A Self-Government Particle Swarm Optimization Algorithm and Its Application in Texaco Gasification

Weitian Lin and Xingsheng Gu*
Research Institute of Automation, East China University of Science and Technology, Shanghai, 200237, China
Email: linwt@sdju.edu.cn, xsgu@ecust.edu.cn

Zhigang Lian, Yufa Xu and Bin Jiao
Shanghai DianJi University, Shanghai, 200240, China
Email: lianzg@sdju.edu.cn, xuyf@sdju.edu.cn, jiaob@sdju.edu.cn

***Abstract*—In this paper, a self-government particle swarm optimizer (SGPSO) is proposed to improve the performance of original PSO, in which particle updating depends on local best information searched at anterior runs as well as individual history best and global best at present. To evaluate the novel algorithm, some benchmark functions are employed in comparison with PSO. Experimental results show that the proposed algorithm can search more optimal solution than PSO and indicate the effectiveness of the novel algorithm to solve optimization problems. Finally, the proposed algorithm is applied in soft-sensing the Texaco furnace temperature. It is convinced that SGPSO based soft sensor is very capable of real-time assessment of the furnace temperature in the Texaco gasification process.**

***Index Terms*—particle swarm optimization, self-government, texaco gasification**

I. INTRODUCTION

Particle swarm optimization (PSO) algorithm is an important and usual member of swarm intelligence methods for solving global optimization problems. In 1995, PSO was originally developed by Eberhart and Kennedy [1,2], inspired by social behavior of bird flocking and fish schooling. Like genetic algorithm, ant colony algorithm etc., PSO is associated with artificial life and evolutionary computation, which can be easily implemented and is computationally inexpensive. In view of these advantages, PSO has been successfully applied in many fields including function optimization, fuzzy control, artificial neural network training and so on. However, for more and more complicated problems, PSO proved to be incapable in some cases. Therefore, in order to strengthen the optimization ability of PSO, researchers gradually proposed many advanced PSO algorithms. Eberhart and Shi [3] firstly concluded developments, applications and resources of PSO. Considering the significance of parameter selection in PSO, Shi and Eberhart [4] studied the effect of different parameter values on the evolutionary performance. Ref. [5] reached the effect of population structure on the performance of PSO. Some scholars [6] used cluster analysis method to investigate PSO's performance. Also, researchers brought mathematic concept into PSO. Ref. [7,8] presented PSO incorporated with function "stretching" to alleviate the local optimization problem. Parsopoulos and Vrahatis [9] made used of nonlinear simplex method to initialize PSO to expand the search space for better solution. Ref. [10] introduced Gaussian mutation scheme into PSO, which was proved to be successful in computing the better solutions than those by PSO. Previously, people only used the simplex algorithm and the advanced versions to settle problems. From another angle, it can be considered to combine other algorithms effectively. For instance, Wang and Li [11] integrated PSO and SA (simulated annealing) to improve the performance of PSO. Shi et al. [12] have presented an improved GA and a novel PSO-GA-based hybrid algorithm. Kao [13] focused on a hybrid method combining two heuristic optimization techniques, GA and PSO for the global multimodal function optimization problems. Liu et al. [14] presented an improved particle swarm optimization combined with chaos. Sun et al. [15,16] proposed a new PSO with quantum behavior (QPSO) which could improve the optimal effect. Besides, Sha and Hsu [17] proposed that the particle movement in PSO is based on the swap operator. Wen and Cao [18] presented a modified particle swami optimizer based on cloud model. Pan et al. [19,20] studied discrete PSO algorithm in detail and applied it successfully. Huang and Gu [21] proposed that binary particle swarm algorithm is integrated into cultural algorithm framework to develop a more efficient cultural binary particle swarm algorithm. Chen [22] developed refined binary particle swarm optimization, in which the

---

This work is supported by the Humanities and Social Sciences Youth Fund Project of Ministry of Education (grant No. 09yjc630151), Shanghai Universities Select and Train Outstanding Young Teachers in special funds for scientific research (grant No. sdju200903), Shanghai Commission of Science and Technology (grant No. 10JC1405800).

*Corresponding author.

© 2013 ACADEMY PUBLISHER
doi:10.4304/jsw.8.2.472-479

individual particle moves stochastically toward the position that is affected by the present velocity, the individual best performance and the best performance of the group. Hao [23] proposed a new stochastic particle swarm optimization algorithm based on cluster analysis for ensuring global convergence. A simplex particle swarm optimization derived from the Nelder-Mead simplex method [24] was proposed to optimize the high dimensionality functions. Song et al. [25] introduced the centroid of particle swarm in standard PSO model, then combining the strong chaotic motion and the simplex method. Shu and Yang [26] presents a new approach to the solution of optimal manufacturers production and delivery scheduling problem, using improved particle swarm optimization technique.

In this paper we propose SGPSO algorithm for optimization problem in the Texaco gasification. The update of each particle's position relies on not only the information from the individual historical best and the global historical optimum at current run, but also the local optima searched at the anterior runs. The effectiveness of the proposed algorithm will be verified by some typical benchmark functions. Then, SGPSO is used in estimation of Texaco furnace temperature. The rest of the paper is organized as follow: The next section introduces the principle of the original PSO algorithm. Thoughts of SGPSO algorithm are presented in Section 3. In Section 4, the verified results of test functions by using SGPSO are provided. Section 5 illustrates the application of SGPSO in the Texaco gasification process. Finally, Section 6 draws a conclusion for this paper.

II. Original Particle Swarm Optimizer

Particle Swarm Optimization (PSO) is an evolutionary computation technique. Each particle in the swarm represents a candidate solution to the solved problem. The state of the particle in the search space is always defined by its position and velocity. Then, the position and the velocity are adjusted dependently on its own and the neighborhood experience regularly.

