Expert Systems with Applications 39 (2012) 4390–4401

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Comparative performance analysis of various binary coded PSO algorithms in multivariable PID controller design

Muhammad Ilyas Menhas *, Ling Wang, Minrui Fei, Hui Pan

School of Mechatronics Engineering and Automation, Shanghai University, Shanghai, China

ARTICLE INFO

Keywords:
PID control
Swarm intelligence
Binary PSO
Particle swarm optimization
PID tuning

ABSTRACT

In this paper, comparative performance analysis of various binary coded PSO algorithms on optimal PI and PID controller design for multiple inputs multiple outputs (MIMO) process is stated. Four algorithms such as modified particle swarm optimization (MPSO), discrete binary PSO (DBPSO), modified discrete binary PSO (MBPSO) and probability based binary PSO (PBPSO) are independently realized using MATLAB. The MIMO process of binary distillation column plant, described by Wood and Berry, with and without a decoupler having two inputs and two outputs is considered. Simulations are carried out to minimize two objective functions, that is, time integral of absolute error (ITAE) and integral of absolute error (IAE) with single stopping criterion for each algorithm called maximum number of fitness evaluations. The simulation experiments are repeated 20 times with each algorithm in each case. The performance measures for comparison of various algorithms such as mean fitness, variance of fitness, and best fitness are computed. The transient performance indicators and computation time are also recorded. The inferences are made based on analysis of statistical data obtained from 20 trials of each algorithm and after having comparison with some recently reported results about same MIMO controller design employing real coded genetic algorithm (RGA) with SBX and multi-crossover approaches, covariance matrix adaptation evolution strategy (CMAES), differential evolution (DE), modified continuous PSO (MPSO) and biggest log modulus tuning (BLT). On the basis of simulation results PBPSO is identified as a comparatively better method in terms of its simplicity, consistency, search and computational efficiency.

1. Introduction

Despite numerous advancements in process control methodologies, Proportional–Integral–Derivative (PID) control is still the most efficient and widely used feedback control strategy. This is due to its simplicity and satisfactory control performance. PID controller was introduced in 1910 and its use and popularity had grown particularly after the Ziegler–Nichols empirical tuning rules in 1942 (Åström & Hägglund, 2001; Ziegler & Nichols, 1942). The development in artificial intelligence and digital technology have resulted in many intelligent control schemes such as fuzzy logic control (Goshal, 2004; Lee, 1990), neural network control (Fukuda & Shibata, 1992) and adaptive control (Astrom & Wittenmark, 1995; Zuo, 1995). But no other technique could replace PID algorithm and more than 90% of industrial controllers are still based on PID control (Ang, Chang, & Li, 2005). The wide use of PID control has sustained research on finding the key methodology for PID tuning to obtain best possible performance out of the PID control (Marsh, 1998).

The optimally combined three terms functioning of PID controller can provide treatment for both the transient and steady state responses. In fact, optimal control performance can only be achieved after identifying the finest set of three gains, that is, proportional gain (K_p), integral gain (K_i) and derivative gain (K_d). Many approaches have been reported in literature for tuning parameters of PID controller. The conventional PID tuning techniques include Z–N, Cohen Coon, and relay feedback methods (Cohen & Coon, 1953; Ziegler & Nichols, 1942). The modern techniques are based on artificial intelligence techniques such as neural network, fuzzy logic and evolutionary computation; these are the most recent techniques (Astrom & Hagglund, 1995).

Recently, many attempts have been made by several researchers to tune the PID controller parameters using various EAs, such as genetic algorithm (GA), covariance matrix adaptation evolution strategy (CMAES), particle swarm optimization (PSO), differential evolution (DE), tribes algorithm (TA), ant colony optimization (ACO), and discrete binary particle swarm optimization (DBPSO) (Bingul, 2004; Chang, 2007, 2009; Chen, Cheng, & Lee, 1995; Coelho & Bernert, 2009; Duan, Wang, & Yu, 2006; Gaing, 2004; Jan, Tseng, & Liu, 2008; Kim, Maruta, & Sugie, 2008; Menhas, Wang, Fei, & Ma, 2011; Mukherjee & Goshal, 2007; Wang, Zhang, & Wang, 2006; Willjuice & Baskar, 2009, 2010; Zhang, Zhuang, Du, & Wang, 2009; Zuo, 1995) for both the single and multivariable processes.

* Corresponding author. Tel.: +86 15821107474.
E-mail address: ilyasminhas75@yahoo.com (M.I. Menhas).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.09.152

AI-based evolutionary computational techniques can determine the most optimal sets of controller gains based on a given objective function in an iterative manner from thousands of possible alternate solutions that best fit the designer's requirements. But the performance of different methods may significantly vary in different applications. In Willjuice and Baskar (2009) comparative performance analysis of various EAs such as real coded genetic algorithm (RGA) with SBX crossover, differential evolution (DE), modified particle swarm optimization (MPSO) and covariance matrix adaptation evolution strategy (CMAES) was done and better performance of CMAES and MPSO in comparison to BLT, RGA with SBX and multi-crossover approaches, was reported in the paper. The MPSO algorithm is a variant of real coded particle swarm optimization algorithm.

