IMPROVING THE PARTICLE SWARM OPTIMIZATION ALGORITHM USING THE SIMPLEX METHOD AT LATE STAGE

Fang Wang, and Yuhui Qiu
Intelligent Software and Software Engineering Laboratory, Southwest-China Normal University, Chongqing, 400715, China

Abstract: This article proposes a hybrid Particle Swarm Optimization (PSO) based on the Nonlinear Simplex Method (NSM). At late stage of PSO running, when the promising regions of solutions have been located, the algorithm isolates particles which are very close to the extrema and applies the NSM to them to enhance the local exploitation. Experimental results on several benchmark functions demonstrate that this approach is very effective and efficient, especially for multimodal function optimizations. It yields better solution qualities and success rates compared to other methods taken from the literature.

Key words: particle swarm optimization, simplex method, multimodal function

1. INTRODUCTION

From the beginning of 90's, a new research field called Swarm Intelligence (SI) arose [1, 2]. These new optimization techniques focus on analogies of swarm behavior of natural creatures, which suggest that the main ideas of intelligent individual's socio-cognition can be effectively applied to develop efficient algorithms for optimization tasks. Ant Colony Optimization (ACO) is the most well known SI algorithm and is mainly used for combinatorial optimization tasks [3]. The Particle Swarm Optimization algorithm is another SI technique, which is mainly used for continuous optimization tasks and has been originally proposed by R.C. Eberhart and J. Kennedy [4] based on the analogy of swarm of bird and fish school. PSO exhibits good performance in solving hard optimization problems and engineering applications, and compares favorably to other optimization algorithms [5, 6].

Since its introduction, numerous variations of the basic PSO algorithm have been developed in the literature [7] to improve its overall performance. Hybrid PSO algorithms with determinate methods, such as the nonlinear simplex method, are proved to be superior to the original techniques and have many advantages over other techniques, because these hybrid procedures can perform the exploration with PSO and the exploitation with determinate methods [8]. Generating initial swarm by the NSM might improve, but is not satisfying in multimodal function optimization [9]. Applying the NSM as an operator to the swarm during the optimization may increase the computational complex considerably.

In this paper, the Nonlinear Simplex Method is adopted at late stage of PSO algorithm when particles fly quite near to the extrema. Experimental results on several famous test functions demonstrate that this is a very promising way to increase the convergence speed and the success rate significantly. We briefly introduce some background knowledge of the PSO algorithm in section 2. In section 3 the proposed algorithm is described, and experimental design and correlative results are given. The paper closes with some conclusions and ideas for further work in Section 4

2. THE PARTICLE SWARM OPTIMIZATION ALGORITHM

In the original PSO formulae, particle i is denoted as $Xi=(x_{i1},x_{i2},...,x_{iD})$, which represents a potential solution to a problem in D-dimensional space. Each particle maintains a memory of its previous best position, $Pi=(p_{i1},p_{i2},...,p_{iD})$, and a velocity along each dimension, represented as $Vi=(v_{i1},v_{i2},...,v_{iD})$. At each iteration, the P vector of the particle with the best fitness in the local neighborhood, designated g, and the P vector of the current particle are combined to adjust the velocity along each dimension, and that velocity is then used to compute a new position for the particle.

The evolutionary equations of the swarm are [6]:

$$v_{id} = w*v_{id}+c1*rand()*(p_{id}-x_{id})+c2*Rand()*(p_{gd}-x_{id}) \quad (1)$$

$$x_{id} = x_{id}+v_{id} \quad (2)$$

Constants c1 and c2 determine the relative influence of the social and cognition components (learning rates), which often both are set to the same value to give each component equal weight. A constant, V_{max}, was used to limit the velocities of the particles. The parameter w, which was introduced as an inertia factor, can dynamically adjust the velocity over time, gradually focusing the PSO into a local search.

Maurice Clerc has derived a constriction coefficient K, a modification of the PSO that runs without V_{max}, reducing some undesirable explosive feedback effects [10]. The constriction factor is computed as:

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \quad \varphi = c_1 + c_2, \quad \varphi > 4 \tag{3}$$

With the constriction factor K, the PSO formula for computing the new velocity is:

$$v_{id} = K*(v_{id}+c1*rand()*(p_{id}-x_{id})+c2*Rand()*(p_{gd}-x_{id})) \tag{4}$$

Carlisle and Doziert investigated the influence of different parameters in PSO, selected c1=2.8, c2=1.3, population size as 30, and proposed the Canonical PSO [11, 12].

