Solving Parameter Identification Problem by Hybrid Particle Swarm Optimization

Erwie Zahara, An Liu

***Abstract*—Ordinary differential equations have been a useful tool for describing the behavior of wide variety of dynamic physical systems. In this study, a method for solving parameter identification problem for ordinary second order differential equations using hybrid Nelder-Mead simplex search and particle swarm optimization (NM-PSO) approach is presented. Experiments using two case problems are presented and compared with the best known solutions reported in the literature. The comparison results demonstrate that NM-PSO produced better estimated results with respect to previous findings from particle swarm optimization and genetic algorithm.**

***Index Terms*—parameter identification, Nelder-Mead simplex search, particle swarm optimization, ordinary differential equations.**

I. Introduction

Parameter identification problem is a problem to estimate the unknown parameters of the mathematical models based on a system of ordinary differential equations by using experimental data obtained from well-defined standard conditions. Traditional optimization methods such as Nelder-Mead [1] or Gauss-Newton [2] can achieve reasonably good solutions for parameter identification of smaller sizes. But they are insufficiently robust for complex problems involving huge search space and they are lack of ability to overcome the local optimum points and reach the global optimum.

To overcome these local optimum points, the paper use hybrid Nelder-Mead simplex search and particle swarm optimization (denoted as NM-PSO) for parameter identification problem. The advantages of NM-PSO method had been proved by Fan and Zahara [3] that it is a promising and viable tool for solving unconstrained optimization problems. Finally the computational results will be compared with particle swarm optimization and genetic algorithm.

II. Parameter identification problem

Let us assume the system is described by second order differential equations of the form

Manuscript received December 20, 2009.

Erwie Zahara is with the Marketing and Logistics Management Department, St John's University, Tamsui, Taiwan 251, Republic of China (phone: 886-2-28013131 Ext. 6875; e-mail: erwi@mail.sju.edu.tw).

An Liu is with the Computer Science and Information Engineering Department, St John's University, Tamsui, Taiwan 251, Republic of China (e-mail: liuan@mail.sju.edu.tw).

$$\frac{d^2 y}{dt^2} = f\left(t, y, \frac{dy}{dt}, p\right) \tag{1}$$

where p is the vector of n unknown real parameters such as $p_1,\ p_2, p_3, \dots p_n$. In addition, experimental data set are given from $(t_i, y_i),\ i = 1, \cdots m$ where t_i represents the independent variable and y_i is the measured value of the corresponding dependent variable. Usually $n << m$, the problem we consider here is to estimate the optimum parameter vector p^* as accurate as possible using the given experimental data, and this is a minimization problem which can be defined by

$$E(p^*) = \min \sum_{i=1}^{m} \left[y(t_i; p) - y_i \right]^2 \quad p \in R^n \tag{2}$$

where $y(t_i; p)$ and y_i are the numeric solution of the mathematical model and experimental data point for the i-th data point, respectively.

III. Hybrid NM-PSO method

The population size of this hybrid NM-PSO approach is set at $3N+1$ when solving an *N*-dimensional problem. The initial population is created in two steps: using a predetermined starting point, *N* particles are spawned with a positive step size of 1.0 in each coordinate direction, and the other 2*N* particles are randomly generated. A total of $3N+1$ particles are sorted by the fitness, and the top $N+1$ particles are then fed into the simplex search method to improve the $(N+1)^{\text{th}}$ particle. The other $2N$ particles are adjusted by the PSO method by taking into account the positions of the $N+1$ best particles. This procedure for adjusting the 2*N* particles involves selection of the global best particle, selection of the neighborhood best particles, and finally velocity updates. The global best particle of the population is determined according to the sorted fitness values. The neighborhood best particles are selected by first evenly dividing 2*N* particles into *N* neighborhoods and denoting the particle with the better fitness value in each neighborhood as the neighborhood best particle. By equations (3) and (4), a velocity update for each of the 2*N* particles is then carried out. The $3N+1$ particles are sorted in preparation for repeating the entire run. The process terminates when a certain convergence criterion is satisfied. Figure 1 summarizes the algorithm of NM-PSO. For further details of NM, PSO and the hybrid NM-PSO, see Nelder and Mead [1], Kennedy and Eberhart [4], and Fan and Zahara [3].

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

1. **Initialization**. Generate a population of size $3N+1$.

