Particle Swarm Optimization - The Fundamental Insight

Dr. G. H. Upadhyay[1] Mr. J. R. Vaghela[2] Mr. N. N.Chaudhary[3]
Prof. B. C. Khatri[4] Prof. U. V. Shah[5]
[1]H.O.D. Mechanical Dept. [2, 3] P. G. Student [4,5]Associate professor
[1, 2, 3, 4, 5] L.D. College of Engineering, Ahmedabad, Gujarat

I. INTRODUCTION TO OPTIMIZATION METHODS

1) In highly competitive manufacturing industries nowadays, the manufactures ultimate goals are to produce high quality product with less cost and time constraints. To achieve these goals, one of the considerations is by optimizing the machining process parameters.
2) Optimization is the act of obtaining the best result under given circumstances. In design, construction, and maintenance of any engineering system, engineers have to take many technological and managerial decisions at several stages. The ultimate goal of all such decisions is either to minimize the effort required or to maximize the desired benefit. Since the effort required or the benefit desired in any practical situation can be expressed as a function of certain decision variables.[5]
3) Optimization can be defined as the process of finding the conditions that give the maximum or minimum value of function under given constraints. The various methods that used in optimization can be described as below.

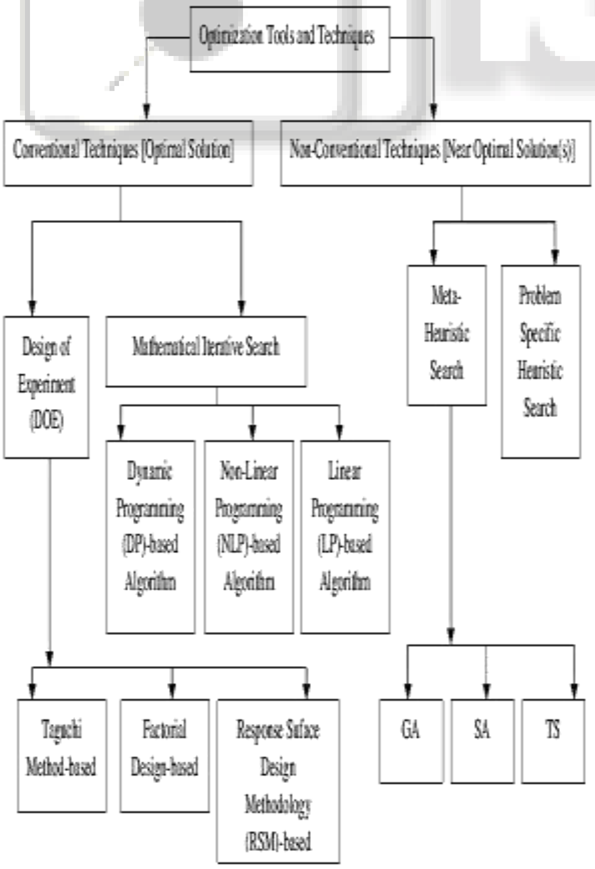


Fig. 1: Different Optimization Methods [1]

II. NON-TRADITIONAL OPTIMIZATION ALGORITHMS

In recent years, some optimization methods that are conceptually different from the traditional mathematical programming techniques have been developed. These methods are labelled as modern or non-traditional methods of optimization. Most of these methods are based on certain characteristics and behaviour of biological, molecular, swarm of insects, and neurobiological systems. The following methods are comes under non-traditional optimization methods:

1) Genetic algorithms
2) Simulated annealing
3) Particle swarm optimization
4) Ant colony optimization
5) Artificial bee colony algorithm (ABC)
6) Sheep flock algorithm (SFA)
7) Biogeography-based optimization (BBO)
8) Fuzzy optimization
9) Neural-network-based methods

Most of these methods have been developed only in recent years and are emerging as popular methods for the solution of complex engineering problems. Most require only the function values (and not the derivatives).

1) The genetic algorithms are based on the principles of natural genetics and natural selection.
2) Simulated annealing is based on the simulation of thermal annealing of critically heated solids. Both genetic algorithms and simulated annealing are stochastic methods that can find the global minimum with a high probability and are naturally applicable for the solution of discrete optimization problems.
3) The particle swarm optimization is based on the behaviour of a colony of living things, such as a swarm of insects, a flock of birds, or a school of fish.
4) Ant colony optimization is based on the cooperative behaviour of real ant colonies, which are able to find the shortest path from their nest to a food source. In many practical systems, the objective function, constraints, and the design data are known only in vague and linguistic terms [6].
5) In Artificial bee colony algorithm (ABC) algorithm, the position of a food source represents a possible solution to the considered optimization problem and the nectar amount of the food source is proportional to the quality or fitness of the associated solution.[2]
6) Sheep flock (SF) algorithm is a new evolutionary computation method based on sheep flocks heredity. When several sheep of one flock are inevitably mixed with the other flocks, the characteristics of the sheep in the neighbouring flocks can be inherent to the sheep in