In a D-dimensional problem space, assume that there is a swarm with m particles. The position of ith particle is denoted by a D-dimensional vector $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$ and its velocity is represented by $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$. The personal best position of particle i found so far is denoted by $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$ and the current best position of the whole swarm is $P_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$. The fitness value of each particle is evaluated by the objective function. At each generation, the velocity and position are updated according to the following equations:

$$\begin{aligned} v_{id}(k+1) &= wv_{id}(k) + c_1 r_1 (p_{id}(k) - x_{id}(k)) \\ &\quad + c_2 r_2 (p_{gd}(k) - x_{id}(k)) \end{aligned} \tag{1}$$

$$\begin{aligned} &x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \\ &(i = 1,2,\cdots,m;\ d = 1,2,\cdots,D) \end{aligned} \tag{2}$$

where k is the iterative number, w is the inertia weight. c_1, c_2 are learning factors, usually set to 2. r_1, $r_2$are two uniform random numbers distributed in the range of [0, 1]. This iterative process is repeated until a user-defined stopping criterion is reached.

III. Self-government Particle Optimization

The model for updating velocity in PSO algorithm refers to two factors, i.e. the personal best position of each particle P_i and the previous best position of the swarm P_g. However, some useful potential information may be overlooked, which influences the global searching ability. In this paper, a self-government particle swarm optimization algorithm is proposed, in which the local optimal solution searched several runs ago will be shared at the next run. The detailed SGPSO is described as below.

Let $p_a(t)$ represent one of the anterior searched local best particles before tth experiment and it can be selected from former t-1 local best particles. The updating equations of the velocity and position are in the following.

$$\begin{aligned} v_{id}(k+1) &= wv_{id}(k) + c_1 r_1 (p_{id}(k) - x_{id}(k)) \\ &\quad + c_2 r_2 (p_{gd}(k) - x_{id}(k)) \end{aligned} \tag{3}$$

$$\begin{aligned} x_{id}(k+1) &= \alpha x_{id}(k) + (1-\alpha)(p_{ad}(t) - x_{id}(k)) \\ &\quad + v_{id}(k+1) \end{aligned} \tag{4}$$

$$p_a(t) \in \{p_a(1), p_a(2), \cdots, p_a(t-1)\} \tag{5}$$

where α is a constant in [0, 1], which represents the level of sharing the information inheritances from anterior searched local optima. At 1st run, there is no inheritanced local optimal information, so α is set to 1. As shown in (5), $p_a(t)$ is one component of t-1 local best particles searched at former t-1 experiments.

The SGPSO can be transformed to random self-government particle swarm optimization algorithm (RSGPSO), in which random weight of anterior searched local optimum is introduced. Equation (4) is changed as follows:

$$\begin{aligned} x_{id}(k+1) &= \alpha x_{id}(k) + (1-\alpha) r_3 (p_{ad}(t) - x_{id}(k)) \\ &\quad + v_{id}(k+1) \end{aligned} \tag{6}$$

And r_3 could be unique random number in [0, 1] at each generation or the component of a D-dimensional random vector for particle i.

The main work conducted in the novel algorithm is to improve the position updating equation from the point of not only sharing the information of the individual history best and the global optimum, but also combining the local optima searched at the anterior runs.

IV. Numerical Simulation

A. Benchmark Functions

To illustrate the effectiveness and performance of SGPSO algorithm for optimizing problems, a set of 8 benchmark functions listed in Table 1 are adopted to be the examples in comparison with PSO algorithm.

© 2013 ACADEMY PUBLISHER

TABLE Ⅰ.
BENCHMARK FUNCTIONS

Function Number	Function Description	Feasible Solution Space
f_1	$f(x)=\sum_{i=1}^{n} x_i^2$	$x_i \in [-100,100]$
f_2	$f(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i} x_j)^2$	$x_i \in [-100,100]$
f_3	$f(x)=\max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	$x_i \in [-100,100]$
f_4	$f(x)=\sum_{i=1}^{n-1}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2)$	$x_i \in [-100,100]$
f_5	$f(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i)+20+e$	$x_i \in [-32,32]$
f_6	$f(x)=\sum_{i=1}^{n}(x_i^2-10\cdot\cos(2\pi x_i)+10)$	$x_i \in [-5.12,5.12]$
f_7	$f(x)=(x_1^2+x_2^2)^{0.25}\cdot(\sin^2(50\cdot(x_1^2+x_2^2)^{0.1})+1.0)$	$x_i \in [-10,10]$
f_8	$f(x)=10^6 x_1^2+\sum_{i=2}^{n} x_i^2$	$x_i \in [-100,100]$

B. Experimental Results and Comparison

The experimental results for each algorithm on the test functions are listed in Table 2. To evaluate the performance of the proposed SGPSO, original PSO is employed for comparison purpose and the solution quality is averaged. All experiments are repeated for 10 runs. The parameters of PSO and SGPSO algorithm are also listed in Table 2. In addition c_1, c_2 are equal to 2 and w is 0.3 for PSO and SGPSO. In Table 2, F and D denote the functions and their dimension respectively; *OS* is the optimal solution; *PS* and *EG* represent the population size and the maximum generation. α is the restraining parameter in SGPSO. The best solutions and the parameters found in SGPSO are illustrated with bold letters.

