Annals of Pure and Applied Mathematics
Vol. 5, No.1, 2013, 82-89
ISSN: 2279-087X (P), 2279-0888(online)
Published on 13 November 2013
www.researchmathsci.org

Annals of
Pure and Applied Mathematics

Advanced Optimization Technique

Dipak Kumar Biswas[1] **and** ***Subash Chandra Panja***[2]

[1]Department of Mathematics, Women's Polytechnic, Jodhpur Park
Kolkata 700068, India. E-mail: dkb.ju08@hotmail.com
[2]Department of Mechanical Engineering, Jadavpur University, Jadavpur
Kolkata 700032, India. E-mail: panja12@gmail.com

Received 20 September 2013; accepted 16 October2013

Abstract. Optimization is a mathematical technique that concerns the finding of maxima or minima of functions within some feasible region. A diversity of optimization techniques fight for the best solution. Particle Swarm Optimization (PSO) is a comparatively new, current, and dominant method of advanced optimization technique that has been empirically shown to perform well on many of these optimization problems. It is lucidly and widely used to find the global optimum solution in a complex search space. This article aims at providing and illustrating for discussion of the most established results on PSO algorithm as well as exposing the most active research topics that can give proposal for future work and help the practitioner improves better result with little effort. This paper also introduces a detailed explanation of the PSO algorithm.

Keywords: Optimization; PSO; dynamic programming; inertia component; ACO

1. Introduction

Scientists, engineers, economists, and managers always have to take many technological and managerial decisions at several times for construction and maintenance of any system. Day by day the world becomes more and more complex and competitive so the decision making must be taken in an optimal way. Therefore optimization is the main act of obtaining the best result under given situations. Optimization originated in the 1940s, when the British military faced the problem of allocating limited resources (for example fighter airplanes, submarines and so on) to several activities [1]. Over the decades, several researchers have generated different solutions to linear and non-liner optimization problems. Mathematically an optimization problem has a fitness function, describing the problem under a set of constraints which represents the solution space for the problem. However, most of the traditional optimization techniques have calculated the first derivatives to locate the optima on a given constrained surface. Due to the difficulties in evaluation the first derivative for many rough and discontinuous optimization spaces, several derivatives free optimization methods have been constructed in recent time.

There is no known single optimization method available for solving all optimization problems. A lot of optimization methods have been developed for solving different types of optimization problems in recent years. The modern optimization methods (sometimes called nontraditional optimization methods) are very powerful and popular methods for solving complex engineering problems. These methods are dynamic programming (DP), particle swarm optimization algorithm (PSO), neural networks (NN), genetic algorithms (GA), ant colony optimization (ACO), artificial immune systems (AIS), and fuzzy optimization (FO)[2].

The Particle Swarm Optimization algorithm (abbreviated as PSO) is a novel population-based stochastic search algorithm and an alternative solution to the complex non-linear optimization problem. The PSO algorithm was first introduced by Dr. Kennedy and Dr. Eberhart in 1995 and its basic idea was originally inspired by simulation of the social behavior of animals such as bird flocking, fish schooling and so on. It is based on the natural process of group communication to share individual knowledge when a group of birds or insects search food or migrate and so forth in a searching space, although all birds or insects do not know where the best position is. But from the nature of the social behavior, if any member can find out a desirable path to go, the rest of the members will follow quickly [3].

2. Optimization

Optimization is the mechanism or technique by which one finds the maximum or minimum value of a function or process. The mechanism is used in the fields of Physics, chemistry, economics, and engineering.

Mathematically, a minimization is defined as:

Given $f : R^n \to R$

Find $\hat{x} \in R^n$ such that $f\left(\hat{x}\right) \leq f(x), \forall x \in R^n$

Similarly, maximization is defined as:

Given $f : R^n \to R$

Find $\hat{x} \in R^n$ such that $f\left(\hat{x}\right) \geq f(x), \forall x \in R^n$

The domain R^n of f is referred as the search space. Each element of R^n is called a candidates solution with $\hat{x}$ being the optimal solution and R^n denotes n dimension search space. The function f is called the object function which maps the search space to the function space. Since a function has only one output, this function space is usually one-dimensional. The function space is then mapped to the one-dimensional fitness space, providing a single fitness value for each set of parameters. This single fitness value determines the optimality of the set of parameters for the desired task.

