a scanned receipt to CINC2010@163.com before June 20, 2010

Sincerely,
Program Committee of CINC 2010
E-mail: CINC2010@163.com
URL: http://www.ieee-ett.cn/cinc2010/registration/

The topic of your paper is well suitable for our international conference. And your demonstration is very good. Furthermore, the structure of your paper is in standard form. Some experts suggest that you try to avoid Chinglish Engslish in your paper.

Answer the following question in scale (1-7, 1-low and 7 high)
Quality of the abstract: 5
Relevance to the conference: 4
Introduction and motivation: 4
Presentation of the "state of the art": 4
Description, originality of the own contribution: 5
Presentation of the results: 4
Conclusions and future work: 4
Readability, quality of the English: 4
Quality of the figures: 4
Quality of format:5
Overall Paper Recommendation (1-7, 1 strong reject, 7strong accept) accepted as regular paper:5

Paper: 212
Title: Generalized Predictive Control based on Particle Swarm Optimization for Linear/Nonlinear Process with constraints

Generalized Predictive Control based on Particle Swarm Optimization for Linear/Nonlinear Process with constraints

Zenghui Wang
School of Engineering
University of South Africa
Florida 1710, South Africa
Email: wangz@unisa.ac.za

Yanxia Sun
French South African Technical Institute in Electronics
Tshwane University of Technology,
Pretoria 0001, South Africa
Email: sunyanxia@gmail.com

***Abstract*—This paper presents an intelligent generalized predictive controller (GPC) based on particle swarm optimization (PSO) for linear or nonlinear process with constraints. We propose several constraints for the plants from the engineering point of view and the cost function is also simplified. No complicated mathematics is used which originated from the characteristics of PSO. This method is easy to be used to control the plants with linear or/and nonlinear constraints. Numerical simulations are used to show the performance of this control technique for linear and nonlinear processes, respectively. In the first simulation, the control signal is computed based on an adaptive linear model. In the second simulation, the proposed method is based on a fixed neural network model for a nonlinear plant. Both of them show that the proposed control scheme can guarantee a good control performance.**

***Keywords*-Generalized Predictive Control; Particle Swarm Optimization; Intelligent control; Constraint, Nonlinear Process;**

I. INTRODUCTION

Model predictive control (MPC) has received a great deal of attention in the past several decades [1]-[4]. This is because the strategy uses the future behavior of the system output which makes the MPC have good robustness properties. If the plant uncertainty is high, MPC should turn to adaptive control. There are many problems in adaptive MPC [8]-[10]. Moreover, it is difficult to deal with the plants with constraints. Although there are some research results, most of the constrained MPC algorithms turn to linear matrix inequalities to take care of the constraints, for example, the control schemes in [11], [12], [13]. They are difficult to be realized in the microprocessors as there are some matrix operations especially the matrix inverse operation. They also involve a lot of mathematic operations which are difficult to be understood by the engineers.

GPC has been proved to be particularly effective and easy to design methods [5]-[7]. In this paper, a generalized predictive controller (GPC) based on constrained particle swarm optimization (PSO) is proposed to deal with linear/nonlinear process with constraints. We propose several constraints for the plants from the engineering point of view. No complicated mathematics is used, which is determined by the characteristics of PSO. This method is easy to be used to control the plants with linear or/and nonlinear constraints. Numerical simulations are used to show the performance of this control scheme for linear and nonlinear processes, respectively. In the first simulation, the control signal is computed using an adaptive linear model. In the second simulation, the proposed method is based on a fixed neural network model.

II. OVERVIEW OF GPC

GPC developed by Clark based on the following model [3]:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \xi(t)/\Delta \qquad (1)$$

where $y(t)$, $u(t)$ and $\xi(t)$ are the output, input and disturbance/noise, respectively.

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{n_a} q^{-n_a} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{n_b} q^{-n_b} \\ \Delta &= 1 - q^{-1} \end{aligned}$$

The GPC control law is obtained by minimizing the following cost function:

$$J = \sum_{j=1}^{N} [\hat{y}(t+j) - w(t+j)]^2 + \lambda \sum_{j=1}^{N_u} [\Delta u(t+j-1)] \qquad (2)$$

where $\hat{y}(t+j)$ is the predicted output at time $t+j$ based on the available input/output data at time t; N and N_u are referred to as the prediction horizon and the control horizon, respectively; $\lambda \geq 0$ is a control increment weight; and $w(t+1)$ is the soften tracking sequence:

$$\begin{cases} w(t) = y(t) \\ w(t+j) = \alpha w(t+j-1) + (1-\alpha) y_r(t), \qquad j = 1, \cdots, N \end{cases}$$

where $y_r(t)$ is the set-point at time t; α $(0 \leq \alpha < 1)$ is a soften factor.

The j-th step forthcoming prediction output and control increment can be obtained through the recursive computing of the following Diophantine equation:

$$1 = E_j A \Delta + q^{-1} F_j, \qquad j = 1, \cdots, N \qquad (3)$$

$$E_j B = G_j + q^{-1} H_j, \qquad j = 1, \cdots, N \qquad (4)$$

where

$$\begin{aligned} E_j &= e_0 + e_1 q^{-1} + \cdots + e_{j-1} q^{-(j-1)}, \\ F_j &= f_{j0} + f_{j1} q^{-1} + \cdots + f_{jn_a} q^{-n_a}, \\ G_j &= g_0 + g_1 q^{-1} + \cdots + g_{j-1} q^{-(j-1)}, \\ H_j &= h_{j0} + h_{j1} q^{-1} + \cdots + h_{jn_b-1} q^{-(n_b-1)} \\ & j = 1, \cdots, N \end{aligned}$$

Minimization of (2) with respect to the future control increments $\Delta(t + j)$, together with the receding horizon strategy, leads to

$$\Delta u(t) = [1, 0, \cdots, 0](G^T G + \lambda I)^{-1} G^T (W - F) \quad (5)$$

where the matrix G is of the form

$$G = \begin{bmatrix} g_0 & 0 & \cdot & \cdot & \cdot & 0 \\ g_1 & g_0 & 0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & g_0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{N-1} & \cdot & \cdot & \cdot & \cdot & g_{NNu} \end{bmatrix}$$

$W = [w(t+1), w(t+2), \cdots, w(t+N)]^T$ and $F = [f(t+1), f(t+2), \cdots, f(t+N)]^T$ where $f(t+j) = F_j y(t) + H_j \Delta u(t-1)$.

Defining $[k_1, k_2, \cdots, k_N] = [1, 0, \cdots, 0](G^T G + \lambda I)^{-1} G^T$, (5) can be rearranged as the following form

$$\begin{aligned} \Delta u(t) = & \sum_{j=1}^{N} [k_j (1 - \alpha^j)] y_r(t) - \sum_{j=1}^{N} [k_j (F_j(q^{-1}) - \alpha^j)] y(t) \\ & - \sum_{j=1}^{N} [k_j H_j(q^{-1})] \Delta u(t-1). \end{aligned} \quad (6)$$

III. Brief introduction of constrained Particle swarm optimization using penalty factor method

Many optimization problems can be represented as the following optimization problem:

$$\begin{aligned} & J(X_i) = min(f(X_i)), \quad X_i = [x_i^1, x_i^2, \ldots, x_i^n]. \\ & \text{subject to } g_j(X_i) \le 0, \quad j = 1, 2, \ldots, k. \end{aligned} \quad (7)$$

Here $f(\cdot)$ is the objective function without constraints; $X_i(t)$ denotes the position vector of particle i consisting of n variables and $g_j(X_i)$ is the j^{th} constraint. The formulation of the constraints in Eq. (7) is not restrictive, since an inequality constraint of the form $g_j(X_i) \ge 0$, can also be represented as $-g_j(X_i) \le 0$, and an equality constraint, $g_j(X_i) = 0$, can be represented by two inequality constraints $g_j(X_i) \le 0$ and $-g_j(X_i) \le 0$.