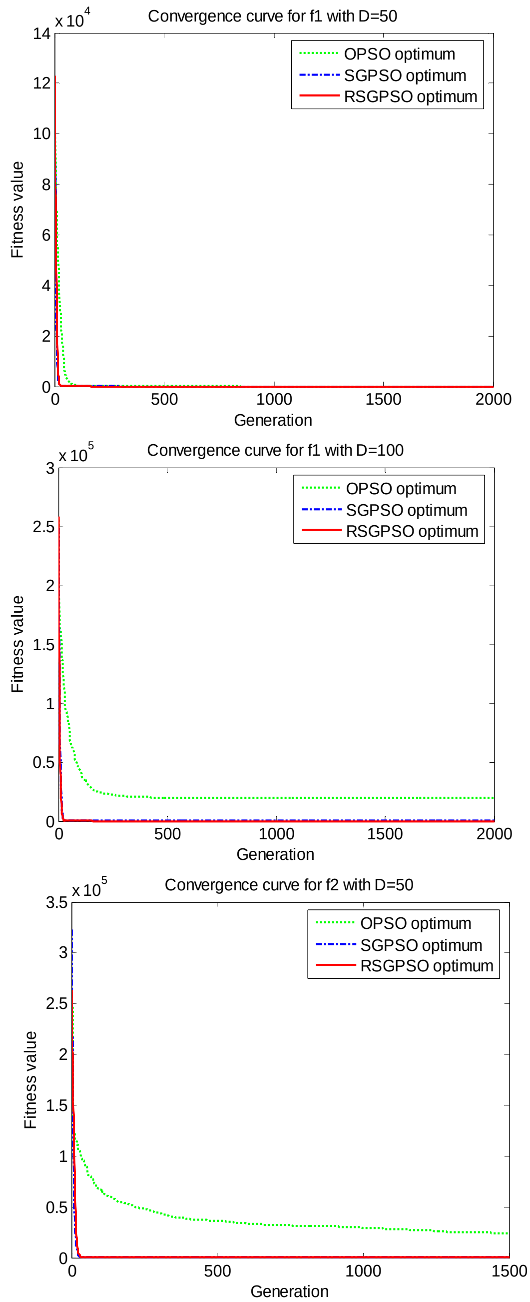
Seen from Table 2, it's obvious that SGPSO and RSGPSO find the better result marked with bold letter for each function, which indicates that the algorithms are superior to PSO. For the small scale problems such as f_3, f_6, f_7 with D=30, SGPSO or RSGPSO gets the better results than that by using PSO. Like f_1, f_2, f_3, f_4, f_5, f_8 with D=50, they are the medium scale problems and SGPSO still shows out its advantage of searching better solution. Also, for the big scale problem like f_1 with D=100, SGPSO performs more effectively than PSO. Therefore, not only the low but also the medium and high dimensional functions are optimized well by SGPSO and RSGPSO.

It is obvious that the solutions of f_1, f_2, f_3, f_4, f_8 found by SGPSO and RSGPSO are much better than that by PSO. Especially for f_1 with D=50 and D=100, the excellent characteristics of the novel method is clearly expressed with the more stable and better results under different values of α. Also, seen from f_1, f_5, f_8, the maximum and average solutions almost don't have big difference from the minimum solutions relatively, which indicates that the new algorithm is robust. Therefore, SGPSO and RSGPSO can be proved to have great advantage on optimizing f_1, f_2, f_3, f_5, f_8.

It is found that α has a great effect on the performance of SGPSO and RSGPSO. For f_1, f_2, f_3, f_8, with the increasing of α, the solution quality gets better simultaneously, and the best solutions are searched when α is 0.9. On the contrary, the best solutions of f_5, f_6 are obtained when α is equal to 0.1. Also, when α is 0.5, f_4, f_7 get the optimal results. It's concluded that α needs to be given different values in [0, 1] for different problems.

The convergence curves of SGPSO and RSGPSO comparing with original PSO for 8 instances with different dimensions are shown in Fig.1.

We can discover from Fig. 1 that the convergence speed of SGPSO algorithm is clearly faster than PSO on every test function. Especially, SGPSO algorithm is more efficacious than PSO for medium and big size function problems.