Repeat

2. **Evaluation & Ranking**. Evaluate the fitness of each particle. Rank them based on the fitness.
3. **Simplex Method**. Apply NM operator to the top $N+1$ particles and replace the $(N+1)^{th}$ particle with the update.
4. **PSO Method**. Apply PSO operator for updating $2N$ particles with worst fitness.
   Selection. From the population select the global best particle and the neigborhood best particles.
   Velocity Update. Apply velocity update to the 2N particles with worst fitness according equations

$$V_{id}^{New}(t+1) = w \times V_{id}^{old}(t) + c_1 \times rand() \times (p_{id}(t) - x_{id}^{old}(t)) + c_2 \times rand() \times (p_{gd}(t) - x_{id}^{old}(t)) \quad (3)$$

$$x_{id}^{New}(t+1) = x_{id}^{old}(t) + V_{id}^{New}(t+1) \quad (4)$$

Until a termination condition is met.

Figure 1. The hybrid NM-PSO algorithm

IV. Numerical Simulation

In this study, the proposed NM-PSO has been applied to solve two well-known problems [5]. The algorithm was coded in Matlab 7.0 and the simulations were run on a Pentium IV 3.0GHz with 1GB memory capacity.

Example 1. Mathematical model of enzyme effusion problem [5] can be expressed as:

$$y_1^{'} = p_1(27.8 - y_1) + \frac{p_4}{2.6}(y_2 - y_1) + \frac{4991}{t\sqrt{2\pi}} \exp\left(-0.5\left(\frac{\ln(t) - p_2}{p_3}\right)^2\right), \quad (5)$$

$$y_2^{'} = \frac{p_4}{2.7}(y_1 - y_2).$$

The NM-PSO will estimate four unknown parameter values p_1, p_2, p_3, p_4 in the model using experimental data given in Table 1 and equation (5) in addition to the initial condition of y_1, y_2.

Table 1. Data for enzyme effusion problem

t	y_1	t	y_1	t	y_1	t	y_1
0.1	27.8	21.3	331.9	42.4	62.3	81.1	23.5
2.5	20.0	22.9	243.5	44.4	58.7	91.1	24.8
3.8	23.5	24.9	212.0	47.9	41.9	101.9	26.1
7.0	63.6	26.8	164.1	53.1	40.2	115.4	33.3
10.9	267.5	30.1	112.7	59.0	31.3	138.7	17.8
15.0	427.8	34.1	88.1	65.1	30.0	163.2	16.8
18.2	339.7	37.8	76.2	73.1	30.6	186.7	16.8

Table 2 shows the results obtained from NM-PSO terminate on 149 iterations. The results are also compared with genetic algorithm (GA) [5] and real-coded genetic algorithm (RGA) [6] and particle swarm optimization (PSO) [7]. Table 2 shows NM-PSO have better estimated results with smaller SSE than GA, RGA or PSO and with less number of iterations. Figure 1 shows the estimated data closely follows the measured data with the sum of square SSE is 3968.9 (number of iteration is 149).

Table 2. Results obtained using PSO

Method	p_1	p_2	p_3	p_4	$y_1(0.1)$	$y_2(0.1)$	Iteration	SSE
GA	0.319	2.701	0.3989	0.078	21.00	38.75	100	5229
GA	0.305	2.698	0.400	0.116	22.02	39.44	200	4547
GA	0.284	2.671	0.392	0.161	23.99	40.14	500	4068
RGA	0.2454	2.6092	0.3326	0.3217	22.005	38.608	100	4431.45
RGA	0.2561	2.6269	0.3449	0.2696	22.043	38.403	200	4193.92
RGA	0.2619	2.6336	0.3524	0.2575	21.986	38.704	300	4136.73
PSO	0.2667	2.6437	0.3636	0.2282	28.5443	0.2339	100	3982.0
PSO	0.2668	2.6440	0.3635	0.2280	28.5443	0.2339	200	3981.9
PSO	0.2713	2.6513	0.3690	0.2079	28.5443	0.2339	300	3971.2
NM-PSO	0.2711	2.6504	0.3686	0.2098	28.5443	0.2339	149	3968.9

