A Still Modified Approach to PSO Using Crossover

Anjali thareja
DITM,Ganaur

Dr.Archna Kumar
DITM,Ganaur

Abstract— Many literatures show the various hybrid version of PSO and prove them efficient. There are a lot of claims in which different crossover operators have shown to perform better. In order to show the performance of hybrid algorithm, this paper throws light on the behaviour of proposed algorithm a modified approach to PSO with crossover operator. Crossover operator used in this experiment is uniform crossover operator. Here, a modified approach to PSO has been approached. A criteria to select the particles for crossover and not to just select them up randomly. So that we can systematically and efficiently calculate local maxima of the region along with the global maxima where all the particles of the search space have to be converged. Thus a merging of matrix algorithm and a crossover algorithm on which the matrix algorithm has been applied is done. Various benchmark functions have also been used to prove the efficiency of these functions.

Keywords—Genetic Crossover, Matrix Based based PSO.

I. INTRODUCTION

Particle swarm optimization (PSO) is a population based heuristic search technique developed by Dr.Eberhart and Dr. Kennedy in 1995 [1], inspired by social behaviour of bird flocking or fish schooling. PSO has some properties similar with other evolutionary techniques such as Genetic algorithms (GA), like GA PSO also selects initial population randomly from solution search space. PSO does not use the operators like crossover and mutation like GA. As PSO is inspired from bird flocking it uses velocity equation to update the solutions and fly towards the best solution. This process continues, iteratively, until either the desired result is converged upon, or it's determined that an acceptable solution cannot be found within computational limits.

First the initial population is selected randomly from the solution search space than the position of particles are updated until the maximum limit of iteration or desired feasible solution found. For an n-dimensional search space, the ith particle of the swarm is represented by an n-dimensional vector, \( X_i = (x_{i1}, x_{i2}, ..., x_{in})^T \). The velocity of this particle is represented by another n-dimensional vector \( V_i = (v_{i1}, v_{i2}, ..., v_{in})^T \). The previously best visited position of the ith particle is denoted as \( P_i = (p_{i1}, p_{i2}, ..., p_{in}) \) [3] . ‘g’ is the index of the best particle in the swarm. The velocity of the ith particle is updated using the velocity update equation given by

\[
vid = vid + c1r1( pid - xid) + c2r2( pgd - xid) \quad (1)
\]

and the position is updated using

\[
xid = xid + vid \quad (2)
\]

where \( d = 1, 2, ..., n \); \( i = 1, 2, ..., S \), where \( S \) is the size of the swarm; \( c1 \) and \( c2 \) are constants, called cognitive and social scaling parameters respectively (usually, \( c1 = c2 ; r1 , r2 \) are random numbers, uniformly distributed in \([0, 1] \)). Equations (1) and (2) are the initial version of PSO algorithm. A constant, Vmax, is used to arbitrarily limit the velocities of the particles and improve the resolution of the search. Further, the concept of an inertia weight was developed to better control exploration and exploitation. The motivation was to be able to eliminate the need for Vmax. The inclusion of an inertia weight (w) in the particle swarm optimization algorithm was first reported in the literature in 1998 (Shi and Eberhart, 1998). The resulting velocity update equation becomes:

\[
vid = w * vid + c1 r1( pid - xid) + c2 r2( pgd - xid) \quad (3)
\]

II. GENETIC CROSSOVER

Genetic algorithms (GA) are a family of computational models developed by Holland who is inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information. GA are often viewed as function optimizers, although the range of problems to which GA have been applied is quite broad.

Crossover does the task of recombination of two individuals to generate individuals of the new generation. The new individual carries some of the characteristics of the first parent and the other characteristics from the other parent. The new individual generated in the process may be fitter than the parents or may be weaker. This depends upon the selection of the characteristics by the individual. The fitness of the individual is computed by the overall performance as a combined representation of the entire individual chromosome. Hence nothing certain can be assured regarding the fitness of individual unless it is measured. [7, 8] Crossover results in constant exchange of characteristics in the individual. GA derives a lot of computational optimality as a better in terms of fitness than both parents. The final result of the GA is the fittest individual. It is hence necessary to generate fitter individuals than parents so that the fitness of the best individual of the population pool keeps getting optimized and can finally be returned by the system.

In this paper, uniform crossover operator is taken as crossover method. Real coded uniform crossover generates two off springs from a pair of parents by uniformly replacing their elements on each locus at certain probability r. generally the value of r is chosen as 0.5.

First, a uniformly distributed random number \( u \in (0; 1) \) is generated. Then, if \( u > r \) is less than or equal to the probability r
than swapping of elements between the parents are done otherwise offsprings elements are same as parents element.