3. THE PROPOSED ALGORITHM AND EXPERIMENTAL RESULTS

The basic simplex method was presented by Spendley et al in 1962, which is an efficient sequential optimization method for function minimization tasks, and then improved by Nelder and Mead, to what is called the Nonlinear Simplex Method (NSM) [13]. It needs only values and not derivatives of the objective function. In general, the NSM is considered as the best method if the figure of merit is “get something to work quickly”.

At late stage of PSO running, promising regions of solutions have been located. Applying the NSM operator for many steps to enhance exploitation search at this stage is capable of improving the solution quality and convergence rate.

We propose a hybrid Nonlinear Simplex Method PSO (NSMPSO), which isolates a particle and apply the NSM to it when it reaches quite close to the extrema (within the diversion radius). If the particle "lands" within a specified precision of a goal solution (error goal) during the NSM running, a PSO process is considered to be successful, otherwise it may be laid back to the swarm and start the next PSO iteration.

The diversion radius is computed as:

$$DRadius = ErrorGoal + \delta \tag{5}$$

$$\delta = \begin{cases} 100 * ErrorGoal, & if \quad ErrorGoal <= 10^{-4} \\ 0.01 * ErrorGoal, & otherwise \end{cases} \tag{6}$$

Table 1. The rate of success, mean function evaluations for each test function

Test Functions	Rate of success			Mean function evaluations		
	NSMPSO	NS-PSO	CPSO	NSMPSO	NS-PSO	CPSO
$Rastrigin_2$	1	0.84	1	2851.1	5573.3	3181.8
Levy $No.3_2$	1	0.83	1	2532	4827	2853.3
$Schaffer_2$	0.74	0.57	0.645	7872.6	9906.8	9119.7
$Rosenbrock_2$	1	0.845	0.97	7372.5	9421.8	8450.3
$Griewank_2$	0.805	0.685	0.735	8353.7	9397.6	8985.9
Levy $No.8_3$	1	0.995	1	2644	2045	2327.6
$Freudenstein_2$	0.98	0.52	0.975	5342.5	9321.1	4349.6
$Goldstern_2$	1	0.94	1	2714.1	3601.2	2955.4
$Sphere_{10}$	1	1	1	8138.3	5723.4	6306.8
$Rosenbrock_{10}$	0.855	0.945	0.84	5480.9	3552	5461.6
$Rastrigin_{10}$	0.965	0.83	0.96	5379.8	5929.5	5094.3
$Griewank_{10}$	0.885	0.8	0.845	8969.5	7084.1	7332.3
$Sphere_{30}$	1	1	1	78671	13789	15448
$Rosenbrock_{30}$	0.805	0.94	0.795	16783	10697	16576
$Rastrigin_{30}$	0.99	1	1	9100.4	3475.5	8511.3
$Griewank_{30}$	1	1	0.995	19388	8999.1	10801

*Note:

The subscript of each test function denotes its dimension.

NSMPSO: the proposed algorithm.

NS-PSO: another NSM hybrid PSO proposed by Parsopoulos and Vrahatis [9].

CPSO: the Canonical PSO, Carlisle A [12].

In a NSM process, an initial simplex is consists of the isolated particle i and other D vertices randomly generated with the mean of X_i and standard deviation of 0.02 times X_{max}.

The benchmark functions [14] on which the proposed algorithm has been tested and compared to other methods in the literature, as well as the equation of each one and the corresponding parameters are listed in Table 1. For all test functions 200 experiments have been done and the maximum number of PSO iterations is set to be 500, swarm size is 60 for 30-dimension functions and 30 for others. Parameters used in the NSM are: α=1.0, γ=2.0, $\beta^+=\beta^-$=0.5. An each time when the NSM operator is applied, it searches 50 steps. Programming environment is: Matlab 7.0, PentiumIV 2.8GHz CPU, 512M RAM, Windows2000 Professional OS.