For a known (differentiable) function f, calculus can fairly easily provide us with maxima and minima of f. But in real-life optimization tasks, this object function f is often not directly known. Instead the objective function is a “black box” to which we apply parameters (the candidate solution) and receive an output value. The result of this evaluation of a candidate solution becomes the solution’s fitness. The final goal of an optimization task is to find the parameters in the search space that maximize or minimize this fitness.

In some optimization tasks, called constrained optimization tasks, the elements in a candidate solution can be subject to certain constraints such as being grater that or less than zero.

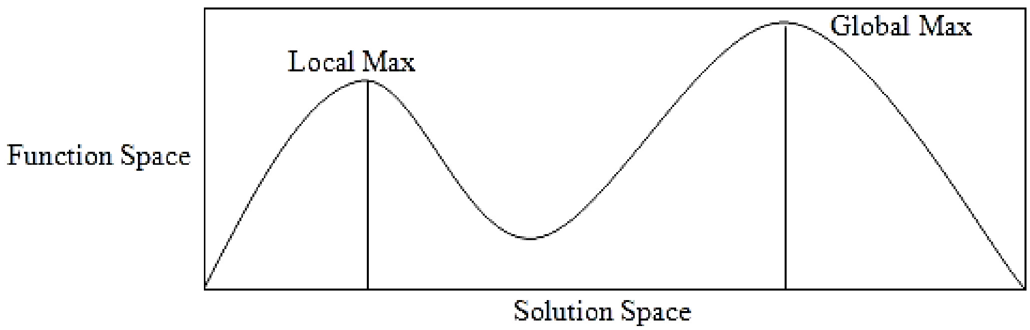

Figure 1: Describe Local and Global maximum

3. The model

The PSO algorithm works by simultaneously maintaining several candidate solutions in the search space. During each iteration of the algorithm, each candidate solution is evaluated by the objective function being optimized, determining the fitness of the solution. Each candidate solution can be thought of as a particle "flying" through the fitness landscape finding the maximum or the minimum of the objective function.[4]

Initially, the PSO algorithm chooses candidate solutions randomly within the search space. Figure 2 shows the initial state of a four-particle PSO algorithm seeking the global maximum in one-dimensional search space. The search space is composed of all the possible solutions along the x-axis; the curve denotes the objective function.

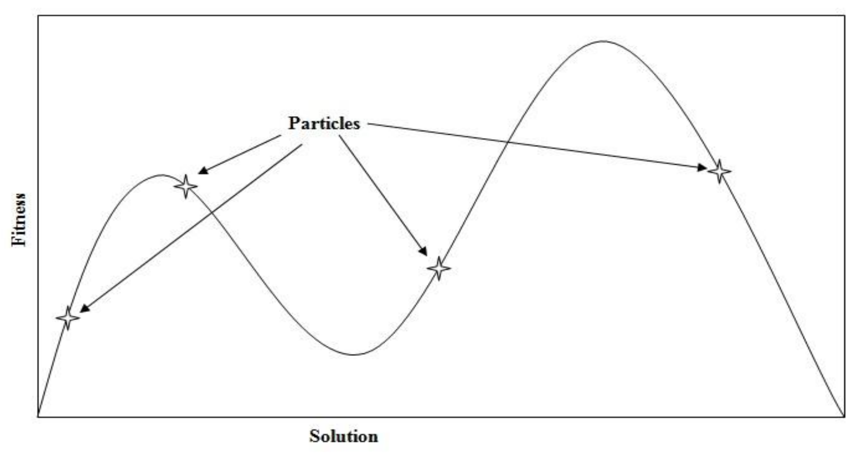

Figure 2: Describe Initial PSO State

It should be noted that the PSO algorithm has no knowledge of underlying objective function, and thus has no way of knowing if any of the candidate solutions are near to or far away from the local or global maximum. The PSO algorithm simply uses the object function to evaluate its candidate solutions, and operates upon the resultant fitness values.