The canonical particle swarm algorithm works by iteratively searching in a region and is concerned with the best previous success of each particle, the best previous success of the particle swarm and the current position and velocity of each particle [14]. Every candidate solution of $J(X)$ is called a "particle". The particle searches the domain of the problem, according to

$$\begin{aligned} V_i(t+1) &= \omega V_i(t) + c_1 R_1 (P_i - X_i(t)) \\ &\quad + c_2 R_2 (P_g - X_i(t)), \end{aligned} \quad (8)$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \quad (9)$$

where $V_i = [v_i^1, v_i^2, \ldots, v_i^n]$ is the velocity of particle i; $X_i = [x_i^1, x_i^2, \ldots, x_i^n]$ represents the position of particle i; P_i represents the best previous position of particle i indicating the best discovery or previous experience of particle i; P_g represents the best previous position among all particles indicating the best discovery or previous experience of the social swarm; ω is the inertia weight that controls the impact of the previous velocity of the particle on its current velocity and is sometimes adaptive [15]; R_1 and R_2 are two random weights whose components r_1^j and r_2^j ($j = 1, 2, \ldots, n$) are chosen uniformly within the interval $[0, 1]$ which might result in an explosion of the particle trajectory; c_1 and c_2 are the learning rates which are positive constant parameters. Generally the value of each component in V_i should be clamped to the range $[-v_{max}, v_{max}]$ to control excessive roaming of particles outside the search space.

For the constrained PSO using penalty method, PSO is not need to be changed except the objective function is changed to

$$F(X) = f(X) + \sum_{j=1}^{k} p_j |g_j(X)|. \quad (10)$$

Here

$$p_j = \{ \begin{array}{ll} h_j & if \quad g_j(X) > 0 \\ 0 & otherwise \end{array}, \quad (11)$$

and h_i is positive penalty factor. Without loss of generality, h_i is set as a big positive constant. In this paper, $h_i = 10^8$.

IV. GPC based on constrained PSO for linear/nonlinear process with constraints

As can be seen from the brief introduction of GPC, the core parts of GPC are equations (1) and (2). Other mathematic operations, such as equations (3)–(5), are used to get (6) with minimizing the cost function (2). Hence, some intelligent optimization algorithms can be used directly to minimize the cost function (2) and get the control signal which can make the control scheme simpler than the classic GPC.

Moreover, according the procedure of calculating the control signal (6), it is difficult to quantify the requirements for the plant just using the objective function (2). However, the requirements can be given using some constrains, for example, some constrains about $\Delta u(k+j-1)$ ($j = 1, 2, \cdots, N_u$). The cost function can also be simplified by getting rid of the term related to control signal, that is, $\lambda \sum_{j=1}^{N_u} [\Delta u(t+j-1)]$,

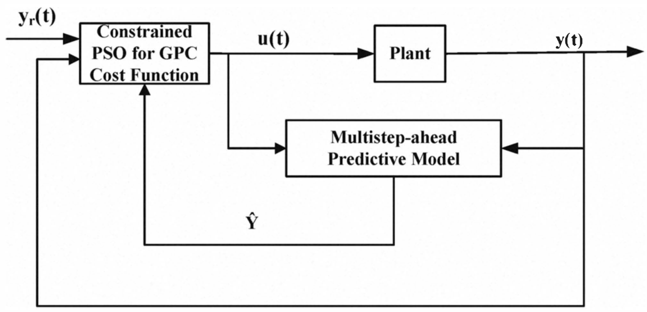

Figure 1. Structure of constrained GPC based on PSO

which is replaced by the constraints. The cost function of the intelligent GPC becomes

$$J = \sum_{j=1}^{N} [\hat{y}(t+j) - w(t+j)]^2 \tag{12}$$

The diagram of constrained GPC based on PSO is shown in Fig. 1.

Several constraints for the plants from the engineering point of view are proposed to take the place of $\lambda \sum_{j=1}^{N_u} [\Delta u(t+j-1)]$. They are
1) the range control signals: $R_1 \le u \le R_2$ where R_1 and R_2 are determined by characteristics of the plant.
2) the first order differential value of the control signals:$|\Delta u| \le D_1$ where D_1 is a positive constant.
3) the second order differential value of the control signals:$|\Delta\Delta u| \le D_2$ where D_2 is a positive constant.

As can be seen from the definitions of these constraints, they have explicit physical meanings and easy to be understood. If it is necessary, other constraints can be used to meet the requirements of the plant, such as the constraints about output of plant. According to the properties of constrained PSO, the program complexity increases very little if several constraints are added which is better than other techniques, for example, LMI technique.

In this paper, we only focus on improving the control scheme other than the identification of the system parameters. Hence, any method of obtaining the dynamical models, which include linear model or nonlinear model, and off-line or on-line model, can be used for the proposed method if the model is good enough.

V. Simulation Studies

A. Linear plant

Firstly, a linear plant is used to show the control performance of this proposed method. The plant model is same as (1) where $A(q^{-1}) = 1 - 2.5q^{-1} + 0.7q^{-2}$, $B(q^{-1}) = 1 + 0.5q^{-1}$ and $\xi(t) = 0.2random()$. As the modulus of one of the characteristic roots is 2.1787, which is greater than 1, this plant is unstable. The predictive model is obtained on-line using recursive least squares technique. If the constraints parameters $R_1 = -1.5$, $R_2 = 1.5$, $D_1 = 0.5$ and $D_2 = 0.2$, the system response of this unstable plant is shown in Fig. 2. In Fig. 2, the rectangle signal is the expected trajectory. It shows that a good control performance is obtained using the proposed method.

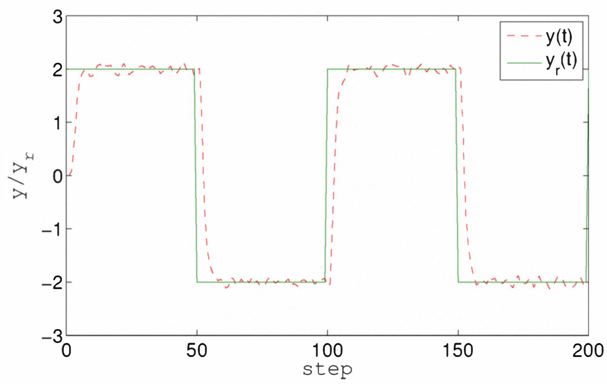

Figure 2. System response

B. Nonlinear plant

To demonstrate the control performance for nonlinear plant, the following nonlinear plant [16] is considered.

$$y(t) = \frac{y(t-1)y(t-2)[y(t-1)+2.5]}{1+y(t-1)^2+y(t-2)^2} + u(t-1) \tag{13}$$

where $n_a = 2$ and $n_b = 1$.

The plant is modelled by a feed forward NN (neural network) with 3–5–1 structure. The input signal applied to plant (13) is a finite sequence of uniformly distributed random variables with range [-3, 3]. Thus it generates input/output samples (patterns), which will be used to train the NN. Among the samples, 200 samples are used as the training NN data, while the rest 200 samples are used as the testing NN data. The testing error is shown in Fig. 3. As can be seen from Fig. 3, the generalization capability of the trained NN are very good. In this simulation, the predictive model is fixed as the trained NN. For the proposed method, the constraints parameters are also set as $R_1 = -1.5$, $R_2 = 1.5$, $D_1 = 0.5$ and $D_2 = 0.2$. If the constrained PSO is used for GPC, the system response is shown in Fig. 4. As can be seen from Fig. 2, the proposed method can guarantee a good control performance.