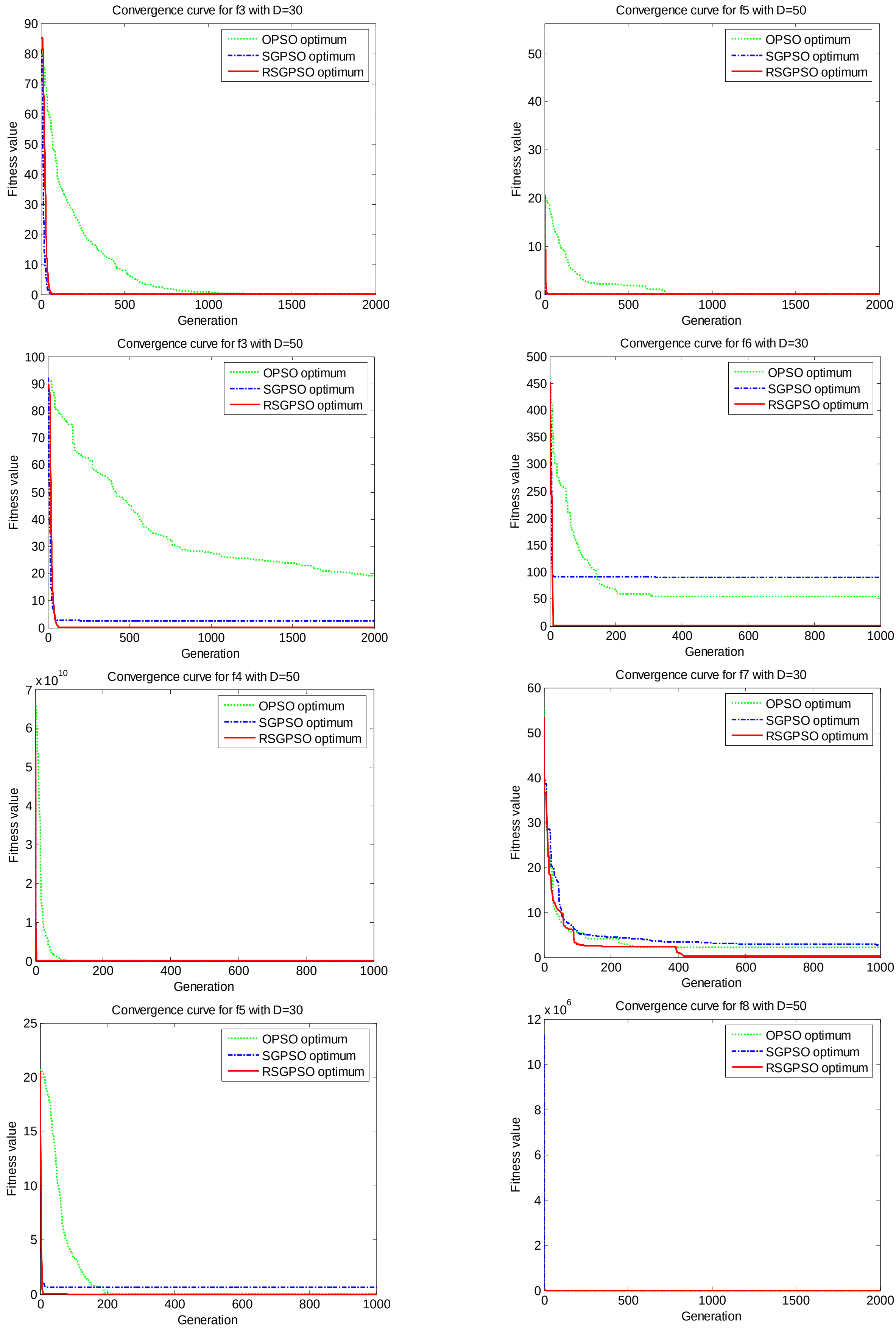


Figure 1. The convergence curves of PSO, SGPSO and RSGPSO for benchmark functions

© 2013 ACADEMY PUBLISHER

Take f_1 with D=100, f_3 with D=30,f_5 with D=50 for example. The curve line of PSO falls more slowly than that of SGPSO and RSGPSO, which shows the strong global searching ability of the novel methods. And for f_3 with D=50 and f_6, PSO is easy to stick into local minimum solutions, but SGPSO is not. In a word, the proposed algorithm improves the ability of searching global optimum and is easy to escape from local solutions. Additionally, from the perspective of optimizing functions, SGPSO algorithm is a more effective and superior algorithm comparing with original PSO.

