Improving the Performance of Particle Swarm Optimization (PSO) using Aging Leader and Challengers (ALC-PSO) - Stagnation Removal in PSO Algorithm

Abstract
The PSO algorithm suffers from stagnation problem i.e. it gets stuck in the local optima and gets converged prematurely, hindering its fast convergence behavior. This problem can be overcome by the ALC-PSO, as it limits the number of candidate solutions, taking part in the final process of finding the gbest position, which can remove the problem, as such solutions cause the velocity of the particles to slow down and these particles accumulate before the intended position, causing the swarm to converge prematurely i.e. the problem of premature convergence (stagnation). This paper presents the stagnation problem occurring in the PSO algorithm and some ways to remove this problem. A variant of PSO, called the PSO with Aging Leader with Challengers (ALC-PSO) overcomes this stagnation problem and improves the efficiency of the swarm. The ALC-PSO has been evaluated on several benchmarks and its performance is compared with the performance of PSO. The global best positions in both the algorithms is compared with each other, which shows the better values in case of ALC-PSO. The performance of ALC-PSO is evaluated using the benchmark functions and the results are compared with that of PSO.

Keywords- Particle swarm optimization, Particle swarm optimization with aging leader and challengers, aging, lifespan, challenger, premature convergence, global best (gbest) position, benchmark, performance.

1. Introduction
PSO is a heuristic global optimization method[1]. It has its roots from the Swarm Intelligence[2]. It is an optimization technique based on stochastic behavior of population[6]. It can be an Artificial Intelligence technique, which could find approximate solution with a difficult problems. PSO is a biologically inspired optimization method[4]. PSO uses swarming behaviors observed in bird flocking, fish schooling, bee swarming and socially interactive behavior of humans.

Particle Swarm Optimization is an optimization technique which is based on the social interactions between the bird flocks. When the birds wander in search of food to different places, one of them could find the place where food can be found. That bird becomes the best one out of the whole flock. This mechanism is used for the optimization problems in order to find the best solution out of various available feasible solutions. The best solution is selected out from them and it is the optimal solution. When we use PSO algorithm, the particles are said to be the mass-less and
volume-less matter, which roam here and there in their swarm with their defined velocities and positions, out of these various particles, one best particle position is found which is said to be the optimal solution of the problem. The positions of the particles are compared with each other by:

- **pbest**: comparing its position with its previous best position. It is the particle’s personal best position.
- **lbest**: another best position in the neighborhood of the particle.
- **gbest**: the globally best position of the whole swarm

When this algorithm is used, the particles update their velocities, taking them from one position to another. The velocity and position update rules are:

\[
v(t+1) = (w \cdot v(t)) + (c_1 \cdot r_1 \cdot (p(t) - x(t))) + (c_2 \cdot r_2 \cdot (g(t) - x(t)))
\]

\[
x(t+1) = x(t) + v(t+1)
\]

where, \(v(t+1)\) is the update velocity of the particle at time \(t+1\)

\(v(t)\) is the velocity of the particle at time \(t\)

\(w\) is the inertia weight which balances the exploration-exploitation ratio, it is a real number like -1.73

\(r_1\) and \(r_2\) are the random numbers in range (0,1)

\(c_1\) and \(c_2\) are cognitive weights

\(p(t)\) is the personal best position of particle at time \(t\)

\(x(t)\) is the current position of particle at time \(t\)

\(g(t)\) is the global best position of particle at time \(t\)

\(x(t+1)\) is the position of particle at time \(t+1\)

At every iteration the velocity and position of every particle are updated.

1.1. **Stagnation problem in PSO**

During this process, the particles may de-accelerate and their velocities may come to zero, making the particles stopping at a position, before the final positions are reached. This can hinder the fast converging feature of the algorithm, thereby undergoing premature convergence, called stagnation problem. The particles may get accumulated at an unintended position before reaching the desired position, and cause convergence at that point. This can be an expensive process as lots of search effort is wasted. With this premature convergence, the useful information may not be fully utilized. So, it is required to improve convergence precision and the velocity of convergence to maintain the fast converging feature of PSO.