VI. Conclusion

This paper presents an intelligent generalized predictive controller (GPC) based on particle swarm optimization (PSO) for linear or nonlinear process with constraints. No complicated mathematics is used according to the characteristics of PSO. This method is easy to be used to control the linear/nonlinear plants with constraints. Numerical simulations are used to show the performance of GPC based on constrained PSO for linear and nonlinear

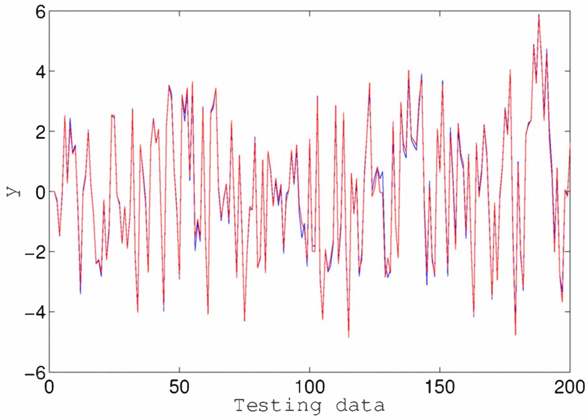

Figure 3. Comparison using testing of NN

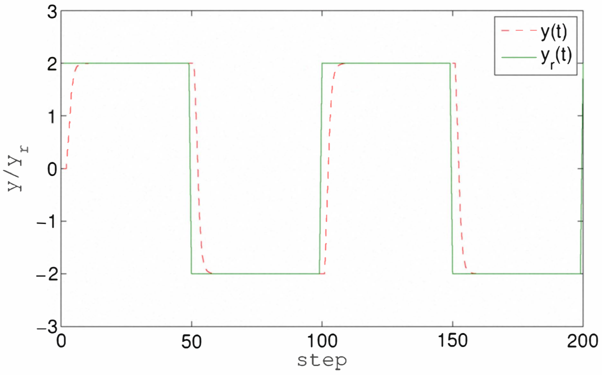

Figure 4. System response

processes, respectively. The simulation results shown that this proposed method can be easily used and has good control performance.

Acknowledgment

This work was supported by the Natural Science Foundation of China (Grant No. 60774088, 60874032) and the Foundation of the Application Base and Frontier Technology Research Project of Tianjin (Grant Nos. 08JCZDJC21900).

References

[1] J. Richalet, A. Rault, J. L. Testud and J. Papon, Model predictive heuristic control: applications to industrial processes. *Automatica* 1978; **14**(5):413-428.

[2] V. Peterka, Predictor-based self-tuning control. *Automatica* 1984; **20**(1):39-50.

[3] D. W. Clark, C. Mohtadi and P. S Tuffs, Generalized predictive control, part1: the basic algorithm, part2: extensions and interpretations. *Automatica* 1987; **23**(2):137-160.

[4] H. Demircioglu and P. J. Gawthrop, Continous generalized predictive control (CGPC). *Automatica* 1991; **27**(1):55-74.

[5] J. M. Martin-sanchez and S. L. Shah, Multivariable adaptive predictive control of binary distillation column. *Automatica* 1984; **20**(5):607-620.

[6] Clark DW, Application of generalized predictive control to industrial process. *IEEE Control Systems Magazine* 1988; **8**(2):49-55.

[7] H. Demircioglu, E. Karasu, Generalized predictive control–a practical application and comparison of discrete- and continuous-time versions. *IEEE Control Systems Magazine* 2000; **20**(5):36-47,

[8] N. K. Sinha and S. Puthenpura, Choice of the sampling interval for the identification of continuous-time systems from samples of input/output data. *IEE Proceedings D* 1985; **132**(6):263-267.

[9] D. W. Clark, Self-tuning control of nonminimum-phase systems. *Automatica* 1984; **20**(5):501-518.

[10] K. J. Astrom, P. Hagander and J. Sternby, Zeros of samples systems. *Automatica* 1984, **20**(1):31-38.

[11] Z. Wan, M. V. Kothare, Efficient robust constrained model predictive control with a time varying terminal constraint set, *System & Control Letter*, **48**: 375–383, 2003.

[12] B. Ding, Y. Xi, S. Li, A synthesis approach of on-line constrained robust model predictive control, *Automatica*, 2004, **40**:163–167.

[13] P. Boucher, D. Dumur, S. A. S. Font and R. Moritz, LMI approach of constrained optimization in generalized predictive control, *Proceedings of the 1998 IEEE International Conference on Control Applications*, Trieste, Italy 1-4 September 1998: 1160–1164.

[14] J. Kennedy,R. Eberhart, 1995. Particle Swarm Optimization. *In: Proc. IEEE Int. Conf. Neural Networks*. Perth, Australia, 942–948.

[15] B. Liu, L. Wang, Y. H. Jin, F. Tang, D. X. Huang, Improved particle swarm optimization combined with chaos. *Chaos, Solitons and Fractals*, 2005, **25**: 1261–1271.

[16] Y. Song, Z. Chen and Z. Yuan, New Chaotic PSO-Based Neural Network Predictive Control for Nonlinear Process, *IEEE Tran. Neural Networks*, 2007, **18**(2):595–600.

sun yanxia <sunyanxia@gmail.com>

ICSI 2013 notification for paper 164

ICSI 2013 <icsi2013@easychair.org> Fri, Mar 1, 2013 at 5:45 PM
To: Yanxia Sun <sunyanxia@gmail.com>

Congratulations! On behalf of the ICSI 2013 international program committee (IPC), the IPC chair, and technical committee, we are very pleased to inform you that your paper:

Paper ID: 164
Author(s): Yanxia Sun
Title: Local and global search based PSO algorithm
----
has been accepted as an ORAL paper for presentation at International Conference on Swarm Intelligence (ICSI 2013), June 12-15, 2013, Harbin, China, and for inclusion in the conference proceedings published by Springer's Lecture Notes in Computer Science (LNCS), http://www.springer.com/computer/lncs?SGWID=0-164-0-0-0

This email provides you with the information you require to revise your paper and submit it for review and inclusion in the proceedings.

Here are the steps you need follow to resubmit the final version of your paper:

1. Please see the Springer's LNCS FORMATTING REQUIREMENTS and REVIEWERS' COMMENTS for your paper, which are intended to help you to improve your paper before resubmitting it for final publication. The listed comments should be addressed carefully in your revision, as this acceptance is conditional on your appropriate response to the formatting requirements and reviewers' comments.

2. Each paper must not exceed 10 pages including figures and references (papers beyond eight pages are subject to page surcharge) and must not less than 6 pages. Your final paper MUST be submitted by March 31, 2013. All papers must be re-submitted even if the reviewers indicated that no change is required.

3. In order for your paper to be published in the ICSI 2013 conference proceedings, a signed Springer LNCS Copyright Form (Consent to Publish) must be submitted for each your accepted paper.

4. The deadline of the conference registration is MARCH 31, 2013. Please register your paper to the conference by following the steps at http://www.ic-si.org/registration.html as early as possible.

If you have any question regarding registration and the reviews of your paper, please feel free to contact us at icsi2013@ic-si.org.

We look forward to seeing you in Harbin, China in this June.

Sincerely,

Program Committee Chair of ICSI 2013:
Yuhui Shi

General Chairs of ICSI 2013:
Russell C. Eberhart, Guihua Xia, Ying Tan

REVIEWERS' COMMENTS

---------------------- REVIEW 1 --------------------
PAPER: 164
TITLE: Local and global search based PSO algorithm

AUTHORS: Yanxia Sun, Zenghui Wang and Barend Jacobus van Wyk

OVERALL EVALUATION: 1 (weak accept)
REVIEWER'S CONFIDENCE: 3 (medium)

----------- REVIEW -----------
The paper proposed an improved PSO. It can be accepted after being revised according to the conference format. Some English errors should be revised carefully.