TABLE Ⅱ.
THE COMPARISON RESULTS OF THE PSO, SGPSO AND RSGPSO ALGORITHM

F, *D*, *OS*	*PS*, *EG*	α	PSO			SGPSO			RSGPSO		
			Min	Max	Average	Min	Max	Average	Min	Max	Average
f_1, 50, 0	100, 2000	0.1	6.06E-33	1.20E-4	1.20E-5	2.03E-35	2.54E-29	3.86E-30	2.93E-24	4.89E-20	5.05E-21
		0.3	**2.70E-34**	7.86E-10	8.14E-11	8.48E-39	7.72E-34	8.74E-35	1.07E-7	3.15E-5	4.88E-6
		0.5	9.11E-33	0.5291	0.0593	1.81E-39	4.13E-32	4.42E-33	5.58E-24	5.48E-21	6.18E-22
		0.7	2.16E-32	1.69E-5	1.69E-6	3.90E-47	3.83E-31	3.86E-32	2.55E-41	4.53E-32	4.81E-33
		0.9	1.31E-32	1.48E-7	1.48E-8	**2.04E-52**	1.38E-30	1.38E-31	**6.08E-50**	7.20E-33	7.24E-34
f_1, 100, 0	100, 2000	0.1	5.96E-7	2.81E+5	2.81E+4	1.96E-8	0.0125	0.0015	1.67E-5	0.0024	4.98E-4
		0.3	5.45E-6	5.4729	0.7263	3.75E-13	6.71E-7	7.54E-8	3.51E-6	8.42E-4	1.70E-4
		0.5	1.40E-6	2.49E+3	251.0544	2.96E-11	5.00E-4	5.17E-5	2.39E-8	3.38E-5	6.66E-6
		0.7	**1.68E-9**	16.8386	1.7233	1.83E-14	3.50E-4	3.72E-5	3.50E-9	6.26E-6	7.77E-7
		0.9	3.11E-5	18.2356	1.8976	**1.86E-27**	7.48E-7	7.49E-8	**2.48E-21**	4.61E-8	4.66E-9
f_2, 50, 0	150, 1500	0.1	1.86E+3	5.06E+3	3.29E+3	0.1244	3.03E+3	526.3157	1.12E+3	2.18E+5	4.34E+4
		0.3	1.42E+3	4.31E+3	2.70E+3	0.0134	5.10E+3	571.6783	60.2672	3.78E+3	1.31E+3
		0.5	2.17E+3	6.03E+3	3.55E+3	1.20E-3	2.68E+3	302.4756	457.8606	8.17E+3	2.12E+3
		0.7	1.01E+3	7.22E+3	3.07E+3	4.51E-4	3.27E+3	352.5909	3.7155	2.56E+3	493.0762
		0.9	**824.7694**	3.14E+3	1.87E+3	**1.39E-11**	2.07E+3	214.2739	**1.00E-3**	1.14E+3	138.1958
f_3, 30, 0	100, 2000	0.1	1.5991	4.5100	3.4126	0.0426	2.7089	0.5502	**4.06E-5**	6.1791	0.9335
		0.3	1.4831	7.4511	3.0705	0.0204	1.6182	0.4488	7.46E-5	3.5437	0.5579
		0.5	**1.2358**	9.5737	4.2166	6.70E-3	3.3361	0.4315	7.51E-5	3.533	0.4553
		0.7	1.3266	8.382	3.7894	3.11E-6	2.3025	0.301	1.44E-4	8.9403	1.3115
		0.9	1.4951	8.0448	3.8716	**2.91E-10**	4.0048	0.4201	1.90E-4	5.7301	0.7794
f_3, 50, 0	100, 2000	0.1	**45.9419**	85.9772	57.6989	0.2031	60.0064	19.277	6.861	89.403	36.0781
		0.3	53.0239	88.3306	70.7442	0.4533	58.7084	14.9346	7.6309	45.1122	21.8718
		0.5	50.3266	91.1924	75.3893	0.3959	65.9432	21.7538	12.5211	60.9642	30.6562
		0.7	52.8727	90.9625	69.8687	6.90E-3	73.975	15.3151	24.3419	91.9584	56.7467
			47.1727	89.7743	68.4389	**4.89E-7**	84.5946	9.8086	**4.61E-4**	61.1601	8.6771
f_4, 50, 0	100, 1000	0.1	25.1521	30.9198	28.5107	20.1456	511.8787	88.4644	5.35E-6	22.8529	2.2946
		0.3	28.8938	85.8848	39.6046	13.0093	343.1682	66.9934	3.39E-5	141.9221	14.3166
		0.5	28.8727	986.6365	156.3852	4.5866	3.18E+3	411.0205	**4.44E-7**	27.1979	2.7223
		0.7	27.1312	3.34E+4	3.50E+3	17.6173	1.78E+3	309.2532	1.16E-5	73.1538	7.3638
		0.9	**5.1168**	33.6682	24.3234	**0.7243**	5.56E+3	763.0506	1.58E-5	16.7609	1.6958
f_5, 30, 0	100, 1000	0.1	1.51E-14	1.5017	0.3999	0.1006	1.778	0.6694	**6.21E-20**	1.51E-14	1.75E-15
		0.3	**7.99E-15**	1.6462	0.1646	**4.44E-15**	1.51E-14	6.93E-15	7.67E-20	4.00E-14	4.55E-15
		0.5	**7.99E-15**	2.0119	0.4669	1.17E-10	1.66E-4	1.85E-5	8.81E-20	1.51E-14	1.70E-15
		0.7	**7.99E-15**	1.778	0.5776	**4.44E-15**	1.15E-14	5.15E-15	1.24E-19	1.15E-14	1.33E-15
		0.9	**7.99E-15**	1.6462	0.1646	**4.44E-15**	1.51E-14	8.35E-15	1.04E-19	3.64E-14	4.11E-15
f_5, 50, 0	100, 2000	0.1	**1.51E-14**	0.0028	2.83E-4	**8.88E-16**	1.51E-14	3.73E-15	**4.44E-15**	2.22E-14	1.01E-14
		0.3	**1.15E-14**	1.1551	0.1155	4.44E-15	1.51E-14	5.51E-15	7.18E-7	1.98E-5	5.73E-6
		0.5	**1.15E-14**	1.471	0.2626	1.74E-5	2.3164	0.3426	**4.44E-15**	1.51E-14	6.57E-15
		0.7	**1.51E-14**	1.8748	0.1875	2.86E-9	1.2697	0.1337	**4.44E-15**	1.51E-14	6.57E-15
		0.9	**1.51E-14**	1.6462	0.4492	7.99E-15	1.87E-14	9.06E-15	1.51E-14	3.55E-6	3.70E-7
f_6, 30, 0	100, 1000	0.1	**28.8538**	85.5662	54.0331	0.0498	305.4811	41.9094	**1.21E-5**	39.7983	4.0356
		0.3	31.8387	65.6671	50.8423	**2.57E-4**	258.6166	72.7993	1.88E-5	32.8336	3.3447
		0.5	39.7983	82.5813	54.0262	0.2702	290.7098	159.8001	1.99E-5	43.7781	4.4381
		0.7	32.8336	77.6066	50.6433	54.7226	228.8895	145.3184	7.55E-5	63.6773	6.452
		0.9	46.763	70.6419	55.8171	50.8843	103.6366	81.9967	2.12E-5	64.6722	6.5157
f_7, 30, 0	100, 1000	0.1	0.0505	1.6041	0.6014	**0.0456**	48.1482	35.7637	1.50E-4	2.7782	0.4688
		0.3	4.80E-3	0.7156	0.2846	1.8071	51.0527	36.6061	2.43E-4	5.0573	0.8355
		0.5	0.0394	1.6206	0.3062	0.0848	52.2338	38.3453	**5.91E-5**	1.9103	0.2566
		0.7	**4.40E-3**	1.317	0.5207	0.194	45.2783	34.1372	1.74E-4	5.313	0.6872
		0.9	0.0696	1.7102	0.7458	0.79	38.0195	25.9668	1.05E-4	4.1093	0.6032
f_8, 50, 0	100, 2000	0.1	1.06E-33	1.62E-05	1.65E-06	2.21E-20	4.37E-16	4.90E-17	6.50E-34	1.37E-31	2.73E-32
		0.3	2.58E-32	2.07E-05	2.07E-06	9.88E-35	3.58E-31	5.58E-32	1.63E-24	6.46E-22	1.08E-22
		0.5	**2.89E-34**	0.7226	7.26E-02	1.17E-41	9.94E-32	1.02E-32	3.40E-17	1.19E-10	1.24E-11
		0.7	4.62E-31	0.0011	1.12E-04	2.75E-41	3.07E-22	3.07E-23	2.13E-31	2.60E-21	2.61E-22
		0.9	4.62E-33	30.4086	3.2582	**3.96E-51**	4.17E-29	4.18E-30	**5.08E-47**	7.91E-30	7.99E-31

E-n denotes $\times 10^n$.