2. **Related Work Done in PSO**

1. **J. Kennedy and R. C. Eberhart, “Particle swarm optimization,”** A concept for the optimization of nonlinear functions using particle swarm methodology was introduced. The evolution of several paradigms was outlined, and an implementation of one of the paradigms was discussed. Benchmark testing of the paradigm was described, and applications, including nonlinear function optimization and neural network training, are proposed. The relationships between particle swarm optimization and both artificial life and genetic algorithms were described.

2. **Yuhui Shi and Russell Eberhart, “A Modified Particle Swarm Optimizer”,** A new parameter, called inertia weight is introduced into the original particle swarm optimizer. Simulations have been done to demonstrate the significant and effective impact of this new parameter on the particle swarm optimizer.
3. Clerc, M, “The swarm and the queen: towards a deterministic and adaptive particle swarm optimization”- A simple iterative particle swarm optimization algorithm was presented, with one equation and one social parameter. A “no-hope” convergence criterion and a “rehope” method was adopted to re-initialize its position.

4. F. Van Den Berg , A. P.Engelbrecht- “A new locally convergent Particle Swarm Optimizer”- A new locally convergent Particle Swarm Optimizer was developed and its performance was checked on Unimodal functions.

5. Chunming Yang and Dan Simon, “A New Particle Swarm Optimization Technique”- This paper proposed NPSO, here, each particle adjusts its position according to its own previous worst solution and its group’s previous worst to find optimal value. The strategy is to avoid particle’s previous worst solution and its group’s previous worst solution. The results show that NPSO always finds better solutions than PSO.

6. Hui Wang, Yong Liu, “Opposition-based Particle Swarm Algorithm with Cauchy Mutation”- Particle Swarm Optimization (PSO) shows fast search speed. But it could easily fall into local optima. This paper presented an Opposition-based PSO (OPSO) to accelerate the convergence of PSO and to avoid premature convergence.

2.1. Limitations in Related Work done in PSO

From the standard PSO, modifications have been made to improve the performance of the algorithm. Several researchers have worked to remove the limitations in the algorithm but the drawbacks were not able to be removed at all. Some flaws remained there.

1. Classical PSO algorithm(1995)- optimizes the solution by using global best but there is a chance to get trapped in local area. There is no suggestion provided for such a situation. The algorithm is more efficient for small number of particles as it takes the best value found by neighbours into consideration. As the number of particles increases, The gbest value becomes more useful.

2. A Modified Particle Swarm Optimizer (1998)- works better than the original algorithm but only small benchmark function is used for its testing. There arises a difficulty to select the value of inertia weight.

3. The swarm and the queen: Towards deterministic and adaptive particle swarm optimization by Clerc M.(1999)- It could not clear that whether the optimal value is dependent on $\phi$ or not. Thus, there is difficulty to select value of $\phi$. The three methods used did not give constant result.

4. A new locally convergent Particle Swarm Optimizer by F. Van Den Berg and A. P.Engelbrecht (2002)-It was tested for Unimodal functions only. There is not anything about its performance for multimodal functions.

5. A New Particle Swarm Optimization Technique by Chunming Yang and Dan Simon (2005)- It Presented the implementation of PSO and NPSO. Each particle moves to a new position but it does not consider whether the new solution is better than the current solution or not. Some changes can be made to make the algorithm move to a better solution, but it may move to a worse position.
6. **Opposition-based Particle Swarm Algorithm with Cauchy Mutation** - It has a faster convergence on number of functions, but in some functions it fell into local optima that does not guarantee the further convergence of particles.