----------------------- REVIEW 2 ---------------------
PAPER: 164
TITLE: Local and global search based PSO algorithm
AUTHORS: Yanxia Sun, Zenghui Wang and Barend Jacobus van Wyk

OVERALL EVALUATION: 2 (accept)
REVIEWER'S CONFIDENCE: 4 (high)

----------- REVIEW -----------
This paper is interesting and in the topic of this conference. But the presentation is not conformed to the conference paper. Paper writing should be improved by a native speaker. The analysis of your algorithm or method is too weak, in particular, lacking of appropriate theoretical analysis and reasoning. The experiments and results are also weak so it is hardly sufficient to verify your proposed method in current form. If possible, more experiments and some comparisons with other up-to-date methods should be added to back your claims in your revision to expand your experiments and analysis further.
The paper format is not conformed to the springer LNCS paper; please revise the presentation of this paper as carefully as you can.

Local and Global Search Based PSO Algorithm

Yanxia Sun[1,*], Zenghui Wang[2], and Barend Jacobus van Wyk[1]

[1] Department of Electrical Engineering, Tshwane University of Technology, Pretoria 0001, South Africa
[2] Department of Electrical and Mining Engineering, University of South Africa, Florida 1710, South Africa
{sunyanxia,wangzengh,vanwykb}@gmail.com

Abstract. In this paper, a new algorithm for particle swarm optimisation (PSO) is proposed. In this algorithm, the particles are divided into two groups. The two groups have different focuses when all the particles are searching the problem space. The first group of particles will search the area around the best experience of their neighbours. The particles in the second group are influenced by the best experience of their neighbors and the individual best experience, which is the same as the standard PSO. Simulation results and comparisons with the standard PSO 2007 demonstrate that the proposed algorithm effectively enhances searching efficiency and improves the quality of searching.

Keywords: Local search, global search, particle swarm optimisation.

1 Introduction of PSO

PSO is an evolutionary computation technique developed by Kennedy and Eberhart [1] in 1995: it is a population-based optimisation technique, inspired by the motion of bird's flocking, or fish schooling. The particle swarms are social organizations whose overall behavior relies on some sort of communication amongst members, and cooperation. All members obey a set of simple rules that model the communication within the flock, between the flocks and the environment. Each solution is a "bird" in the flock and is referred to as a "particle". PSO is not largely affected by the size and nonlinearity of the problem, and can converge to the optimal solution in many problems [2-5] where most analytical methods fail to converge. It can, therefore, be effectively applied to different optimisation problems.

The standard particle swarm algorithm works by iteratively searching in a region and is concerned with the best previous success of each particle, the best previous success of the particle swarm as a whole, the current position and the velocity of each particle [4]. The particle searches the domain of the problem, according to

$$V_i(t+1) = \omega V_i(t) + c_1 R_1 (P_i - X_i(t)) + c_2 R_2 (P_g - X_i(t)), \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{2}$$

[*] Corresponding author.

Y. Tan, Y. Shi, and H. Mo (Eds.): ICSI 2013, Part I, LNCS 7928, pp. 129–136, 2013.

where $V_i = \left[v_i^1, v_i^2, \cdots v_i^n\right]$ is the velocity of particle i ; $X_i = \left[x_i^1, x_i^2, \cdots x_i^n\right]$ represents the position of particle i ; P_i represents the best previous position of particle i (indicating the best discoveries or previous experience of particle i); P_g represents the best previous position among all particles (indicating the best discovery or previous experience of the social swarm); ω is the inertia weight that controls the impact of the previous velocity of the particle on its current velocity and is sometimes adaptive. R_1 and R_2 are two random weights whose components r_1^j and r_2^j $(j = 1, 2 \cdots, n,)$ are chosen uniformly within the interval $[0,1]$ which might not guarantee the convergence of the particle trajectory; c_1 and c_2 are the positive constant parameters. Generally the value of each component in V_i should be clamped to the range $[-v_{\max}, v_{\max}]$ to control excessive roaming of particles outside the search space.

Among these parameters, the inertia weight ω plays an important role and affects the global and local search ability of PSOs. If the value of ω is too big, the global search ability of PSO will be improved, but its local search ability will not be adequate. Otherwise, if the value of ω is small, the global search ability will decrease and the particles easily fall in premature. Some parameters of adaptive PSOs has been proposed but these usually change the inertia weight: ω is large at the beginning of the search procedure and ω decreases as time increased [7, 13]. However, there is a similar problem with the fixed inertia weight method: 1) at the beginning, the local search ability is not effective as ω is big; while 2) the global search ability is not satisfactory at the end of the search procedure as ω becomes small. To balance the local search and global search ability at the same time, a new particle swarm optimisation algorithm is proposed which can perform the local and global search es simultaneously.

In the proposed algorithm, the particles are divided into two groups. The velocity of the first group of particles is only influenced by the best experience of its neighbors. And the velocity of the second group is influenced by both the best experience of its neighbors and its own best experience. The rest of this paper is arranged as follows: In Section 2, the proposed algorithm is described. Section 3 describes the problems used to evaluate the new algorithm and the results are obtained. Finally, the concluding remarks appear in Section 4.

2 Local and Global Search Based PSO Algorithm

Referring to equation (1), the right side consists of three parts: the first is the previous velocity of the particle; the second and third are those parts contributing to the change in the velocity of a particle. As explained in [7], without these two parts, the particles will keep on flying at the current speed in the same direction until they hit the boundary. PSO will not find an acceptable solution unless there are acceptable solutions on their flying trajectories. But this is a rare case. On the other hand, referring to equation (1) without the first part, the flying particles' velocities are only determined by their current positions and their best positions in its history. At the same time, each other particle will be flying toward its weighted centroid of its own best position and the global best position of the population [8]. Some authors have suggested adjustments to the parameters of the PSO algorithm: adding a random component to the inertia weight [9, 10], using a secondary PSO to find the optimal parameters of a primary PSO [11], and adaptive critics [12]. From our literature study and simulation experience, the optimum is often found near the global best experience in various optimisation problems. To help the particles to enhance searching the region around the global best experience, the first group particles are separated from the whole set of particles to search the area around the global best experience. Then equations (1) and (2) will be altered to

$$V_i(t+1) = 0.5 \times \omega \times V_i(t) + c_1 R_1 (P_g - X_i(t)) + c_2 R_2 (P_g - X_i(t)), \tag{3}$$

$$X_i(t+1) = X_i(t) + V_i(t+1). \tag{4}$$

As can be seen from equation (3), the particles will focus on searching the area around the best experience among their neighbors.

The particles in the second group will continue to the search the global experience of the swarm and its own best experience according to equations (1) and (2), which are the same as the standard PSO.

The following procedure can be used for implementing the proposed particle swarm algorithm:

1) Initialize the swarm, assign a random position in the problem hyperspace to each particle and calculate the fitness function which is yielded by the optimisation problem whose variables are corresponding to the elements of particle position coordinates.

2) The particles in the first group search the area according to equations (3) and (4). Meanwhile, those in the second group search the area according to equations (1) and (2).

3) Evaluate the fitness function for each particle.

4) For each individual particle, compare the particle's fitness value with its previous best fitness value. For each individual particle, compare the particle's fitness value with its previous best fitness value. If the current value X_i is better than the previous best value P_i, then set P_i as X_i.

5) Repeat steps 2) - 4) until the criterion for stopping is met (e.g., maximum number of iterations or a sufficiently good fitness value).