© 2013 ACADEMY PUBLISHER

V. Soft Sensor of Texaco Furnace Temperature Based on SGPSO

A. *Texaco Gasification Process*

Texaco gasification is one of the most developing coal gasification technologies currently. It adopts the way of wetting ball mill with high security and reliability. The gasification provides the syngas as the raw material for the following systems. Thus, its stability would guarantee the reliable operation of these systems. The great progress of Texaco gasifier mainly relies on the operator's experience. That is because it is very difficult to measure the gasifier furnace temperature. In the Texaco gasification process, the furnace temperature is a critical control parameter, which is closely related to the components of the syngas and the gasification efficiency thus affecting the product quality and yield.

Currently, two methods are used for the furnace temperature measurement: (1) thermocouple measurement; (2) indirect estimation through methane content. The former one uses the thermocouple to directly measure the furnace temperature. But its life only lasts for about a week, which leads to the invalidation of indicating the temperature in a long time. On the other hand, the estimated temperature through methane content is time-delay and the accuracy is unsatisfactory. Therefore, a more precise measurement of Texaco furnace temperature is urgently needed. In this paper, two soft-sensing models that combine the proposed SGPSO and RSGPSO with BP neural network (SGPSO-NN and RSGPSO-NN) are established to realize the real-time measurement of the temperature so as to avoid the problems caused by the two former methods.

B. *Soft Sensor Modeling of Gasifier Furnace Temperature*

Through the study on the Texaco gasification process, 14 operational parameters that correlate with the furnace temperature are selected as the auxiliary variables for the soft sensors. The tag number in DCS and the descriptions of these variables are shown in Table 3.

Among them, the flow of slurry to gasifer is the middle value of three measurements of FT-205/206/207; the slurry temperature is set to any value in the measurements of TE-203/253; the oxygen pressure is the arbitrary value in the PT-203/253 measurements, and the oxygen temperature is the any value in the TE-205/255 measurements.

In the 14 auxiliary variables, X_1-X_{13} are sampled through DCS system. The methane content (X_{14}) is calculated and recorded by the analyzer 3 times per day. As the aforementioned, the lifetime of the thermocouple is very short, so the historical data of the furnace temperature are obtained before the thermocouples are broken. Finally, 373 groups of modeling data are divided into two parts. The 273 groups are used as the training data to learn the parameters of the soft-sensing models and the remaining 100 groups as testing data to evaluate the generalization capability. The normalized historical data are listed in Table 4.

TABLE III.
INPUT AND OUTPUT VARIABLES FOR GASIFIER TEMPERATURE SOFT SENSOR

Variable Name	Tag No.	Variable Description	Unit
X_1	FT-205/206/207	Flow of slurry to gasifer	m^3/h
X_2	PT-202	Pressure of slurry to gasifer	MPa
X_3	TE-203/253	Temperature of slurry to gasifer	℃
X_4	FIC-208	Flow of oxygen to gasifer	m^3/h
X_5	PT-203/253	Pressure of oxygen to gasifer	MPa
X_6	TE-205/255	Temperature of oxygen to gasifer	℃
X_7	FT-217	Flow of quenching water to gasifer	m^3/h
X_8	TT-236	Temperature of quenching water to gasifer	℃
X_9	FT-212	Flow of water from gasifer quench chamber	m^3/h
X_{10}	TE-265	Temperature of water from gasifer quench chamber	℃
X_{11}	FIQ-213	Flow of syngas from gasifer	m^3/h
X_{12}	PT-206	Pressure of syngas from gasifer	MPa
X_{13}	TE-213	Temperature of syngas from gasifer	℃
X_{14}	AT-202	Methane content	ppm
Y	——	Furnace temperature	℃

TABLE IV.
NORMALIZED DATA OF SOFT SENSOR VARIABLES

No.	X_1	X_2	X_3	X_4	…	X_{12}	X_{13}	X_{14}	Y
1	0.593	0.954	0.948	0.836	…	0.948	0.765	-0.419	-0.424
2	0.576	0.947	0.953	0.875	…	0.956	0.822	-0.457	-0.400
3	0.561	0.955	0.956	0.875	…	0.959	0.757	-0.492	-0.368
4	0.518	0.943	0.959	0.904	…	0.968	0.763	-0.478	-0.397
5	0.512	0.946	0.948	0.914	…	0.965	0.800	-0.488	-0.421
6	0.502	0.938	0.957	0.925	…	0.974	0.830	-0.506	-0.412
7	0.483	0.665	0.718	0.936	…	0.814	0.677	-0.302	-0.706
8	0.493	0.695	0.720	0.964	…	0.831	0.711	-0.305	-0.775
9	0.494	0.688	0.707	0.927	…	0.807	0.661	-0.314	-0.957
10	0.440	0.617	0.646	0.938	…	0.844	0.604	-0.319	-0.885
…	…	…	…	…	…	…	…	…	…
364	0.659	0.660	0.655	0.850	…	0.919	0.756	-0.571	0.321
365	0.511	0.598	0.596	0.837	…	0.909	0.683	-0.593	0.157
366	0.427	0.574	0.575	0.816	…	0.917	0.670	-0.683	0.230
367	0.257	0.477	0.483	0.781	…	0.963	0.604	-0.922	0.427
368	0.336	0.482	0.484	0.772	…	0.953	0.648	-0.919	0.593
369	0.558	0.577	0.578	0.745	…	0.894	0.730	-0.595	0.096
370	-0.263	-0.222	-0.208	0.218	…	0.682	0.024	-0.773	0.452
371	-0.847	-0.810	-0.805	-0.071	…	0.641	-0.604	-0.697	0.633
372	-0.704	-0.698	-0.685	-0.103	…	0.570	-0.557	-0.710	-0.175
373	-0.738	-0.708	-0.697	-0.098	…	0.553	-0.575	-0.736	-0.564