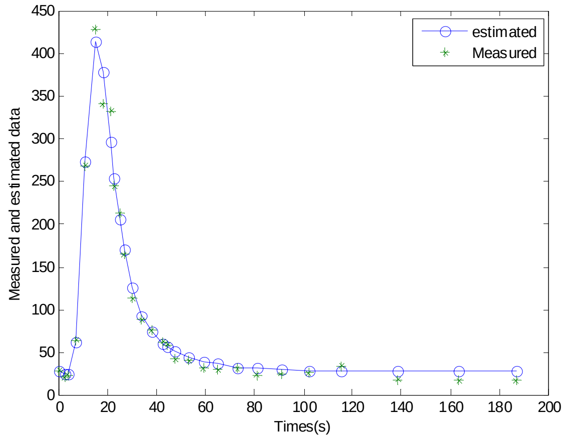

Figure 1. Variations of estimated and measured data with time for example 1

Example 2. According to the given data in Table 3, one has to estimate the parameter values p_1, p_2, p_3, p_4 of the function below

$$y(t, p) = p_1 \exp(p_3 t) + p_2 \exp(p_4 t), \qquad y^{''} - (p_3 + p_4)y^{'} + p_3 p_4 = 0 \quad (6)$$

Table 4 shows comparison results of GA, RGA, PSO and NM-PSO. As can be expected, NM-PSO estimated the same results but need less number of iterations than the other methods and Figure 2 shows the estimated parameters are very close to the measured data in the given range.

Table 3. Data for the second case

y	64.0	66.0	69.5	74.0	80.8	91.0	103.5
t	-1	-2/3	-1/3	0	1/3	2/3	1

V. Conclusion

In this paper, NM-PSO is proposed for solving parameter identification problems. Practical application of NM-PSO for two well-known problems leads us to allege that NM-PSO is indeed more accurate, reliable and efficient at locating global optima than the other alternative. In the future, we will apply NM-PSO to various problems found in the real world.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 4. Results for the second case

Method	p_1	p_2	p_3	p_4	Iteration	SSE
GA	43.2233	30.8774	0.6170	-0.2812	100	0.4396
GA	40.8112	33.3234	0.6400	-0.2464	200	0.3859
GA	37.7414	36.3533	0.6753	-0.2123	500	0.3347
RGA	42.5032	31.5469	0.6265	-0.2749	100	0.3204
RGA	41.4371	32.6713	0.6346	-0.2563	200	0.2827
PSO	30.0761	44.0683	0.7680	-0.1279	100	0.2851
PSO	30.7589	43.3814	0.7587	-0.1348	200	0.2847
NM-PSO	30.5386	43.6025	0.7617	-0.1326	60	0.2847

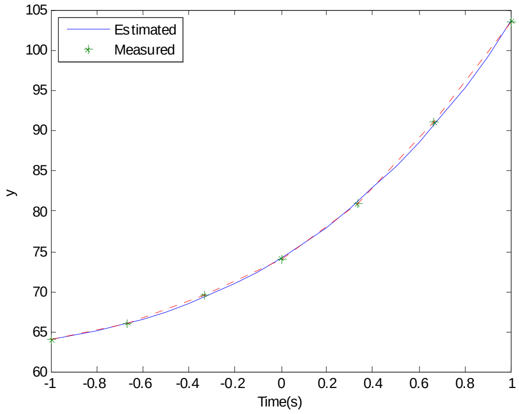

Figure 2. Variations of estimated and measured data with time for example 2 (ite=60)

REFERENCES

[1] J. E. Dennis Jr. and V. Torcon, "Direct search methods on parallel machines", *SIAM Journal of Optimization*, 1991, 1, pp. 448-474.

[2] N. Yildirim, F. Akcay, H. Okur and D. Yildirim, "Parameter estimation of nonlinear models in biochemistry: A comparative study on optimization methods", *Applied Mathematics and Computation*, 2003, 140, pp.29-36.

[3] S. K. S. Fan and E. Zahara, "A hybrid simplex search and particle swarm optimization for unconstrained optimization", *European Journal of Operational Research*, 2006, 181, pp.527-548.

[4] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39-43.

[5] E. K. Nyarko and R. Scitovski, "Solving the parameter identification problem of mathematical model using genetic algorithm", *Applied Mathematics and Computation*, 2004, 153, pp.651-658.

[6] M. A. Khalik, M. Sherif, S. Saraya and F. Areed, "Parameter identification problem: real-coded GA approach", *Applied Mathematics and Computation*, 2007, 187, pp. 1495-1501.

[7] A. Liu, E. Zahara, "Parameter identification problem using particle swarm optimization" The 5th International Conference on Natural Computation, 2009, August 14-16, Tianjin, China.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)