III. PSO WITH CROSSOVER
PSO algorithms have been successful in solving a wide variety of problems, their performance is criticized one certain aspects. For example the problem of the loss of diversity after subsequent iterations which lead to premature convergence leading to sub optimal solution [4, 5]. Loss of diversity becomes more prominent for multimodal functions having several optima or noisy functions where the optimum keeps shifting from one position to other. Loss of diversity generally takes place when the balance between the two antagonists processes exploration (searching of the search space) and exploitation(convergence towards the optimum) is disturbed.

IV. MATRIX BASED PSO WITH CROSSOVER
Crossover as we know does the task of recombination of two individuals to generate individuals of the new generation. The new individual carries some of the characteristics of the first parent and the other characteristics from the other parent. The new individual generated in the process may be fitter than the parents or may be weaker. This depends upon the selection of the characteristics by the individual. The selection of good characteristics from both the parents that result in high fitness results in better individuals and vice versa. This selection of particles for crossover is sometimes a topic of concern that is we select two random particles for crossover and then apply crossover to them. Here we have randomly distributed particles from which we select the particles which are the fittest and these are then converged based on their location that is to local optima or to a global optima.

The designed approach first applies a matrix algorithm to the search space region. It divides the entire region into a matrix of desired size depending on the number of particles. If the particles are less, we can divide the space into less no. of regions but if we have large no. of particles we can divide the entire space into more no. of regions. So that we can cover up all our particles efficiently. After dividing into region now we have got a matrix. Now in each entry of the given matrix, we enter a matrix again so that now we have matrix of matrices. From these small matrices, during each iteration that is at each step a matrix is entered in the first place, its local optima is calculated and is stored in a vector i.e. array. When all the sub-matrices have been evaluated like this and local optima which we are taking as the maximum value for all have been calculated, then a global max algorithm rolls over this array and calculate the global maxima of all the locations which here is the globally optimum location of the search space. The concept is similar to the one used for calculating the maximum of two numbers and has been showed in the same way here to make it easy to understand. The algorithm for the approach is –

//Crossover criteria met
for i = 0 to the maximum bound of the number of function evaluation do
for s = 0 to the swarm size do
for d = 0 to the problem dimension do
Update velocity
Update position
end for d
Compute fitness of updated position
If needed, update historical information for Pi and Pg
end for s
if Crossover criteria met then
Select two particles as parent particles from the current swarm(current block or region) for crossover operation.
Apply crossover operation.
New offsprings generated from parents as a result of crossover. Replace the worst parent particle with the
best new offspring if it is better.

If Pg meets problem requirement then
Terminate
end if
end if
end for
i

V. RESULTS

The matrix algorithm above works similar to the program for calculating maximum of two numbers as already stated above. The snapshots taken below show it.

From the standard set of benchmark problems available in the literature, Five important functions are considered to test the efficacy of the proposed method. All the problems are composed of continuous variables and have different degree of complexity and multi-modality. The set of test functions include unimodal and multimodal functions which are scalable (the problem size can be varied as per the user’s choice). In our experiments, problem size for all problems is set to 30. These are minimization problems having a minimum at 0. The problems are listed in Table I.

| Table 2:PSO with Crossover |
|-----------------|---|---|---|---|---|
| PSO             | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Sphere          | 0.849 | 0.840 | 0.834 | 0.815 | 0.90 |
| Griwank         | 0.874 | 0.868 | 0.874 | 0.872 | 0.995 |
| Rosenbrock      | 0.913 | 0.154 | 0.421 | 0.096 | 0.49 |
| Rastrigin       | 0.209 | 0.068 | 0.273 | 0.293 | 0.742 |
| Ackley          | 0.363 | 0.151 | 0.926 | 0.858 | 0.382 |

Table 3: New PSO with Matrix Operator

| Table 3: New PSO with Matrix Operator |
|-----------------|---|---|---|---|---|
| PSO             | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| Sphere          | 0.871 | 0.865 | 0.879 | 0.876 | 0.864 |
| Griwank         | 0.871 | 0.865 | 0.879 | 0.876 | 0.864 |
| Rosenbrock      | 0.09 | 0.46 | 0.57 | 0.54 | 0.49 |
| Rastrigin       | 0.54 | 0.68 | 0.273 | 0.293 | 0.742 |
| Ackley          | 0.898 | 0.886 | 0.838 | 0.848 | 0.898 |

So on, we enter the values for all the regions and so on the process computes the desired result. It has also been applied to various benchmark functions and it has been shown that the above algorithm improves the performance although not drastically but obviously to certain extent.
The above algorithm when merged with the basic PSO Crossover algorithm proves to be better then it.

VI. CONCLUSIONS
The proposed matrix algorithm when merged with the basic crossover algorithm help to efficiently calculate the global optima and thus aids in easy convergence. The result is further proved when we apply the above formula to standard benchmark functions from literature. The results themselves prove the efficiency of above algorithm.

VII. REFERENCES