© 2013 ACADEMY PUBLISHER

The SGPSO-NN and RSGPSO-NN models adopt 7-14-1 BP network structure. Both of the models have the same parameter settings as follows: the maximum generation is 300; the population size is 80; the inertia weight is 0.3; c_1, c_2 are 2. Because the restraining parameter α is important for optimization performance, 5 different values of α are selected in [0, 1] and 10 independent simulation experiments are executed for each value, respectively. The experimental results are shown in Table 5. Max, Min, Average, MSE respectively represent the maximum, minimum, average error and mean square error. It can be seen from Table 5 that the optimization performance of SGPSO is much better than RSGPSO and SGPSO can obtain best optimization results when α=0.3.

TABLE V.
THE RESULTS OF SGPSO-NN AND RSGPSO-NN WITH DIFFERENT α

α		0.1	0.3	0.5	0.7	0.9
SGPSO	Max	12.319	6.153	6.868	6.602	9.863
	Min	7.204	6.009	6.436	6.080	6.385
	Average	7.742	6.064	6.648	6.289	8.823
	MSE	1.608	0.045	0.146	0.169	1.032
RSGPSO	Max	13.085	14.153	13.058	12.806	11.686
	Min	7.327	6.652	7.850	6.299	7.641
	Average	10.720	11.335	10.920	9.980	10.323
	MSE	1.853	2.259	1.744	2.317	1.132

Fig. 2 shows the convergence curves of the SGPSO and RSGPSO algorithms for optimizing the soft sensor parameters. It can be indicated that SGPSO performs better than RSGPSO algorithm both in convergence rate and accuracy.

Fig. 3 and Fig. 4 illustrate the comparisons of training and testing results of SGPSO-NN models with the actual values. It indicates that the SGPSO-NN model outputs have good agreement with training data and provide satisfactory generalization capability for the distinct testing data.

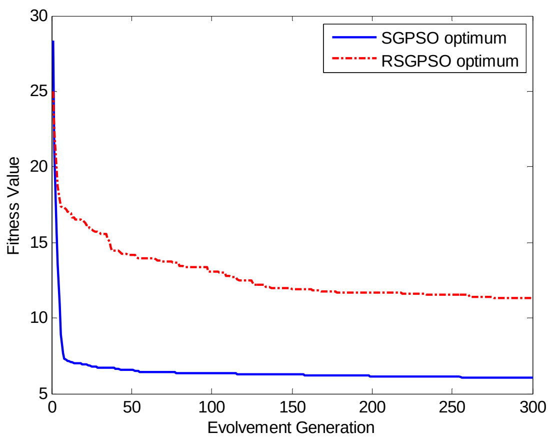


Figure 2. The convergence curve of SGPSO and RSGPSO for soft-sensing

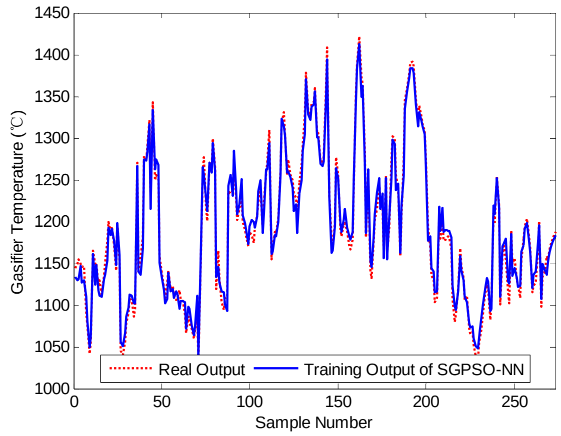


Figure 3. Comparison between real output and SGPSO-NN output for training data

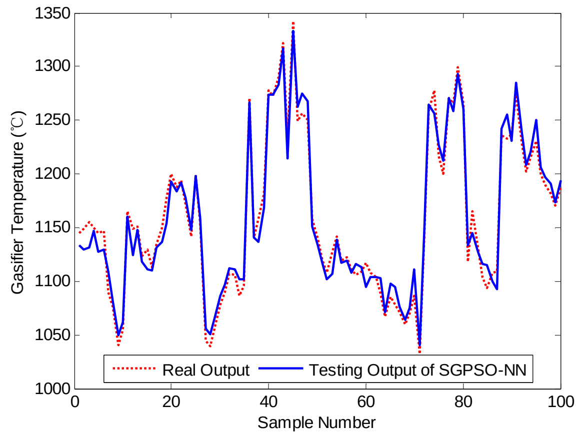


Figure 4. Comparison between real output and SGPSO-NN output for testing data

VI. CONCLUSION

In this paper, we proposed a novel self-government particle swarm optimization algorithm. The feature of the proposed method is that the updating of each particle is dependent on not only the individual historical best position and the global best position of the swarm, but also the local best optima searched at the anterior experiments. The performance of SGPSO was evaluated by using eight benchmark functions. The experimental results indicated the great effectiveness of the proposed algorithm and that SGPSO is obviously better than the original PSO. Because of global optimization capability, SGPSO-based neural network is applied to soft-sensing the gasifier temperature in the Texaco gasification process. The results imply that the performance of SGPSO meets the real-world requirements.

In the future works, two directions can be considered to make some advancement in this novel algorithm. On the one hand, the more appropriate setting of α would be found to improve the performance of SGPSO. On the other hand, the proposed method can be used to solve some discrete combinatorial optimization problems such as FSSP, JSSP besides the continuous optimization problem mentioned in this paper.