3 Numerical Simulation

To demonstrate the efficiency of the proposed technique, six well-known benchmarks are used to compare the proposed method and the standard PSO 2007 (Matlab version compiled in 2011) [14]. These six optimisation problems were used as shown in Table 1. Their parameters are given in Table 2. These six are famous test functions for minimization methods; each of them has several local minima. In the numerical simulation of the proposed LGPSO method and standard PSO, the particle swarm population size is set floor($10+2\sqrt{D}$). Here D is the dimension of the optimisation problems and the function floor(A) rounds the elements of float number A to the nearest integers less than or equal to A. The rest of the parameters are as follows: inertia weight $\omega=\frac{1}{(2\log 2)}\approx 0.7213$, learning rates $c_1=c_2=0.5+\log 2$, and velocity Vmax set to the dynamic range of the particle in each dimension. The topology of LGPSO is the same as the standard PSO 2007 (SPSO 2011) [14]. It should be noted that the initial variables are set random float numbers in the range [0, 1] to check the effect of big search range. The maximum number of function evaluations is 2000 for these two methods with 100 independent runs. The optimisation statistical analysis of these two algorithms is reported in Table 3. The evolutionary curves of LGPSO and the standard PSO 2007 are depicted in Figures 1-6.

Table 1. Functions used to test the effects of the LGPSO method

Problem	Objective functions
Rosenbrock	$f(x)=\sum_{i=1}^{D}(100(x_{i+1}-x_i^{\ 2}))+(x_i-1)^2)$
Ackley	$f(x)=20+e-20e^{-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}}-e^{-\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)}$
Griewank	$f(x)=\frac{1}{4000}\sum_{i=1}^{D}(x_i-100)^2-\prod_{i=1}^{D}\cos(\frac{(x_i-100)}{\sqrt{i}})+1$
Salomon	$f(x)=\cos(2\pi\sqrt{\sum_{i=1}^{D}x_i^2})+0.1\sqrt{\sum_{i=1}^{D}x_i^2}+1$
Rotated-hyper-ellipsoid	$f(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i}x_j)^2$
Quartic function	$f(x)=\sum_{i=1}^{D}ix_i^4+rand()$
Alpine	$f(x)=\sum_{i=1}^{D}\left\lvert x_i\sin(x_i)+0.1x_i\right\rvert$
Levy	$f(x)=\sin^2(\pi x_1)+\begin{pmatrix}\sum_{i=1}^{D-1}(x_i-1)^2\left(1+10\sin^2(\pi x_i+1)\right)\\+(y_{D-1}-1)^2(1+\sin^2(2\pi x_{D-1}))\end{pmatrix}$

Table 2. Functions parameters for the test problems

Problem	Dimension	Search range	Initial range
Rosenbrock	30	±500	[0, 1]
Ackley	30	±500	[0, 1]
Griewank	30	±500	[0, 1]
Salomon	30	±500	[0, 1]
Rotated-hyper-ellipsoid	30	±500	[0, 1]
Quartic function	30	±500	[0, 1]
Alpine	30	±500	[0, 1]
Levy	30	±500	[0, 1]

Table 3. Comparison between standard PSO 2007 and LGPSO

Problem	Method	best	Mean	Std.dev	Worst
Rosenbrock	Standard PSO 2007	122.3422	222.7063	31.1602	343.6297
Rosenbrock	LGPSO	122.0898	183.5776	24.3594	261.8783
Ackley	Standard PSO 2007	1.8158	2.3669	0.2861	3.0259
Ackley	LGPSO	1.2924	2.0563	0.3337	2.9711
Griewank	Standard PSO 2007	0.0411	0.0788	0.0226	0.1827
Griewank	LGPSO	0.0247	0.0584	0.0172	0.1110
Salomon	Standard PSO 2007	0.2999	0.2999	1.0235e-004	0.3005
Salomon	LGPSO	0.2999	0.2999	6.9229e-006	0.2999
Rotated hyper-ellipsoid	Standard PSO 2007	38.1115	144.2199	67.4747	432.1835
Rotated hyper-ellipsoid	LGPSO	12.1226	48.0499	30.0913	143.8851
Quartic function	Standard PSO 2007	1.8093	5.8692	2.5883	15.6572
Quartic function	LGPSO	1.3892	3.7318	1.5546	8.2722
Alpine function	Standard PSO 2007	0.9011	1.9667	0.6130	4.0312
Alpine function	LGPSO	0.4702	1.4589	0.5381	3.5732
Levy function	Standard PSO 2007	0.4363	0.7213	0.1312	1.1353
Levy function	LGPSO	0.2605	0.5791	0.1267	0.8630