Each particle maintains its position, composed of candidate solutions and its evaluated fitness, and its velocity. Additionally, it remembers the best fitness value it has achieved thus for during the operation of the algorithm, referred to as the individual best fitness, and the candidate solution that achieved this fitness, referred to as the individual best position or individual best candidate solution. Finally, the PSO algorithm maintains the best fitness value achieved among all particles in the swarm, called the global best fitness, and the candidate solution that achieved this fitness called the global best position or global best candidate solution.

4. Neighborhood topologies

Swarm topology

In PSO, there have been two basic topologies used in the literature

1. Ring Topology (neighborhood of 3)
2. Star Topology (global neighborhood)

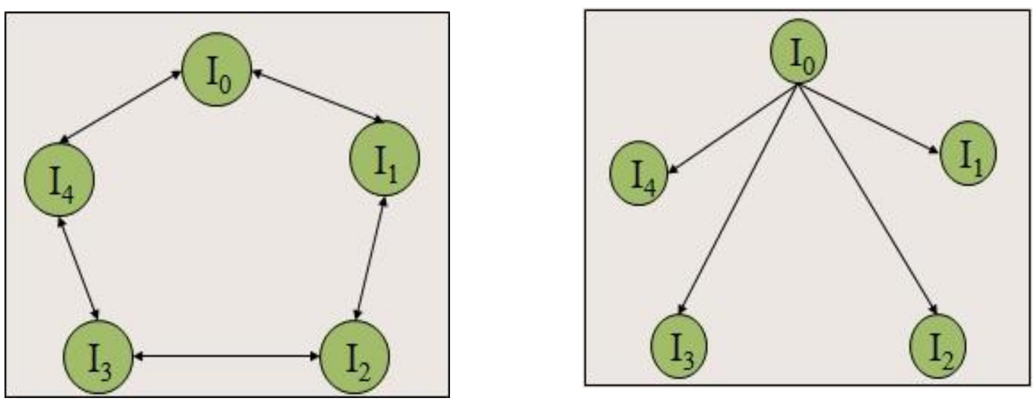

Figure 4: Describe Ring Topology and Star Topology

The anatomy of a particle

A particle (individual) is composed of:

Three vectors:

1. The **x-vector** records the current position (location) of the particle in the search space,
2. The **p-vector** records the location of the best solution found so far by the particle, and
3. The **v-vector** contains a gradient (direction) for which particle will travel in if without interruption.

Two fitness values:

1. The **x-fitness** records the fitness of the x-vector, and
2. The **p-fitness** records the fitness of the p-vector.

5. Calculation steps

The PSO algorithm consists of just four steps, which are repeated until some stopping condition is met

1. Population is initialized by assigning random position and velocities
2. The fitness of each particle is evaluated.
3. Individual and global best fitness and positions are updated.
4. The velocity and position of each particle is updated.

The first three steps are fairly trivial. Fitness evaluation is conducted by supplying the candidate solution to the objective function. Individual and global best fitness and positions are updated by comparing the best fitness and positions as necessary.

The velocity and position update step is responsible for the optimization ability of the PSO algorithm. The velocity and position of each particle in the swarm is updated using the following equations:

1. $v_i(t+1) = \omega v_i(t) + c_1 r_1[\hat{x}_i(t) - x_i(t+1)] + c_2 r_2[g(t) - x_i(t)]$
2. $x_i(t+1) = x_i(t) + v_i(t+1)$

The index of the particle is represented by *i*. Thus $v_i(t)$ is the velocity of particle i at any time t and $x_i(t)$ is the position of particle i at time t. The parameters ω, c_1 and c_2 $(0 \le \omega \le 1.2, 0 \le c_1 \le 2 and 0 \le c_2 \le 2)$ are random values regenerated for each velocity update. The value $\hat{x}_i(t)$ is the individual best candidate solution for particle i at time t, and g(t) is the swarm global best candidate solution at time t.

Each of the three terms of the velocity update equation has different roles in the PSO algorithm. The first term $\omega v_i(t)$ **is the inertia component**, responsible for keeping the particle moving in the same direction it was originally heading. The value of the "inertial coefficient ω" is typically between 0.8 and 1.2, which can either dampen the particle's inertia or accelerate the particle in the original direction. Generally, lower values of the inertial coefficient speed up the convergence of the swarm to optima, and higher values of the inertial coefficient encourage exploration of the entire search space.