Table 2. The average optima and total CPU time for each test function

Test Functions	Average optima			Total CPU time		
	NSMPSO	NS-PSO	CPSO	NSMPSO	NS-PSO	CPSO
$Rastrigin_2$	4.8541e-9	0.42783	4.9856e-9	14.266	27.156	14.906
Levy $No.3_2$	-176.54	-163.17	-176.54	12.578	22.688	12.813
$Schaffer_2$	-0.99755	-0.99334	-0.99664	29.188	35.484	31.047
$Rosenbrock_2$	4.9915e-9	0.76162	0.002373	30.578	36.422	31.781
$Griewank_2$	1.0012	1.003	1.0016	44.016	49.141	44.813
Levy $No.8_3$	6.4267e-9	5.8974e-9	6.0348e-9	14.656	10.891	11.188
$Freudenstein_2$	0.97969	80.237	1.2246	16.078	31.875	14.516
$Goldstern_2$	3	2240.1	3	9.8906	12.859	9.8125
$Sphere_{10}$	8.8273e-9	8.2146e-9	8.2073e-9	38.828	23.125	24.656
$Rosenbrock_{10}$	31.146	461.95	20.688	22.922	15.266	22.547
$Rastrigin_{10}$	9.6321	10.14	9.7158	29.547	32.625	27.641
$Griewank_{10}$	9.0987	9.1038	9.1003	55.156	42.25	43.156
$Sphere_{30}$	9.2039e-5	9.3474e-5	9.2407e-5	494.11	67.594	68.609
$Rosenbrock_{30}$	2009.3	100.63	2856.7	81.484	56.281	80.047
$Rastrigin_{30}$	97.418	95.818	97.752	65.109	27.574	60.75
$Griewank_{30}$	29.098	29.094	29.094	156.08	71.609	81.547

In table 1 and 2, the rate of success, mean function evaluations, average optima and total CPU time for each test function are given. From the table we can see that the overall performance of NSMPSO algorithm is superior to other published algorithms in terms of success rate, solution quality and convergence speed, especially on multimodal functions such as Levy No.3, Schaffer, Rosenbrock and Griewank. As to high dimension function optimizing, NSMPSO operates appreciably inferior to NS-PSO due to its computational expense, but is still equal to the Canonical PSO algorithm.

4. CONCLUSIONS AND FUTURE WORK

A new hybrid Particle Swarm Optimization algorithm is proposed in this paper, which applies the Nonlinear Simplex Method at late stage of PSO running when the most promising regions of solutions are fixed. We implement wide variety of experiments on well-known benchmark functions to test the proposed algorithm. The results compared to other 3 published methods indicate that this method is reliable and efficient, especially for continuous multimodal function optimizations.

Future work may focus on accelerating the convergence for high dimension problems, developing practical applications of this hybrid approach in neuro-fuzzy network optimization, and extending the approach to constrained multi-objective optimization.

REFERENCE

[1] Bonabeau E, Dorigo M, andTheraulaz G. Swarm Intelligence: From Natural to Artificial Systems. Oxford Press, 1999

[2] Kennedy J, and Eberhart R C. Swarm Intelligence. Kaufmann Publishers, Morgan, 2001

[3] Deneubourg J L, Goss S, Franks N, Sendova-Franks A, Detrain C, and Chr´etien L. The dynamics of collective sorting: Robot-like ants and ant-like robots. Proceedings of the First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 1, 1991, MIT Press, Cambridge, MA: 356-365

[4] Kennedy J, and Eberhart R C. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, 1995, Piscataway, NJ: 1942-1948

[5] Shi Y, and Eberhart R C. Parameter selection in particle swarm optimization. Evolutionary Programming VII: Proceedings of the 7^{th} Annual Conference on Evolutionary Programming, 1998, New York: 591-600

[6] Shi Y, and Eberhart R C. Empirical study of particle swarm optimization. Proceedings of the IEEE Congress on Evolutionary Computation, 1999, Piscataway, NJ: 1945-1950

[7] Parsopoulos K E, and Vrahatis M N. Recent approaches to global optimization problems through particle swarm optimization. Natural Computing, 2002, Vol. 1: 235-306

[8] Shu-Kai S. Fan, Yun-Chia Liang, and Erwie Zahara. Hybrid Simplex Search and Particle Swarm Optimization for the Global Optimization of Multimodal Functions. Engineering Optimization, 2004, 36(4): 401-418

[9] Parsopoulos K E, and Vrahatis M N. Initializing the particle swarm optimizer using the nonlinear simplex Method. Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, 2002, WSEAS Press: 216-221

[10] Clerc M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Proceedings of the IEEE Congress on Evolutionary Computation, 1999: 1951-1957

[11] Carlisle A, and Dozier G. An off-the-shelf PSO. Proceedings of the Workshop on Particle Swarm Optimization, 2001, Indianapolis

[12] Clerc M, and Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58-73

[13] Nelder J, and Mead R. A simplex method for function minimization. Computer Journal, 1965, Vol. 7: 308-313

[14] Levy A, Montalvo A, Gomez S, et al. Topics in Global Optimization. Springer-Verlag, New York: 1981