CAM TO LEVER MECHANISM FOR HEAVY LOAD APPLICATIONS BY PARTICLE SWAM OPTIMIZATION PROCESS

VENKANNA BABU MENDI
Reg No:-24917086
Research Scholar
DEPARTMENT OF MECHANICAL ENGINEERING
Shri JJT University
Rajasthan

DR. S. CHAKRADHAR GOUD
JJT/ENG/2K9/308
DEPARTMENT OF MECHANICAL ENGINEERING
Shri JJT University
Rajasthan

Abstract:
The Particle Swarm Optimization (PSO) algorithm, as one of the latest algorithms inspired from the nature, was introduced in the mid-1990s and since then, it has been utilized as an optimization tool in various applications, ranging from biological and medical applications to computer graphics and music composition. In this paper, following a brief introduction to the PSO algorithm, the chronology of its evolution is presented and all major PSO-based methods are comprehensively surveyed. Next, these methods are studied separately, and their important factors and parameters are summarized in a comparative table. In addition, a new taxonomy of PSO-based methods is presented. It is the purpose of this paper is to present an overview of previous and present conditions of the PSO algorithm as well as its opportunities and challenges. Accordingly, the history, various methods, and taxonomy of this algorithm are discussed and its different applications together with an analysis of these applications are evaluated. Optimization is a mathematical technique that concerns the finding of maxima or minima of functions in some feasible region.

INTRODUCTION

Path planning is different from motion planning where dynamics must be considered. Its purpose is to find the optimal path of motion in the least amount of time and to model the environment completely. For the path planning problem of mobile agents, several researchers have proposed many algorithms, which can be classified into two categories, i.e., traditional path planning methods and bionic intelligent algorithm-based methods. The former method includes the A algorithm, Dijkstra, RTT, and artificial potential field method. Bionic intelligent algorithms include differential evolution algorithm, genetic algorithm, ant colony algorithm, artificial fish swarm algorithm, and PSO. PSO algorithm is widely used in the practical application and theoretical research of mobile agent path planning due to its strong searchability, fast convergence speed, and high efficiency. The PSO is a population-based randomization technique to find the optimal value for the foraging of the birds. PSO has the advantages of fast search speed, memory, few parameters, and simple structure and has easier implementation at the stage of validation. Besides, its shortcomings include global search and local search imbalance, low convergence precision, easy falling into optimal local solution, and poor robustness. To obtain better optimization results, a PSO optimization technique is proposed in to converge at the global minimum, and a custom algorithm is used to generate the coordinates of the search space. The coordinate values generated by the custom algorithm are passed to the PSO algorithm, which uses these coordinate values to determine the shortest path between two given end positions. Thus, it is not limited to only finding the optimal value. Still, it can improve the speed of the algorithm. However, the direct transfer of its two-point coordinates is easy to fall into the local optimum, and the obtained optimal value is often considered as the global suboptimal value; the literature in proposed a random disturbance method. The adaptive PSO optimization method introduces a disturbing global update mechanism to the optimal global position, which prevents the algorithm stalling. Besides, a new adaptive strategy is proposed to fine-tune the three control parameters in the algorithm. Moreover, the need for dynamic adjustment and exploration of the three parameters increases the computational complexity of the optimization algorithm as well as low execution efficiency. The work in proposed a modified particle swarm optimization (MPSO) with constraints to jointly obtain a smooth path, but the method does not improve the efficiency of the particle diversity, which makes the particles stagnate and fall into the optimal local solution. The studies in proposed a fusion of chaotic PSO and ant colony algorithm (ACO). The algorithm effectively adjusts the particle swarm optimization parameters. It reduces the number of iterations of the ant colony algorithm, called the chaotic particle swarm algorithm, thereby effectively reduces the search time. However, due to the improved algorithm parameters, its speed and the position updates are proportional, and this can limit the particle global searchability. Moreover, it is a solution that can fall into the optimal local solution. The hybrid genetic particle swarm optimization algorithm (GA-PSO) is proposed in that established a path planning. The mathematical model of the problem proposed a time-first based particle-first iteration mechanism, which makes the evolution process more directional and accelerates the development of path planning problems. However, the hybrid algorithm has too many parameters that need control. This increases the computational complexity and has reduced the execution efficiency of the algorithm. It signifies that it is difficult to improve the diversity of particles due to the easy way of falling into the optimal local solution. Therefore, to combine the advantages of the above-mentioned various improved algorithms and improve their defects, this paper proposed a new method to improve the PSO integration scheme based on improved details, which is used to solve the global path planning of mobile robots in the indoor environment. In the proposed algorithm, the navigation point model is selected as the working area model of the mobile robot, and the uniform distribution, exponential attenuation inertia weight, cubic spline interpolation function, and learning factor of enhanced control are introduced to PSO to improve its performance. Finally, the advancement and effectiveness of the standard test function and obstacle environment are verified in the paper. The results show that, compared with other standard functions, IPSO achieves better optimal results and more iteration steps. Compared with the other four path planning algorithms, IPSO reaches the optimal path length with less than 20 iteration steps and reduces the path length and simulation time by 2.8% and 1.1 seconds, respectively.