The second term, $c_1 r_1[\hat{x}_i(t) - x_i(t)]$, called the **cognitive component**, acts as the particle's memory, causing it to tend to return to the regions of the search space in which it has experienced high individual fitness. The cognitive coefficient c_1 is usually close to 2, and affects the size of the steps the particle takes toward its individual best candidate solution $\hat{x}_i$.

The third term, $c_2 r_2[g(t) - x_i(t)]$, called the **social component**, cause the particle to move to the best region the swarm has found so far. The social coefficient c_2 typically close to 2, and represents the size of the step the particle takes toward the global best candidate solution $g(t)$ the swarm has found up until the point.

The random values r_1 and r_2 cause the cognitive and social components to have stochastic influence on the velocity update. This stochastic nature cause each particle to move in a semi-random manner heavily influenced in the directions of the individual best solution of the particle and global best solution of the swarm.

In order to keep the particles from moving too far beyond the search space, we use a technique called velocity clamping to limit the maximum velocity of each particle. For a search space bounded by the range $[-x_{\max}, x_{\max}]$, velocity clamping limit to the range of $[-v_{\max}, v_{\max}]$, where $v_{\max} = k \times x_{\max}$. The value of k represents a user-supplied velocity clamping factor, $0.1 \leq k \leq 1.0$. In many optimization tasks, such as the ones discussed in the paper, the search space is not centered on 0 and the range $[-x_{\max}, x_{\max}]$ is not an adequate definition of the search space. In such a case where the search space is bounded by $[-x_{\max}, x_{\max}]$, we define $v_{\max} = k \times (x_{\max} - x_{\min})/2$

Once the velocity for each particle is calculated, each particle's position is updated by applying the new velocity to the particle's previous position:

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

This process is repeated until some stopping condition is met. Some common stopping conditions include: a present number of iterations of the PSO algorithm, a number of iterations since the last update of the global best candidate solution or a predefined target fitness value.

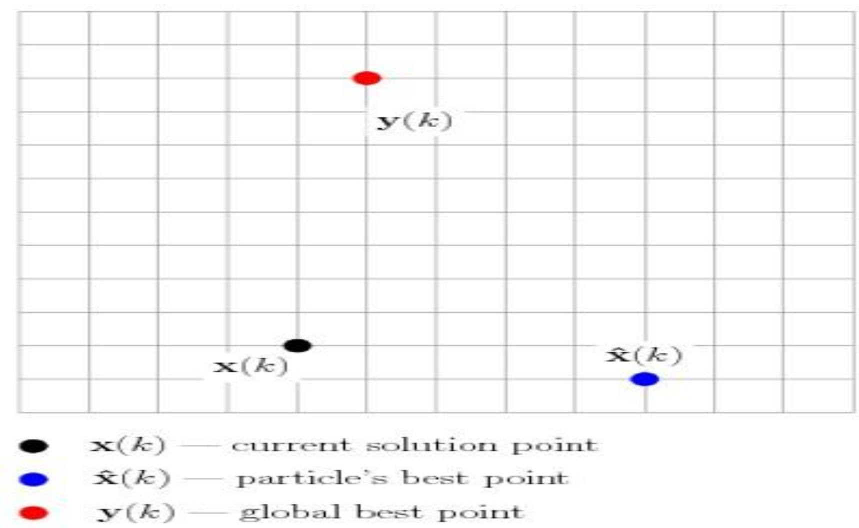

Figure 3: Describe PSO Solution update in 2D

The particle velocities build up too fast and the maximum of the objective function is passed over. In PSO, particle velocity is very important, since it is the step size of the swarm. At each step, all particles proceed by adjusting the velocity that each particle moves in every dimension of the search space. There are two characteristics:

- Exploration
- Exploitation

Exploration: Exploration is the ability to explore different area of the search space for locating a good optimum.

Exploitation: Exploitation is the ability to concentrate the search around a searching area for refining a hopeful solution.