3. **Overcoming the problem of stagnation with Particle Swarm Optimization with Aging Leader and Challengers (ALC-PSO)**

   As every organism grows older, its lifespan decreases. Aging is a vital process [10], which can bring diversity in nature [3]. This aging mechanism can be applied to the PSO algorithm [5] to prevent it from the stagnation problem. The Aging leader mechanism does not allow those weak particles to participate in the final process and thus, does not let those particles to converge at a point before the destination as it replaces them by best particles to become the leader of the swarm, so that only the best solution is found at the end of the process.

   The aging leader mechanism works on the particles of the swarm. One of the particle is made to be the leader, which can take the whole swarm to the best solution. This leader is selected on the basis of the lifespan of the particle and its leading power. The leading power of the challenger is compared with the previous leader, one with the better leading power is made to lead the whole swarm. The election of the leader from among various available challengers is done based on its leadership performance and lifespan. Based on the leading power of the leader, its lifespan is adjusted. If it's good leading power, it lives longer leading the swarm, and brings all of the members of the swarm towards best position so found but when isn't capable of leading the swarm, new challengers emerge as new leader, claiming the leading position in swarm.

   Whenever the leader of population becomes aged, new challengers come up to lead the population. The new challengers are generated using two parameters i.e. performance and lifespan. The lifespan of the leader is tuned by the lifespan controller according to its leading power and new challengers are generated. Using some function evaluations, the generator continues generating the challengers till the maximum number of predefined evaluations are reached. The best challenger becomes the new leader of the swarm [3].

3.1. **Designing and Working of ALC-PSO**

   The designing of ALC-PSO can be done in three steps:

   1. **Design lifespan controller** - adjusting the lifespan of the leader.
   2. **Generating challengers** - generation of challengers for challenging the position of the current leader.
   3. **Accepting challenger** - deciding whether generated challenger can be accepted as new leader.

   ALC-PSO is different from original PSO as in simple PSO there is no limit on lifespan of leader of the swarm but in ALC-PSO, the leader ages within a limited lifespan. This lifespan depends on the leading power of leader of swarm which can be adjusted accordingly. When lifespan of leader gets exhausted, the leader is replaced by a new particle, which challenges the position of the leader and makes itself becomes the leader. The velocity update rule is changed to:

   \[ v_i = w \ast v_i + c_1 \ast r_1 \ast (pBest_i - x_i) + c_2 \ast r_2 \ast (Leader_i - x_i) \]

   Here leader is a particle with adequate leading power generated by aging mechanism.
1. **Lifespan Controller**

After updating the positions of the particles, the leading power of leader to improve the entire swarm is evaluated. Lifespan \( b \) is adjusted by the lifespan controller. The generated leader checks the gBest and has three cases:

1. \( gBest < 0 \): In this case, the leader can efficiently lead the population, so its lifespan is increased by 2.
2. \( gBest = 0 \): In this case, the leader can satisfactorily lead the population and its performance can be enhanced to some extent, so its lifespan is increased by 1.
3. \( gBest > 0 \): In this case, there is no hope for improvement in performance, so the leader’s lifespan is decreased by 1.

2. **Generation of the Challenger**: New challenger is generated when the lifespan of the old leader gets exhausted. When the performance of particle is greater than the previous leader, the leader is updated and when the best solution of the population is found, it is reported.

3. **Accepting the challenger** - The leading power of newly generated challenger is evaluated, if this challenger has enough leading power, it replaces the old leader and itself becomes the new leader. The leading power of newly generated challenger is evaluated, if this challenger has enough leading power, it replaces the old leader and itself becomes the new leader. Age is initialized to 0 i.e. \( b = 0 \). Lifespan \( t \) is initialized to \( t_0 \) else the old leader remains unchanged and will continue leading the swarm[7]. When this aging mechanism is applied to the PSO algorithm, the algorithm restricts some particles from becoming the leader of the swarm and thus, does not let those weaker particles to enter the process of finding the best solution. The challengers are generated by comparing their leading power. If the leading power of the challenger is better, its lifespan is increased and it becomes the leader of the swarm. Thus, the stagnation problem is removed.