Improved PSO (IPSO)

Classical PSO

The fundamental core of the PSO method is to share the information through the individuals in the group so that the movement of the whole group can be transformed from disorder to order in the problem of solution space to obtain the optimal solution of the problem. The answer to each optimization problem is the "particle" speed and position update through (1) and (2). The first term of (1) indicates that the next move of the particle is affected by the magnitude and direction of the last flight speed; the second term means that the following action of the particle originates from its own experience; the third term indicates that the next move of the particle originates from the population for the best companion to learn. That is, the next step of the particle is determined by the experience and the best experience of the companion:

$$V_{id}^{k+1} = \omega V_{id}^{k} + c_1 r_1 \left(P_{id}^{k} - X_{id}^{k}\right) + c_2 r_2 \left(P_{gd}^{k} - X_{id}^{k}\right), \quad (1)$$

$$X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k+1}, \quad (2)$$

where V^k_{id} is the dth component of the flight velocity vector of the particle i of the kth iteration; X^k_{id} indicates the dth element of the position vector of the particle i of the kth iteration; c_1 and c_2 represent the learning factor, which is used to adjust the learned to the maximum step size; r_1 and r_2 are two random functions with a range of value from 0 to 1 and are used to increase the randomness searching; and is the inertia weight to adjust the searchability for the solution space. Although the classical PSO is simple to implement and has few adjustment parameters when it is used in the path planning method, it is prone to poor searchability, falls into the optimal local solution, and has reduced particle diversity, low convergence precision, and low accuracy of path planning. Therefore, this paper systematically improves the classical PSO algorithm by combining various improved methods.

Improvement of PSO Algorithm Integration

We aim to ensure that the PSO algorithm does not lose the randomness of the particles when initializing the population and, at the same time, avoid the problem of excessive concentration of the initial positions of the particles, which is not conducive to global search and postprocessing, especially with the possibility that the search stagnation may occur in the late search. In this paper, continuous uniform random distribution is used to initialize the particles so that the particles are relatively uniformly distributed in the search space to facilitate later search and avoid particles falling into the local optimal solution. The formula for uniform position distribution is shown as follows:

$$x = \frac{x_{\max} - x_{\min}}{n - 1}, \quad (3)$$

$$y = \frac{y_{\max} - y_{\min}}{n - 1}, \quad (4)$$

Where x_{max} and y_{max} are the upper limits of the variables, x_{min} and y_{min} are the lower limits of the variables, and n is the number of the decision variables.

Exponential Decay Inertia Weight

The change of the inertia weight ꞶꞶ affects the position of the particle. The larger the value of ꞶꞶ, the stronger the global searchability and the weaker the local mining ability. Therefore, good results can be obtained when the values of the ꞶꞶ are dynamic (adjustable) compared to the fixed values. The value of ꞶꞶ can vary linearly during the PSO search process, or dynamically based on a measure function of PSO performance. By analyzing and summarizing the linear decreasing inertia weight proposed in, the improved particle swarm optimization algorithm based on the improved inertia weight proposed in, and the enhanced method of fuzzy inertia weight strategy proposed in, this paper presents an improvement in the above-mentioned methods. The fixed, predictable, and iterative step adopted in the previous ways is a global transformation that is not conducive to the particle, such as fixed transformation that can narrow the search range of the particle and affect the diversity of the particle. This paper uses the study in to introduce the inertia weight of exponential decay which is more in line with the characteristics of the exponential function. By adopting the changes of the presteps and poststeps to the search area, its speed can be successfully updated at different periods. This eliminates the premature particles and improves the robustness of the algorithm. Henceforth, the global search and local mining ability are maintained, and this can solve the problems mentioned above to some extent. The expression is as follows:

$$A = it\frac{\ln \omega_{\max} - \ln \omega_{\min}}{MaxIt} - \ln \omega_{\max},$$

$$\omega = \exp(-A),$$

Where ω_{min} is the minimum weight, ω_{max} is the maximum weight, as well as the current number of iterations, Max It is the maximum number of iterations, and it is the current number of iterations.