The particle swarm optimization algorithm (PSO) was introduced by Kennedy and Eberhart (1995) by simulation of swarms behavior in performing their tasks. The PSO has several advantages, it works by maintaining a population of solutions and hence allows for parallel evaluations of several solutions, it does not require that the optimization problem must be differentiable and comprises very simple mathematics. The PSO's simplicity and capability of solving very difficult problems have motivated many researchers for its further development. Some recent developments can be seen in Baskar and Suganthan (2004), Zhao and Suganthan (2009), Van den Bergh and Engelbrecht (2004) and Zhan et al. (2009).

Although continuous PSO and most of its variants have been successfully applied in many real-world engineering applications, it is widely held that PSO gets trapped in local optima. Furthermore, many of proposed strategies to deal with the weaknesses of continuous PSO algorithm have significantly increased the computational costs.

In 1997, Kennedy and Eberhart further extended the continuous PSO algorithm to deal with the combinatorial optimization problems and proposed a discrete binary version of PSO. Unlike continuous PSO, the discrete version of PSO (DBPSO) uses binary bits to represent each dimension of particle position vector. The binary coded PSO algorithm can cover a wide range of applications as binary sequences can be transformed to match the requirements of any problem space.

With the purpose of finding a suitable tuning technique, this paper is focused on the performance evaluation of various binary coded PSOs such as DBPSO, MBPSO, and PBPSO algorithms on optimal design of multivariable PI and PID controllers with and without decoupling of process described by Wood and Berry (1973), Chang (2007) and Willjuice and Baskar (2009) which is considered as a case study for the MIMO tuning problem.

In addition, the comparative performance analysis of computational techniques is useful in updating and integrating current developments for further research and development.

The remaining paper is organized as follows: Section 2 describes various methods under consideration, Section 3 illustrate implementation of proposed methods, Section 4 details experiments and simulation results, finally conclusions are drawn in Section 5.

2. Techniques

This section briefly explains the PSO and some binary PSO variants.

2.1. Particle swarm optimization (PSO)

The PSO was introduced by Kennedy and Eberhart by simulating social behavior of birds flocks in 1995 (Kennedy & Eberhart, 1995). It works by having a group of m particles. Each particle can be considered as a candidate solution to an optimization problem and it can be represented by a point or a position vector $x_{ij} = [x_{i1}, \ldots, x_{id}]$ in a d dimensional search space which keeps on moving toward new points in the search space with the addition of a velocity vector $v_{ij} = [v_{i1}, \ldots, v_{id}]$ to further facilitate the search procedure. The initial positions and velocities of particles are random from a normal population $u \in [0,1]$. All particles move in the search space to optimize an objective function $f(x)$. Each member of the group gets a score after its evaluation on objective function $f(x)$; the score is regarded as a fitness value. The member with the highest score is called global best. Each particle memorizes its previous best positions. During the search process all particles move toward the areas of potential solutions by utilizing the cognitive and social learning components. The process is repeated until any prescribed stopping criterion is reached. After any iteration, each particle updates its position and velocity to achieve better fitness values according to the following Eqs. (1) and (2)

$$V_{ij}(t+1) = w \cdot V_{ij}(t) + c_1 \cdot r_1(p_{ij}(t) - x_{ij}(t)) + c_2 \cdot r_2(g_{1j}(t) - x_{ij}(t)) \quad (1)$$

$$x_{ij}(t+1) = V_{ij}(t+1) + x_{ij}(t) \quad (2)$$

where c_1, c_2 are two constants, called acceleration factors; w is inertia weight; $V(t)$ is velocity of each particle during iteration t; $x(t)$ is current position of particle at iteration t; $p(t)$ is previous best position of each particle till t; $g(t)$ is the position of the best particle in the group; r_1, r_2 are two quasi random numbers $u \in [0,1]$; t is the current iteration time or index.

The pseudo code of PSO algorithm for the multivariable PID controller design is provided in Fig. 1.

2.2. Discrete binary PSO (DBPSO) algorithm

Kennedy and Eberhart introduced a discrete version of the PSO aiming to deal with combinatorial optimization problems. In discrete binary PSO (DBPSO) (Kennedy & Eberhart, 1997), particle position vector is represented by a binary string with each component of this vector bounded to have a value either one or zero. In DBPSO, velocity update rule Eq. (1) of the PSO was preserved, however, it was considered as a pseudo probability for any component of position vector to take a value in the binary do-

Pseudo code of PSO for Multivariable PID controller design

1. Initialize the swarm of size N for the controller parameters $K_{p1}, K_{i1}, K_{d1}, K_{p2}, K_{i2}, K_{d2}$ each representing dimensions of particle's position vector in real parameter space.
2. for each particle ***i*** **do**
   Evaluate each particle on objective function ***f***.
3. Set initial $\mathbf{p}_{best}$ and $\mathbf{g}_{best}$.
4. **While** stopping criteria is not reached **do**
   for every particle ***i*** **do**
   update velocity ***V*** according to equation (1)
   update position ***X*** according to equation (2)
5. for each particle ***i*** **do**
   Evaluate each particle on objective function ***f***.
6. Renew $\mathbf{p}_{best}$ and $\mathbf{g}_{best}$.
7. end **while**
8. Return ***gbest solution***.

Fig. 1. Pseudo code of PSO for multivariable PID controller design.