4. **Comparison of PSO and ALC-PSO using benchmark functions**

4.1. **Benchmark Functions**

Benchmark functions play an essential role in validating and comparing the performance of optimization algorithms. The benchmark functions should possess some diverse properties, which can be useful in testing of any new algorithm. The efficiency, reliability and validation of optimization algorithms can be done using a pair of standard benchmarks or benchmark functions. For almost any new optimization, it is required to validate its performance and compare it with other existing algorithms employing a good pair of benchmark functions. Benchmark functions are used to judge characteristics of optimization algorithms, such as: velocity of convergence, precision, robustness, general performance. Every time a new algorithm is usually to be evaluated, the benchmark functions are employed to check on its reliability, efficiency and validity.

4.2. **Comparing the gbest values of ALC-PSO and PSO using Benchmark Functions**

The PSO and ALC-PSO algorithms are implemented on MATLAB (R2011b). The algorithm gives the convergence point at which all the particles of the swarm get
accumulated, means the optimal solution is found and the optimum point is achieved.
The error between the actual value and the desired value is found in order to compare the gbest values.
Error= difference between actual value and the desired value
Average best-so-far = the average of best values found so far
Iterations = the number of times the process is run
Number of iterations=10
The gbest values of both the algorithms for some benchmark functions are tabulated here, from which it can be clearly seen that the gbest values for the ALC-PSO algorithm is smaller than the PSO algorithm, so the better results are found, preserving the fast convergence feature of the original PSO and overcoming the limitation of stagnation. The graphs of both the algorithms are plotted for better comparisons. The error is found to check the difference between the actual and the desired values.

PSO with Ackley Function

ALC-PSO with Ackley Function
PSO with Griewangk Function

ALC-PSO with Griewangk Function

PSO with Rastrigin Function
ALC-PSO with Rastrigin Function

PSO with Schwefel Function

ALC-PSO with Schwefel Function
Table 1 - gbest values for PSO and ALC-PSO

<table>
<thead>
<tr>
<th>Benchmark function</th>
<th>gbest value for PSO</th>
<th>gbest value for ALC-PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ackley</td>
<td>0.0781</td>
<td>0.0041</td>
</tr>
<tr>
<td>Griewangk</td>
<td>0.1228</td>
<td>0.0948</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>0.9467</td>
<td>0.1284</td>
</tr>
<tr>
<td>Schwefel</td>
<td>1.906</td>
<td>1.2783</td>
</tr>
</tbody>
</table>

5. Conclusion

Particle Swarm Optimization algorithm is a easy and less costly optimization method but it suffers from the stagnation problem, which can limit the algorithm to converge prematurely, in order to remove this problem, the concept of aging leader and challengers is used in the PSO algorithm and this process is named as- ALC-PSO. Generally, PSO is applied on those behaviors, in which there's no leader to lead the population like: bird flocking and bee swarming, but in ALC-PSO, one of many members of the population is made to function as leader to lead the population and bring all of them to the best position in whole swarm. The aging mechanism is applied on the PSO, to ensure that some parameter be set to test the performance of the leader of swarm. In the event, the leader is insufficient to lead the swarm, a new leader is found which can efficiently bring the whole swarm toward a most useful position. The generation of challengers is done with a couple function evaluations. The challengers are evaluated and the best challenger is built to be the leader of the swarm, improving the best position in the swarm and thus, improving the performance of PSO algorithm. Benchmark functions are important in testing or evaluating any algorithm. These functions are well-suited to gauge a new algorithm, by comparing its efficiency with other algorithms and testing its validity using different parameters could be done. The performance of both the algorithms i.e. the PSO algorithm and the ALC-PSO algorithm is checked. The gbest values of both are compared, to show that ALC-PSO works better and leads to increase in the performance of PSO algorithm. Decreasing the search time for searching of the challenger is still a drawback in this algorithm, which can be taken as a future scope.

6. References


Authors

Er. Avneet Kaur