METHODOLOGY

Hybrid Particle Swam Approach with 2-opt strategy

As previously mentioned, PSO simulates the action of birds flocking during food hunt and is then used in problems of optimisation, in particular TSP. Each solution to the problem of optimization is taken in the search field bird and named a particle. Each particle has a fitness value, which is calculated by its destination and its distance and has a pace. Both particles aim for the best locations and positions of the best particles in the swarm in the solution space. Initially a random particle category is formed and then repetitive searches are used to locate the optimal solutions. A particle follows two best methods in each iteration to renew itself. First of all, for particle the best location is sought and secondly, for the swarm. PSO is most generally used for solving the TSP dilemma and is also integrated in local optima. A local 2-opt scan technique is used to eliminate any repetitive particle motions. This is explained in following algorithm.

Parameters are initialised Begin population particles Assess population particles

To perform local scans for citizens Percentage Facilitated (not termination) Pick the right individual (pbest) Find the greatest in the country (gbest) Updating community particles

Assess population particles

Stop local looking for citizens and return worldwide (gbest)

Table: Values of the Regression Line Method Constants

Parameters	1994	1995	1996	1997	1998	1999
Pavg	12617.84	12817.07	12881.84	13052.52	13193.94	13343.33
R	0.001326	0.001397	0.001574	0.001427	0.001226	0.001834
Tavg	48.9959	49.1262	49.4532	48.61903	48.7989	51.54045
Havg	37.86579	37.93797	38.4562	38.22395	37.14249	41.31872
Davg	3.145205	3.139726	3.136986	3.144809	3.145205	3.145205
σ_p	18.21926	18.84424	17.93716	17.46683	16.76527	15.74825
σ_T	19.00326	20.14151	18.69683	19.3137	17.80588	16.74203
σ_H	1.124209	1.124904	1.128887	1.122698	1.124209	1.124209
σ_D	1425.365	1615.975	1507.773	1360.784	1317.05	1369.757

Table: Weight Factors used in the Euclidean Norm

Weights	**Values**
w_1	0.0204
w_2	0.0121
w_3	0.7999

Table Values of Parameters of Membership Functions of Fuzzy Variables

FUZZY INFERENCE SYSTEM	**(a1,a2)**	**(a3,a4)**	**(a5,a6)**
Parameters of membership functions	(-10000,10000)	(-40, 40)	(-35, 35)

of input variables			
Parameters of membership functions of output variables	(b1,b2,b3)	(-0.3,-0.25,-0.2)	
	(b3,b4,b5)	(-0.2,-0.15,-0.10)	
	(b4,b5,b6)	(-0.15,-0.10,-0.05)	
	(b5,b6,b7)	(-0.10,-0.05,0)	
	(b6,b7,b8)	(-0.05,0,0.05)	
	(b7,b8,b9)	(0,0.05,0.1)	
	(b8,b9,b10)	(0.05,0.1,0.15)	
	(b9,b10,b11)	(0.1,0.15,0.2)	

	(b10,b11,b12)	(0.15,0.2,0.25)
	(b11,b12,b13)	(0.2,0.25,0.3)

The daily load forecasting results of two weeks of August'97 are shown in the Table IX. This was an initial attempt for performing the Short Term Load Forecasting based on the Similar Day Approach based on the Euclidean Distance Norm corrected by the Fuzzy Inference System. We name this technique as Fuzzy-1Y since only previous month of one Year (1Y) is considered for the selection of similar days. The results clearly indicate that the technique is not that much successful as the MAPE results accuracy is low and MAPE is below 3.0% for only 6 out of the 14 days as shown in the simulation results.