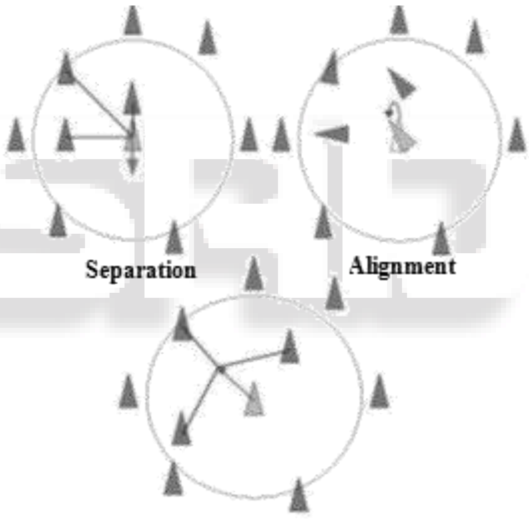
other flocks in this occasion. Then, in the field, the flock of the sheep which has better fitness characteristics breeds most.[2]

7) The fundamental idea behind BBO algorithm is how species migrate from one island to another, how new species arise, and how species become extinct.[2]
8) Fuzzy optimization methods have been developed for solving such problems.
9) In neural-network-based methods, the problem is modelled as a network consisting of several neurons, and the network is trained suitably to solve the optimization problem efficiently [6].
10) We are going to discuss PSO as a potential tool of optimization process.

III. PARTICLE SWARM OTIMIZATION (PSO)

1) Particle swarm optimization, abbreviated as PSO, is based on the behaviour of a colony or swarm of insects, such as ants, termites, bees, and wasps; a flock of birds; or a school of fish.
2) The particle swarm optimization algorithm mimics the behaviour of these social organisms. The word ‗particle' denotes, for example, a bee in a colony or a bird in a flock.
3) Each individual or particle in a swarm behaves in a distributed way using its own intelligence and the collective or group intelligence of the swarm. As such, if one particle discovers a good path to ‗food', the rest of the swarm will also be able to follow the good path instantly even if their location is far away in the swarm.
4) Optimization methods based on swarm intelligence are called behaviourally inspired algorithms as opposed to the genetic algorithms, which are called evolution-based procedures.
5) The PSO algorithm was originally proposed by Kennedy and Eberhart in 1995. In the context of multivariable optimization, the swarm is assumed to be of specified or fixed size with each particle located initially at random locations in the multidimensional design space.
6) Each particle is assumed to have two characteristics: a position and a velocity. Each particle wanders around in the design space and remembers the best position (in terms of the food source or objective function value) it has discovered.
7) The particles communicate information or good positions to each other and adjust their individual positions and velocities based on the information received on the good positions.
8) As an example, consider the behaviour of birds in a flock. Although each bird has a limited intelligence by itself, it follows the following simple rules:
   1) It tries not to come too close to other birds.
   2) It steers toward the average direction of other birds.
   3) It tries to fit the ―average position" between other birds with no wide gaps in the flock. [6]
9) Thus the behaviour of the flock or swarm is based on a combination of three simple factors:
   1) Separations— don't come too close.
   2) Alignment— follows the general heading of the flock.'
   3) Cohesion— stick together.
9) The PSO is developed based on the following model:
   1) When one bird locates a target or food (or maximum of the objective function), it instantaneously transmits the information to all other birds.
   2) All other birds gravitate to the target or food (or maximum of the objective function), but not directly.
   3) There is a component of each bird's own independent thinking as well as its past ‗memory'.

Fig. 2: Bird flock, School of fish [4]



IV. PSO – A METHODOLOGY

1) Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy (1995) inspired by social behaviour of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as genetic algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation.
2) In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. The detailed information will be given in following sections.
3) Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization, artificial neural network

training, fuzzy system control, and other areas where GA can be applied.

4) In the multivariate optimization procedure, a point in a problem spaces is called a particle which it has a random initial position and velocity. Each particle can wander around the problem spaces, find the best positions (best values of objective function) and remember them. Then the particles can communicate to each other and modified the velocities and positions to find approach to global optimum. The pseudo program for step by step implementation of particle swarm optimization procedure is expressed as follow:
5) Consider an unconstrained maximization problem:
6) Maximize f (X) with $X^{(l)} \le X \le X^{(u)}$ where $X^{(l)}$ and $X^{(u)}$ denote the lower and upper bounds on X, respectively.

A. First step (initializing):

Set the number of particles in the swarm N and allocate the random locations and velocities for each particle in swarm x_j, $v_j (j = 1, . . . N)$. Set iteration number i = 0, we must assume a smaller size of the swarm. But with too small a swarm size it is likely to take us longer to find a solution or, in some cases, we may not be able to find a solution at all. Usually a size of 20 to 30 particles is assumed for the swarm as a compromise.