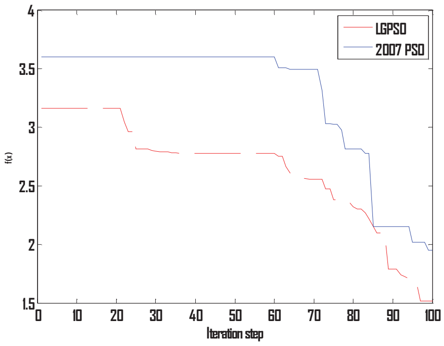

Fig. 1. Comparison for Rosenbrock function

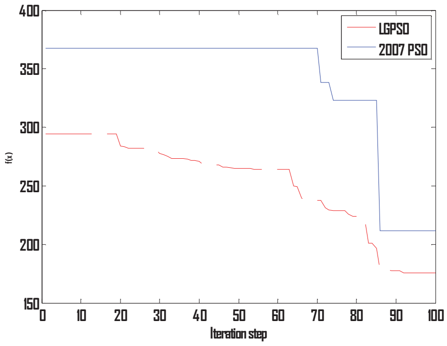

Fig. 2. Comparison for Ackley function

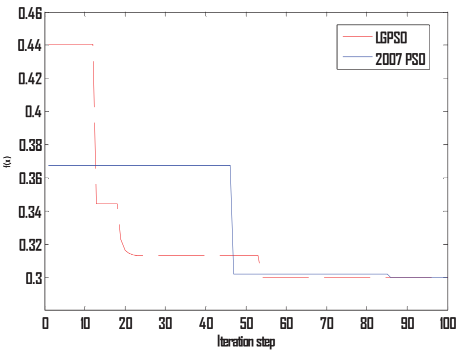

Fig. 3. Comparison for Griewank function

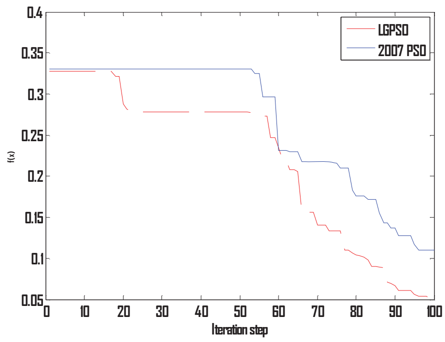

Fig. 4. Comparison for Salomon function

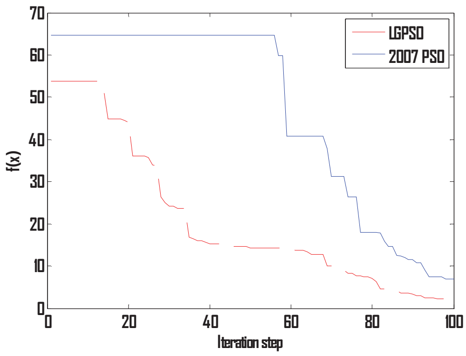

Fig. 5. Comparison for Rotated hyper-ellipsoid function

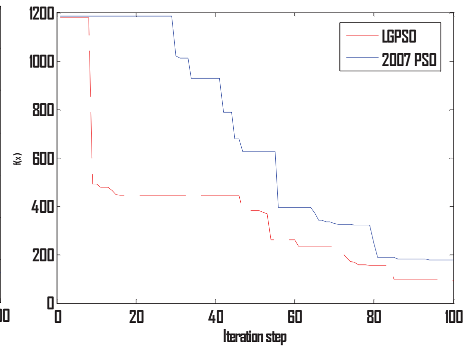

Fig. 6. Comparison for Quartic function

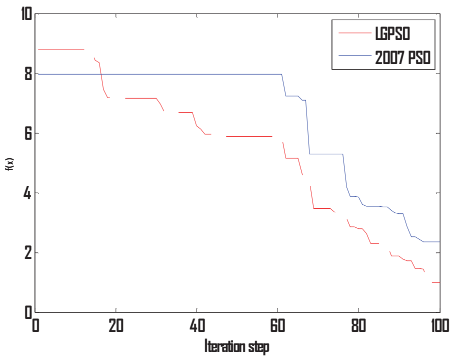

Fig. 7. Comparison for Alpine function

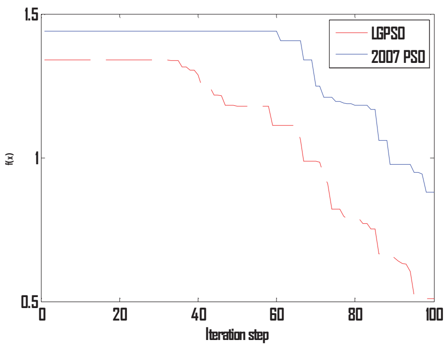

Fig. 8. Comparison for Levy function

As can be seen from Table 3, for the test functions, the best, mean, standard deviation and worst results obtained by the LGPSO are better than the results gained from the standard PSO 2007. The optimization performance of the proposed method is also more stable than the standard PSO 2007 according to the statistical analysis of mean and standard deviation. From Figures 1-8, the optimisation performance is better when the procedure begins, as the local search is added into the algorithms. The simulation results obtained by the LGPSO are better than the results from the standard PSO 2007, which means the final solutions obtained from the LGPSO are more closely focused on the best solution than those from the standard PSO 2007.

4 Conclusion

In this paper, a local and global search based particle swarm optimisation (LGPSO) method was proposed to improve the optimisation performance of the PSO. In this new model, the first group of particles focused on the search around the global best experience while the second group particles are influenced by both the best experience of their group and their own best experience. The simulations showed that the proposed method can achieve good optimisation performance no matter whether at the beginning or at the end of the search period. Moreover, the complexity of the proposed algorithm is not increased over that of the Standard PSO 2007 while the performance of the proposed FCPSO is more stable and more accurate than the Standard PSO 2007.

Acknowledgements. This work was supported by China/South Africa Research Cooperation Programme (No. 78673 & CS06-L02), South African National Research Foundation Incentive Grant (No. 81705), SDUST Research Fund (No. 2010KYTD101) and Key Scientific Support Program of Qingdao City (No. 11-2-3-51-nsh).

The authors thank the Matlab version codes of standard PSO 2007 compiled by Mahamed Omran in 2011, which are available at the Particle Swarm Central website [14].

References

1. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimisation. In: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948 (1995)
2. Hu, X., Shi, Y., Eberhart, R.: Recent Advances in Particle Swarm. In: Congress on Evolutionary Computation, pp. 90–97. IEEE Service Center, Piscataway (2004)
3. Huang, C.M., Huang, C.J., Wang, M.L.: A Particle Swarm Optimisation to Identifying the ARMAX Model for Short term Load Forecasting. IEEE Transactions on Power Systems 20, 1126–1133 (2005)
4. Clerc, M.: Particle Swarm Optimisation. ISTE Publishing Company (2006)
5. Nedjah, N., Mourelle, L.D.M.: Systems Engineering Using Particle Swarm Optimisation. Nova Science Publishers (2007)
6. Aihara, K., Takabe, T., Toyoda, M.: Chaotic Neural Networks. Physics Letter A 144(6-7), 333–340 (1990)
7. Shi, Y., Eberhart, R.: A Modified Particle Swarm Optimiser. In: IEEE International Conference on Evolutionary Computation, pp. 69–73. IEEE Press, Piscataway (1998)
8. Zhang, W.J., Xie, X.F.: DEPSO: Hybrid Particle Swarm with Differential Evolution Operator. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Washington DC, USA, pp. 3816–3821 (2003)
9. Mohagheghi, S., Del Valle, Y., Venayagamoorthy, G., Harley, R.: A Comparison of PSO and Back Propagation for Training RBF Neural Networks for Identification of a Power System with STATCO. In: Proceedings of IEEE Swarm Intelligence Symposium, pp. 381–384 (June 2005)
10. del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J., Harley, R.G.: Particle Swarm Optimisation: Basic Concepts, Variants and Applications in Power Systems. IEEE Transactions On Evolutionary Computation 12(2), 171–195 (2008)
11. Doctor, S., Venayagamoorthy, G., Gudise, V.: Optimal PSO for Collective Robotic sSearch Applications. In: Proceeding IEEE Congress on Evolutionary Computation, Portland, Oregon, USA, pp. 1390–1395 (2004)
12. Venayagamoorthy, G.: Adaptive Critics for Dynamic Particle Swarm Optimisation. In: Proceedings of IEEE International Symposium on Intelligence Control, Taipei, Taiwan, pp. 380–384 (September 2004)
13. Liu, B., Wang, L., Jin, Y.H., Tang, F., Huang, D.X.: Improved Particle Swarm Optimisation Combined with chaos. Chaos, Solitons and Fractals 25, 1261–1271 (2005)
14. Kennedy, J., Clerc, M., et al.: Particle Swarm Central (2012), `http://www.particleswarm.info/Programs.html`

sun yanxia <sunyanxia@gmail.com>

ICSI 2013 notification for paper 175

ICSI 2013 <icsi2013@easychair.org> Sat, Mar 2, 2013 at 7:05 AM
To: Yanxia Sun <sunyanxia@gmail.com>

Congratulations!
On behalf of the ICSI 2013 international program committee (IPC), the IPC chair, and technical committee, we are pleased to inform you that your paper:

Paper ID: 175
Author(s): Yanxia Sun
Title: Cask theory based parameter optimization for Particle Swarm Optimization
----
has been accepted conditionally as a POSTER at International Conference on Swarm Intelligence (ICSI 2013), June 12-15, 2013, Harbin, China, and for possible inclusion in the conference proceedings published by Springer's Lecture Notes in Computer Science (LNCS), http://www.springer.com/computer/lncs?SGWID=0-164-0-0-0 if you can revise your paper according to the reviewers' comments carefully and register your paper to the ICSI 2013 conference before March 31, 2013.

This email provides you with the information you require to revise your paper and submit it for review and inclusion in the proceedings.

Here are the steps you need follow to resubmit the final version of your paper:

1. Please see the Springer's LNCS FORMATTING REQUIREMENTS and REVIEWERS' COMMENTS for your paper, which are intended to help you to improve your paper before resubmitting it for final publication. The listed comments should be addressed carefully in your revision, as this acceptance is conditional on your appropriate response to the formatting requirements and reviewers' comments.

2. Each paper must not exceed 10 pages including figures and references (papers beyond eight pages are subject to page surcharge) and must not less than 6 pages. Your final paper MUST be submitted by March 31, 2013. All papers must be re-submitted even if the reviewers indicated that no change is required.

3. In order for your paper to be published in the ICSI 2013 conference proceedings, a signed Springer LNCS Copyright Form (Consent to Publish) must be submitted for each your accepted paper.

4. The deadline of the conference registration is MARCH 31, 2013. Please register your paper to the conference by following the steps at http://www.ic-si.org/registration.html as early as possible.

If you have any question regarding registration and the reviews of your paper, please feel free to contact us at icsi2013@ic-si.org.

We look forward to seeing you in Harbin, China in this June.

Sincerely,

Program Committee Chair of ICSI 2013:
Yuhui Shi

General Chairs of ICSI 2013:
Russell C. Eberhart, Guihua Xia, Ying Tan

REVIEWERS' COMMENTS

---------------------- REVIEW 1 ---------------------
PAPER: 175
TITLE: Cask theory based parameter optimization for Particle Swarm Optimization
AUTHORS: Zenghui Wang and Yanxia Sun

----------- REVIEW -----------
Parameters optimized for certain tasks must be better for the tasks in general. However, it increases computational costs. Showing only the performance does not have meaning. Important point is to discuss on the balance of performance and computational cost from practical point of view.