© 2013 ACADEMY PUBLISHER

ACKNOWLEDGEMENT

We are very grateful to the editors and anonymous reviewers for their valuable comments and suggestions to help improve our paper. This work is supported by the Humanities and Social Sciences Youth Fund Project of Ministry of Education (grant No. 09yjc630151), Shanghai Universities Select and Train Outstanding Young Teachers in special funds for scientific research (grant No. sdju200903), Shanghai Commission of Science and Technology (grant No. 10JC1405800) and Key Discipline of Shanghai Municipal Education Commission(J51901).

REFERENCES

[1] R. C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39–43.
[2] Y. Shi, R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, Anchorage, AK, 1998a, pp. 69–73.
[3] R. C. Eberhart, Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, South Korea, 2001, pp. 81–86.
[4] Y. Shi, R. C. Eberhart, "Parameter selection in particle swarm optimization," *Lect. Notes. Comput. Sci*, vol. 1447, pp. 591–600, 1998.
[5] J. Kennedy, R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the 2002 Congress on Evolutionary Computation*, Honolulu, HI, USA, 2002, pp. 1671–1676.
[6] J. Kennedy, "Stereotyping: improving particle swarm performance with cluster analysis," in *Proceedings of the 2000 Congress on Evolutionary Computation*, La Jolla, CA, USA, 2000, pp. 1507–1512.
[7] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, M. N. Vrahatis, "Improving the particle swarm optimizer by function "stretching"," in *Advances in Convex Analysis and Global Optimization*, N. Hadjisavvas, P. M. Pardalos, Eds. Kluwer Academic Publishers, 2001a, pp. 445 – 457.
[8] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, M. N. Vrahatis, "Stretching technique for obtaining global minimizers through particle swarm optimization," in *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, USA, 2001b, pp. 22 – 29.
[9] K. E. Parsopoulos, M. N. Vrahatis, "Initializing the particle swarm optimizer using the nonlinear simplex method," in *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, A. Grmela, N. E. Mastorakis, Eds. WSEAS Press, 2002, pp. 216 – 221.
[10] N. Higashi, H. Iba, "Particle swarm optimization with Gaussian mutation," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, Indianapolis, Indiana, USA, 2003, pp. 72 – 79.
[11] X. H. Wang, J. J. Li, "Hybrid particle swarm optimization with simulated annealing," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, 2004, pp. 2402 – 2405.
[12] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, L. M. Wang, "An improved GA and a novel PSO-GA-based hybrid algorithm," *Inform. Process. Lett*., vol. 93, pp. 255–261, 2005.
[13] Y. Kao, E. Zahara, "A hybrid genetic algorithm and particle swarm optimization for multimodal functions," *Appl. Soft. Comput*., vol. 8, pp. 849 – 857, 2008.
[14] B. Liu, L. Wang, Y. H. Jin, F. Tang, D. X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos. Solitons. Fractals*, vol. 25, 1261 – 1271, 2005.
[15] J. Sun, B. Feng, W. Xu, "Particle swarm optimization with particles having quantum behavior," in *2004 Congress on Evolutionary Computation*, 2004a, pp. 325–331.
[16] J. Sun, W. Xu, B. Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *2004 IEEE Conference on Cybernetics and Intelligent Systems*, 2004b, pp. 111 – 116.
[17] D. Y. Sha, C. Y. Hsu, "A new particle swarm optimization for the open shop scheduling problem," *Comput. Oper. Res*., vol. 35, 3243 – 3261, 2008.
[18] J. Wen, B. Cao, "A modified particle swami optimizer based on cloud model," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2008, pp. 1238 – 1241.
[19] Q. K. Pan, M. F. Tasgetiren, Y. C. Liang, "A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem," *Comput. Oper. Res*., vol. 35, 2807 – 2839, 2008a.
[20] Q. K. Pan, L. Wang, M. F. Tasgetiren, B. H. Zhao, "A hybrid discrete particle swarm optimization algorithm for the no-wait flow shop scheduling problem with makespan criterion," *Int. J. Adv. Manuf. Technol.*, vol. 38, 337 – 347, 2008b.
[21] Huang, H., Gu, X., 2009. Cultural binary particle swarm optimization algorithm and its application in fault diagnosis. J. Donghua. Univ. 26, 474–481.
[22] P. H. Chen, C. C. Kuo, F. H. Chen, C. C. Chen, "Refined binary particle swarm optimization and application in power system," *WSEAS Trans. Sys*., vol. 8, l69 – 178, 2009.
[23] W. Hao, J. Zeng, "Stochastic particle swarm optimization algorithm based on cluster analysis," *Comput. Eng. Appl*., vol. 46, pp. 40–44, 2010. (in Chinese)
[24] H. Xiao, G. Tan, "A novel particle swarm optimizer without velocity: Simplex-PSO," *J. Cent. South. Univ. Technol*., vol. 17, pp. 349–356, 2010.
[25] S. Song, Y. Gan, L. Kong, J. Chen, "A novel PSO algorithm based on local chaos & simplex search strategy and its application," *J. Software.* Vol. 6, pp. 604-611, 2011.
[26] L. Shu, L. Yang, "A modified PSO to optimize manufacturers production and delivery," *J. Software.* Vol. 7, pp. 2325-2332, 2012.

Weitian Lin was born in Shanghai on December 15, 1962. He received his bachelor's degree in Major of Industrial Automation, Shanghai University of Engineering Science, in 1983, and Master degree in Control Engineering, East China University of Science and Technology, in 2004. He is currently a Ph.D. candidate in Key Laboratory of Advanced Control and Optimization for Chemical Processes, East China University of Science and Technology.

He has been an associate professor of Electrical Engineering, Shanghai DianJi University since 2004. His research interests include intelligent modeling and optimization methods and the applications in complex industries.

© 2013 ACADEMY PUBLISHER