B. Second step (calculation of objective function):

Generate the initial population of X in the range $X^{(l)}$ and $X^{(u)}$ randomly as $X_1, X_2, . . . , X_N$. Hereafter, for convenience, the particle (position of) j and its velocity in iteration i are denoted as $X_j(i)$ and $V_j(i)$, respectively. Thus the particles generated initially are denoted $X_1(0), X_2(0), . . . , X_N(0)$. The vectors $X_j(0)$ (j = 1, 2, . . . ,N) are called particles or vectors of coordinates of particles (similar to chromosomes in genetic algorithms). Evaluate the objective function values corresponding to the particles as f $[X_1(0)]$, f $[X_2(0)]$, . . . , f $[X_N(0)]$.

C. Third step (evaluation of best solutions):

1) Find the two important used by a typical particle j

(a) The historical best value of $X_j(i)$ (coordinates of jth particle in the current iteration i), $P_{best,j}$, with the highest value of the objective function, $f[X_j(i)]$, encountered by particle j in all the previous iterations.
The historical best value of $X_j(i)$ (coordinates of all particles upto that iteration), G_{best}, with the highest value of the objective function $f[X_j(i)]$, encountered in all the previous iterations by any of the N particles.

(b) Find the velocity of particle j in the i^{th} iteration as follows:

$$V_j(i) = V_j(i - 1) + c_1 r_1 [P_{best,j} - X_j(i - 1)] + c_2 r_2 [G_{best} - X_j(i - 1)];\ j = 1, 2, . . . , N$$

Where $c_1$and c_2 are the cognitive (individual) and social (group) learning rates, respectively, and r_1 and r_2 are uniformly distributed random numbers in the range 0 and 1. The parameters c_1 and c_2 denote the relative importance of the memory (position) of the particle itself to the memory (position) of the swarm. The values of c_1 and c_2 are usually assumed to be 2 so that $c_1 r_1$ and $c_2 r_2$ ensure that the particles would overfly the target about half the time.

(c) Find the position or coordinate of the j^{th} particle in i^{th} iterations:

$$X_j(i) = X_j(i - 1) + V_j(i);\ j = 1, 2, . . . , N.$$

Then evaluate the objective function values corresponding to the particles as $f[X_1(i)], f[X_2(i)], . . ., f[X_N(i)]$.

D. Fourth step (stop condition):

Check the convergence of the current solution. If the positions of all particles converge to the same set of values, the method is assumed to have converged. If the convergence criterion is not satisfied, step 3 is repeated by updating the iteration number as i = i + 1, and by computing the new values of $P_{best,j}$ and G_{best}. The iterative process is continued until all particles converge to the same optimum solution.[6],[7],[3]

1) Thus the model simulates a random search in the design space for the maximum value of the objective function. As such, gradually over much iteration, the birds go to the target (or maximum of the objective function).
2) Particle swarm optimization (PSO) is a stochastic optimization technique which is initialized with a population of random solutions and searches for the optimal by updating generations.
3) Unlike GA, it has no evolution operators, such as crossover and mutation. In PSO algorithm, the potential solutions, called particles, fly through the problem space by following the current optimal particles.
4) Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. This value is called p_{best}. Another ‗best' value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called l_{best}.
5) When a particle takes all the population as its topological neighbors, the best value is a global best and is called g_{best}. The PSO concept consists of, at each time step, changing the velocity of (accelerating) each particle towards its p_{best} and l_{best} locations.
6) Acceleration is weighted by a random term, with separate random numbers being generated for acceleration towards p_{best} and l_{best} locations. The movements of the particles are guided by their own best known position in the search space as well as the entire swarm‘s best known position. When improved positions are discovered, these will then come to guide the movements of the swarm [2].

REFERENCES

[1] Norfadzlan Yusup a,b, Azlan Mohd Zain a,⇑, Siti Zaiton Mohd Hashim, Evolutionary techniques in optimizing machining parameters: Review and recent applications (2007–2011), Expert Systems with Applications 39 (2012) 9909–9927
[2] Rajarshi Mukherjee, Shankar Chakraborty∗, Suman Samanta, Selection of wire electrical discharge machining process parameters using non-traditional optimization algorithms, Applied Soft Computing 12 (2012) 2506–2516

[3] Ali Vazini Shayana, Reza Azar Afzac, Reza Teimourib,Parametric study along with selection of optimal solutions in dry wirecut machining of cemented tungsten carbide (WC-Co), Journal of Manufacturing Processes xxx (2013) xxx–xxx
[4] Marco A. Montes de Oca, Particle Swarm Optimization Introduction Marco A. Montes de Oca IRIDIA-CoDE, Universit´e Libre de Bruxelles (U.L.B.).
[5] www.Wikipedia.com, the free encyclopaedia.
[6] Engineering Optimization: Theory and Practice, Fourth Edition Singiresu S. Rao.
[7] Advanced Modeling and Optimization of Manufacturing Processes by R.Venkata Rao. Sringer series in Advance Manufacturing.