There are many papers that discuss adaptive parameters or optimized parameters for evolutionary computation. Survey for these related works is too weak.

---------------------- REVIEW 2 ---------------------
PAPER: 175
TITLE: Cask theory based parameter optimization for Particle Swarm Optimization
AUTHORS: Zenghui Wang and Yanxia Sun

----------- REVIEW -----------
This paper is interesting and in the topic of this conference. But the presentation is not conformed to the conference paper. Paper writing should be improved by a native speaker. The analysis of your algorithm or method is too weak, in particular, lacking of appropriate theoretical analysis and reasoning. The experiments and results are also weak so it is hardly sufficient to verify your proposed method in current form. If possible, more experiments and some comparisons with other up-to-date methods should be added to back your claims in your revision to expand your experiments and analysis further.
The paper format is not conformed to the springer LNCS paper; please revise the presentation of this paper as carefully as you can.

Cask Theory Based Parameter Optimization for Particle Swarm Optimization

Zenghui Wang[1,*] and Yanxia Sun[2]

[1] Department of Electrical and Mining engineering, University of South Africa, Florida 1710, South Africa
[2] Department of Electrical engineering, Tshwane University of Technology, Pretoria 0001, South Africa
{wangzengh,sunyanxia}@gmail.com

Abstract. To avoid the bored try and error method of finding a set of parameters of Particle Swarm Optimization (PSO) and achieve good optimization performance, it is desired to get an adaptive optimization method to search a good set of parameters. A nested optimization method is proposed in this paper and it can be used to search the tuned parameters such as inertia weight ω, acceleration coefficients c_1 and c_2, and so on. This method considers the cask theory to achieve a better optimization performance. Several famous benchmarks were used to validate the proposed method and the simulation results showed the efficiency of the proposed method.

Keywords: PSO, Parameter Optimization, Try and Error method, Nested Optimization method, Cask theory.

1 Introduction

Particle Swarm Optimization (PSO) was developed by Kennedy and Eberhart [1]. This algorithm is inspired by the social behavior of a flock of migrating birds trying to reach an unknown destination. All members obey a set of simple rules that model the communication within the flock, between the flocks and the environment. Each solution is a "bird" in the flock and is referred to as a "particle". PSO has attracted a lot of attention as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions [2, 4-7]. The formula of PSO is realized by two update functions:

$$V_i(t+1) = \omega V_i(t) + c_1 R_1 (P_i - X_i(t)) + c_2 R_2 (P_g - X_i(t)), \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1). \tag{2}$$

Here $V_i = \left[v_i^1, v_i^2, \cdots v_i^n\right]$ is the velocity of particle i; $X_i = \left[x_i^1, x_i^2, \cdots x_i^n\right]$ represents the position of particle i; P_i represents the best previous position of particle i

---

[*] Corresponding author.

Y. Tan, Y. Shi, and H. Mo (Eds.): ICSI 2013, Part I, LNCS 7928, pp. 137–143, 2013.

(indicating the best discoveries or previous experience of particle i); P_g represents the best previous position among all particles (indicating the best discovery or previous experience of the social swarm); ω is the inertia weight that controls the impact of the previous velocity of the particle on its current velocity and is sometimes adaptive. R_1 and R_2 are two random weights whose components r_1^j and r_2^j $(j=1,2\cdots,n,)$ are chosen uniformly within the interval $[0,1]$ which might not guarantee the convergence of the particle trajectory; c_1 and c_2 are the positive constant parameters. Generally the value of each component in V_i should be clamped to the range $[-v_{\max}, v_{\max}]$ to control excessive roaming of particles outside the search space.

The generalized procedure of applying standard PSO 2011 (SPSO 2011) [8] is

1) Initialize the swarm and assign a random position in the problem hyperspace to each particle and calculate the fitness function which is given by the optimization problem whose variables are corresponding to the elements of particle position coordinates; and set the topology of the whole particles.

2) The particles search the area according to equations (1) and (2); check the velocity and position of particles to find whether they violate the boundaries.

3) Evaluate the fitness function for each particle.

4) For each individual particle, compare the particle's fitness value with its previous best fitness value. For each individual particle, compare the particle's fitness value with its previous best fitness value. If the current value X_i is better than the previous best value P_i, then set P_i as X_i.

5) Change the topology if the optimization performance is not improved in a certain number of iterations.

6) Repeat steps 2)-5) until a stopping criterion is met (e.g., maximum number of iterations or a sufficiently good fitness value).

As can be seen from (1) and (2), there are several parameters which should be determined before PSO was applied. Similar as most of the evolutionary optimization algorithms, the parameters of PSO need to be chosen carefully to achieve good optimization performance. The parameters are often chosen based try and error method as different optimization problems have different characteristics and the parameters should not be same to achieve good optimization results. Hence, it is desired to find a suitable set of parameters of PSO without using the bored try and error method. For the evolutionary optimization algorithms, there are some methods optimizing the parameters of the optimization algorithms which are called meta-optimization. Meta-optimization is reported to have been used as early as in the late 1970s by Mercer and Sampson for finding optimal parameter settings of a genetic algorithm [9]. There are some meta-optimizations [10], [11], [12]. For different meta-optimizations, there are different performance indexes.

In this paper, an automatic parameters searching method is proposed based on the particle swarm optimization algorithm and the cask theory. The rest of this paper is arranged as follows: Section 2 presents the proposed algorithm with details. Simulations and comparison are given in Section 3. Finally, the concluding remarks appear in Section 4.

2 Cask Theory Based Parameter Optimization

As the optimization performance depends on the optimization problems, the parameters of optimization algorithms should also depend on the optimization problems, which means different optimization problems should have different sets of parameters of optimization algorithms. As the optimization algorithms can find optimal or sub-optimal solution for the optimization problems, the optimization algorithms can also be used to find the optimal or sub-optimal parameters for PSO. Similar as the optimization procedure, the objective function or criteria related to the parameters of PSO must be defined firstly. There is an important theory is cask theory or barrel theory in Management Science [3]. The cask theory describes that the cubage of a cask is dependent on the shortest wood plate as shown in Fig. 1. This method takes the worst case as the performance criteria and it is possible to make the optimization performance not worse than the achieved one.

Fig. 1. Cask theory (www.baike.com)

The parameter optimization procedure is same with the standard one as mentioned in Section 1. The core of the parameter optimization is defining the objective function or criteria. The followings are the factors, which should be considered, when design the objective function for PSO parameter optimization:

1) Important parameters of PSO should be chosen and they will be the inputs of the objective function.
2) The optimization problem should be considered as the implicit objective as the parameters are used to achieve good optimization performance for the optimization problem.
3) The optimization performance should be stable when the obtained parameters are implemented.
4) The output of the objective function should follow cask theory to guarantee the worst optimization performance is not too bad.

Here, without loss of generality, the algorithm of the SPSO 2011 [8] is chosen as the optimization algorithm whose parameters (inertia weight ω, and acceleration coefficients c_1 and c_2) will be optimized and the SPSO 2011 with fixed parameter is used to optimize these parameters. Hence, for 1) the inputs of the objective function are the inertia weight ω, the acceleration coefficients c_1 and c_2. For 2), the optimization problem will be the target of the SPSO 2011 with variant parameters (VSPSO 2011). For 3) and 4), the optimization problem should be optimized several runs by VSPSO to make sure the results are not stochastic; and the worst fitness value is chosen as the output of the objective function which follows the cask theory.

After the set of parameters are obtained, the normal procedure of PSO will be used to optimize the optimization problems.