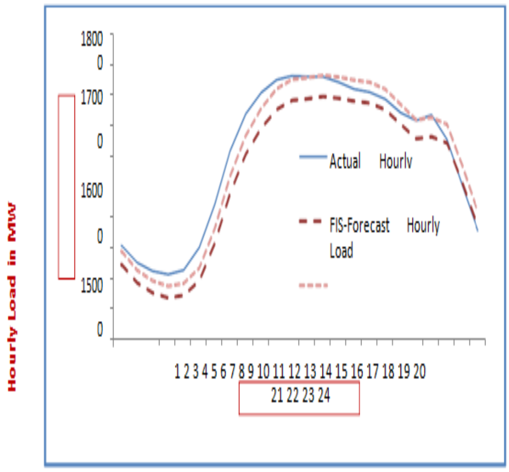

Figure: Actual Hourly Load and Forecasted Hourly Load Comparative graph of 4th Aug'97(Monday)

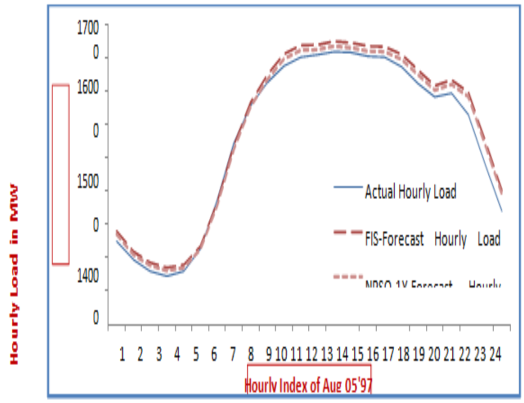

Figure: Actual Hourly Load and Forecasted Hourly Load Comparative graph of 5th Aug'97(Tuesday)

Conclusion

The particle swarm optimization (PSO) and cubic spline interpolation are combined to solve the minimum of five test functions and robot path planning problems. In view of the problems of the classical particle swarm optimization algorithm, such as poor searchability, low convergence accuracy, easy falling into local optimal solution, poor robustness, and poor path smoothness, the classical particle swarm optimization algorithm is improved from the following aspects: The uniform initialization strategy is adopted for the population to improve the later searchability of our IPSO algorithm. This prevents the algorithm from falling into the local optimal solutions, because the random initialization of the particles is not evenly distributed, which is not conducive to the searchability at the later stages. The introduction of the exponential decay for inertia weight makes the particles grow up in the early stage of the search and is beneficial to the global search. The particle step size in the later stages of the search is small in the local development, and the optimization accuracy is high. Experimental results show that the method increases the disturbance and diversity of particles to a certain extent. The use of sine and cosine function can control the independent variable as the inertia weight. The learning factor makes the three variables become one variable; this reduces the parameters of the improved algorithm and thus reduces the complexity of our algorithm.

References:

1. *Carlisle, A., Dozier, G., "Adapting Particle Swarm Optimization to Dynamic Environments", Proceedings of the International Conference on Artificial Intelligence, pp. 429-434, 2000.*
2. *Charbel Farhat, "implementation Aspects of concurrent finite element computations in Parallel Computations and their Impact on computational mechanics", ASME New York, 1987.*
3. *Chen, S. M., Chien, C., Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques, In Expert Systems with Applications, Volume 38, Issue 12, 2011, Pages 14439-14450.*
4. *Clerc, M., "The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization", Proceedings of the IEEE Congress on Evolutionary Computation, vol. 3, pp. 1951-1957, Jul. 1999.*
5. *Clerc, M., "Think Locally, Act Locally: The Way of Life of Cheap-PSO, an Adaptive PSO", Technical Report, http://clerc.maurice.free.fr/pso/, 2001.*
6. *Clerc, M., Kennedy, J., "The Particle Swarm-Explosion, Stability and Convergence in a Multidimensional Complex Space", IEEE Transactions on Evolutionary Computation, 6(1):pp. 58-73, 2002.*
7. *Conradie, E., Miikkulainen, R., "C. Aldrich, Intelligent process control utilising symbiotic memetic neuroevolution", Proceedings of the IEEE Congress on Evolutionary Computation 1 (2002) 623–628.*
8. *Das, S., Abraham, A., Konar, A., "Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives", Ying Liu et al. (Eds.), Advances of Computational Intelligence in Industrial Systems, Studies in Computational Intelligence, Springer Verlag, Germany, 2008, pp. 1–38.*
9. *Dong, X. et al, "A hybrid discrete PSO-SA algorithm to find optimal elimination orderings for Bayesian networks," 2010 2nd International Conference on Industrial and Information Systems, Dalian, 2010, pp. 510-513.*
10. *Eberhart, R.C., Shi, Y., "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization", Proceedings of the IEEE Congress on Evolutionary Computation, vol. 1, pp. 84-88, Jul. 2000.*
11. *Eberhart, R.C., Shi, Y., "Particle Swarm Optimization: Developments, Applications and Resources", Proceedings of IEEE Congress on Evolutionary Computation, Volume 1, IEEE Press, May 2001, pp. 27-30.*