Therefore these two characteristics have to balance in a good optimization algorithm. When the velocity increases to large values, then particle's positions update quickly. As a result, particles leave the boundaries of the search space and diverge. Therefore, to control this divergence, particles' velocities are reduced in order to stay within boundary constraints. The following techniques have been developed to improve speed of convergence, to balance the exploration- exploitation trade-off, and to find a quality of solutions for the PSO:

6. Velocity clamping

Eberhart and Kennedy first introduced velocity clamping; it helps particles to stay within the boundary and to take reasonably step size in order to comb through the search space. Without this velocity clamping in the searching space the process will be prone to explode and particles' positions change rapidly. Maximum velocity controls the granularity of the search space by clamping velocities and creates a better balance between global exploration and local exploitation.

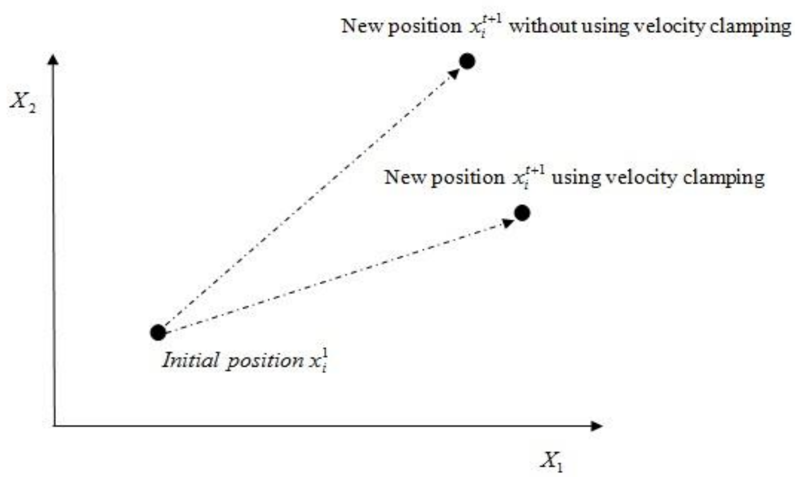

Figure 4: Illustration of effects of Velocity clamping

7. Inertia weight

The inertia weight, denoted by ω, is considered to replace $V_{\max}$ by adjusting the influence of the previous velocities in the process, i.e. it controls the momentum of the particle by weighing the contribution of the previous velocity. The inertia weight 'ω' will at every step be multiplied by the velocity at the previous time step, i.e. $v_i(t)$ therefore, in the 'gbest' PSO, the velocity equation of the particle with the inertia weight changes from equation

$$v_i(t+1) = v_i(t) + c_1 r_1[\hat{x}_i(t) - x_i(t)] + c_2 r_2[g(t) - x_i(t)]$$

to

$$v_i(t+1) = \omega v_i(t) + c_1 r_1[\hat{x}_i(t) - x_i(t)] + c_2 r_2[g(t) - x_i(t)]$$

In the 'lbest' PSO, the velocity equation changes in a similar way as the above velocity equation do.

7. Conclusion

Apart from the canonical PSO algorithm described above, many variations of the PSO algorithm exit. Unlike in genetic algorithms, evolutionary programming and evolutionary strategies, in PSO, there is no selection operation. All particles in PSO are kept as members of the population through the course of the run. PSO are the only algorithm that does not implement the survival of the fittest. No crossover operation in PSO. In EP balance between the global and local search can be adjusted through the strategy parameter while in PSO the balance is achieved through the inertial weight factor (w). Discrete PSO can handle discrete binary variables. MINLP PSO can handle both discrete binary and continuous variables. Hybrid PSO utilizes basic mechanism of PSO and the natural selection mechanism, which is usually utilized by EC methods such as GAs.

REFERENCES

1. S.S.Rao, *Engineering Optimization Theory and Practice*, 4th edition, Ed.: John Wiley and Sons, (2009).
2. J. Kennedy and R. Eberhart, PSO in Proceeding of the IEEE International Conference on NN, IV (2005), 1942-1948.
3. Frans van den Bergh, An Analysis of PSO. PhD thesis, University of Pretoria, (2001).
4. E.Zitzler, M.Laumanns and S.Bleuler. A tutorial on evolutionary multiobjective optimization, (2002).