3 Numerical Simulation

To demonstrate the efficiency of the proposed technique, eight well-known benchmarks are used to compare the proposed method and the standard PSO 2011 (Matlab version) [8]. The eight optimization problems were used as shown in Table 1. The parameters of these optimization problems are given in Table 2. These eight optimization problems are famous test functions for minimization methods and each of them has high dimension and several local minima. In the numerical simulation of SPSO 2011 with fixed parameters, the particle swarm population size is set floor(10 + $2\sqrt{D}$). Here D is the dimension of the optimization problems and function floor(A) rounds the elements of float number A to the nearest integers less than or equal to A. The rest of the parameters are as follows: inertia weight $\omega = \frac{1}{(2\log 2)} \approx 0.7213$, learning rates $c_1 = c_2 = 0.5 + \log 2$, and velocity Vmax set to the dynamic range of the particle in each dimension. For VSPSO 2011, the inertia weight ω, the acceleration coefficients c_1 and c_2 are the parameters to be optimized and all the initial ranges of ω, c_1 and c_2 are [0.2, 3]. To reduce the run time, the maximum number of function evaluations is 500 with 10 independent runs. The maximum number of function evaluations is 500 for these VSPSO 2011 using the parameters obtained and SPSO 2011 with 100 independent runs.

The optimized parameters were given in Table 3. The optimization statistical analysis of proposed method and SPSO 2011 with fixed parameters was given in Table 4. As can be seen from Table 3, the parameters are totally different from the fixed parameters of SPSO 2011 and there are no rules to follow as the optimization problems are totally different. As can be seen from the Table 4, the optimization performance of VSPSO 2011 is more stable and it can guarantee the worst results are not worse than the worst results of SPSO 2011 as the proposed parameter optimization method is cask theory based parameter optimization method.

Table 1. Functions used to test the effects of the LGPSO method

Sphere	$f(x)=\sum_{i=1}^{D} x_i^2$
Rastrigin	$f(x)=10D+\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i))$
Step	$f(x)=\sum_{i=1}^{D}[(x_i+0.5)^2]$, [.] is rounding function
Rosenbrock	$f(x)=\sum_{i=1}^{D}(100(x_{i+1}-x_i^{\,2}))+(x_i-1)^2)$
Ackley	$f(x)=20+e-20e^{-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}}-e^{-\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)}$
Griewank	$f(x)=\frac{1}{4000}\sum_{i=1}^{D}(x_i-100)^2-\prod_{i=1}^{D}\cos(\frac{(x_i-100)}{\sqrt{i}})+1$
Salomon	$f(x)=\cos(2\pi\sqrt{\sum_{i=1}^{D}x_i^2})+0.1\sqrt{\sum_{i=1}^{D}x_i^2}+1$
Rotated hyper-ellipsoid	$f(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i}x_j)^2$

Table 2. Functions parameters for the test problems

Functions	Dimension	Initial range
Sphere	30	±500
Rastrigin	30	±500
Step	30	±500
Rosenbrock	30	±500
Ackley	30	±500
Griewank	30	±500
Salomon	30	±500
Rotated hyper-ellipsoid	30	±500

Table 3. Optimized parameters for the test problems

Functions	Inertia weight ω, and	Acceleration coefficient c_1	Acceleration coefficient c_2
Sphere	0.5728	0.6336	0.8422
Rastrigin	0.5908	0.6726	0.9059
Step	0.6539	0.5442	0.6911
Rosenbrock	0.6392	1.2737	0.5954
Ackley	3.0000	3.0000	2.9441
Griewank	0.5901	0.9769	0.7857
Salomon	0.5424	0.3778	0.5264
Rotated hyper-ellipsoid	0.5360	0.8172	0.6147

Table 4. Comparison between standard PSO 2011 and VSPSO 2011

Problem	Method	best	Mean	Std.dev	Worst
Sphere	Standard PSO 2011	1.0482e+005	2.2438e+005	5.6333e+004	4.1521e+005
Sphere	VSPSO 2011	2.4199e+004	7.0329e+004	2.1287e+004	1.2790e+005
Rastrigin	Standard PSO 2011	1.1813e+005	2.2891e+005	4.7518e+004	3.5182e+005
Rastrigin	VSPSO 2011	3.1282e+004	7.6292e+004	2.2688e+004	1.3040e+005
Step	Standard PSO 2011	114834	2.1631e+005	4.3216e+004	345796
Step	VSPSO 2011	27158	7.2645e+004	2.2680e+004	151432
Rosenbrock	Standard PSO 2011	9.3528e+010	4.8421e+011	2.1182e+011	1.2005e+012
Rosenbrock	VSPSO 2011	4.8036e+009	3.5983e+010	2.4238e+010	1.4794e+011
Ackley	Standard PSO 2011	20	20.2424	0.1278	20.5651
Ackley	VSPSO 2011	20	20	0	20
Griewank	Standard PSO 2011	27.3732	57.4863	11.4453	90.7654
Griewank	VSPSO 2011	7.7910	16.8858	4.8584	35.0051
Salomon	Standard PSO 2011	37.5720	47.5551	4.7485	59.1266
Salomon	VSPSO 2011	17.2006	29.1404	4.0430	37.6110
Rotated hyper-ellipsoid	Standard PSO 2011	4.5205e+005	7.6140e+005	1.7838e+005	1.2665e+006
Rotated hyper-ellipsoid	VSPSO 2011	6.8559e+004	1.4094e+005	4.2328e+004	2.7582e+005

4 Conclusion

In this paper, a cask theory based parameter optimization based particle swarm optimization was proposed to find a good set of parameter of. This method can find sets of optimized parameters and using the obtained parameters can achieve better optimization performance than the standard set of parameters. No prior experience is needed for this method. The simulations showed that the proposed method can achieve good optimization performance comparing with the SPSO 2011. Moreover, the simulations show that it can make sure the worst results are not worse than the worst results of SPSO 2011 as this is cask theory based parameter optimization method. This method can also be used to find the parameters of other optimization algorithms.

Acknowledgements. This work was supported by China/South Africa Research Cooperation Programme (No. 78673 & CS06-L02), South African National Research Foundation Incentive Grant (No. 81705), SDUST Research Fund (No. 2010KYTD101) and Key scientific support program of Qingdao City (No. 11-2-3-51-nsh).

The authors thank the Matlab version codes of standard PSO 2011 compiled by Mahamed Omran and the codes are available at the particle swarm central website [8].

References

1. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference Neural Networks, Perth, Australia, pp. 1942–1948 (1995)
2. Clerc, M.: Particle Swarm Optimization. ISTE Publishing Company (2006)
3. Ju, J., Wei, S.: Endowment versus Finance: A Wooden Barrel Theory of International Trade, CEPR Discussion Papers 5109, C.E.P.R. Discussion Papers (2005)
4. Xinchao, Z.: A perturbed particle swarm algorithm for numerical optimization. Applied Soft Computing 10, 119–124 (2010)
5. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2008)
6. Roy, R., Dehuri, S., Cho, S.B.: A Novel Particle Swarm Optimization Algorithm for Multi-Objective Combinatorial Optimization Problem. International Journal of Applied Metaheuristic Computing 2(4), 41–57 (2012)
7. Chen, W., Zhang, J.: A novel set-based particle swarm optimization method for discrete optimization problem. IEEE Transactions on Evolutionary Computation 14, 278–300 (2010)
8. Kennedy, J., Clerc, M., et al.: Particle Swarm Central (2012), `http://www.particleswarm.info/Programs.html`
9. Mercer, R.E., Sampson, J.R.: Adaptive search using a reproductive metaplan. Kybernetes 7(3), 215–228 (1978)
10. Keane, A.J.: Genetic algorithm optimization in multi-peak problems: studies in convergence and robustness. Artificial Intelligence in Engineering 9(2), 75–83 (1995)
11. Pedersen, M.E.H., Chipperfield, A.J.: Simplifying particle swarm optimization. Applied Soft Computing 10(2), 618–628 (2010)
12. Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K.: A racing algorithm for configuring metaheuristics. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 11–18 (2002)