Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Engineering 15 (2011) 4920 – 4924
Procedia Engineering
www.elsevier.com/locate/procedia

Advanced in Control Engineering and Information Science

Conflict Resolution in Product Optimization Design based on Adaptive Particle Swarm Optimization

Xiaolei Wang [a]*

[a]School of Mechanical Engineering, North China Electric Power University, Baoding 071003, China

Abstract

To resolving the conflict in engineering optimization design of highly complex and nonlinear constraints, a new particle swarm optimization algorithm with adaptive inertia weight is proposed. In this algorithm, inertia weight is adaptively changed according to the current evolution speed and aggregation degree of the swarm, which provides the algorithm with dynamic adaptability, enhances the search ability and convergence performance of the algorithm. Finally, the validity of the algorithm is verified through an optimization example.

Keywords: Conflict Resolution, optimization design, particle swarm optimization, evolution speed, aggregation degree;

1. Introduction

In collaborative product designs, the conflicts of collaborative product design are opposition and lack of agreement in design goals or variables determination among designer. Supposed that, X_j *is* the available design variable *of* designer D_j （$X_j \in Q_j$） .

Usually, design team expects the design is the optimal in each sub goal, that is

$$\begin{cases} \text{Max} \quad (O_1(X_j,\overline{X_j}),\ O_2(X_j,\overline{X_j}),\ \cdots,\ O_i(X_j,\overline{X_j}),\cdots,\ O_k(X_j,\overline{X_j})) \\ \quad X_j \in Q_j \\ S.t. \quad g_i(X) \le 0 \quad i = 1, 2, \cdots, I. \end{cases} \tag{1}$$

We can know, as product and engineering design becomes more and more complicated, the objective function of optimization design is increasingly high dimensional, non-convex, and highly nonlinear. Therefore, finding a simple optimization method that can obtain global optimal quickly and effectively has an important significance to design conflict resolution and optimization.

In recent years, particle swarm optimization (PSO) has been proven to be a better global optimization method with simple operation and parallel search[1]. PSO is a stochastic optimization technique

* Corresponding author. Tel.: 15081271728,03127525049.
E-mail address: wxlwxl916@126.com.

1877-7058 © 2011 Published by Elsevier Ltd.
doi:10.1016/j.proeng.2011.08.916

motivated by the behavior of a flock of birds. It was first developed and introduced by Kennedy and Eberhart in 1995[2]. The algorithm is robust, well suited to handle non-linear, non-convex design problems. Recently, Because of its high convergence speed and relative simplicity, it has attracted a lot of attention from researchers and been successfully used to solve global optimization problems in engineering design[3,4,5]. But in PSO, particles demonstrate strong convergence character during searching process, leading to rapidly species diversity loss and easily premature convergence. Thus, how to choose a reasonable value of inertia weight is the key to solve this problem. Aiming at the problem, we improve PSO to provide a more efficient and effective design optimization method.

2. Basic PSO and LDIPSO

In Basic PSO, population is called a swarm and individuals are referred as particles. Particles fly in the n-dimensional search space according to a speed. Suppose that there are M particles in the swarm, Each particle represents a possible solution to the optimization problem. Particle i has a position $x_i=(x_{i1}, x_{i2}, \cdots, x_{in})$ and a velocity $v_i=(v_{i1}, v_{i2}, \cdots, v_{in})$.

Assume that particle i attains the best position in the current iteration t, the position and velocity of particles are adapted according to the following equations.

$$v_{id}(t+1)=wv_{id}(t)+ c_1r_1(p_{id}(t)-x_{id}(t))+ c_2r_2(p_{gd}(t)-x_{id}(t)). \tag{2}$$

$$x_{id}(t+1)= x_{id}(t)+v_{id}(t+1) \quad i=1, 2,\cdots, M; d=1, 2,\cdots, n. \tag{3}$$

where v_{id} is the velocity of particle i in dimension d, r_1 and r_2 are random numbers in the interval [0, 1]. c_1 and c_2 are two positive constants, denoting cognitive coefficient and social coefficient respectively. Typically, they are both set to a value of 2.0. $P_{best}(t)$ is the personal best position of particle i at t-th iteration, $g_{best}(t)$ is the global best position of the swarm at t-th iteration. w is inertia weight, it is employed to control the exploration abilities and convergence behaviour of the swarm.

PSO with Linearly Decreasing Inertia weight (LDIPSO). As a swarm intelligence method, BPSO always convergence prematurely, especially in complex multi-peak optimization problems. Experiments have proved that large w allow for wide velocity updates, while small inertia values concentrate the velocity updates. It is better to initially set w to a large value to make better global exploration and gradually decrease the weight to get more refined solutions. In order to get a better global optimal solution, w is usually decreased linearly by Eq.4.

$$w(t) = w_{\max} - \frac{(w_{\max} - w_{\min}) * t}{Max_{iter}} \tag{4}$$

where, w_{max} and w_{min} is the maximum and minimum value of inertia weight w, t is the current iteration, Max_{iter} is the maximum number of iterations.

However, in LDIPSO, on one hand, w is only related to the number of iteration, it can not make adjustments effectively according to the non-linear changing. On the other hand, as w linearly decreases, LDIPSO lacks global search ability in the latter stage of search process, easily converges to local optima.

3. PSO with adaptive inertia weight

In order to deal with these problems, an improved PSO with adaptive inertia weight, called AIWPSO, is proposed. Inertia weight w changes adaptively and dynamically according to the evolution speed and aggregation degree of the swarm

Evolution speed of swarm. An important factor which influences the performance of PSO is the evolution speed. In search process, the current fitness value of global optimal solution $f(g_{best}(t))$ should be