INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 9, Number 2, Pages 371–377

A HYBRID PARTICLE SWARM OPTIMIZATION ALGORITHM BASED ON SPACE TRANSFORMATION SEARCH AND A MODIFIED VELOCITY MODEL

SONG YU, ZHIJIAN WU, HUI WANG, ZHANGXIN CHEN, AND HE ZHONG

Abstract. Particle Swarm Optimization (PSO) has shown its fast search speed in many complicated optimization and search problems. However, PSO often easily falls into local optima because the particles would quickly get closer to the best particle. Under these circumstances, the best particle could hardly be improved. This paper proposes a new hybrid PSO (HPSO) to solve this problem by combining space transformation search (STS) with a new modified velocity model. Experimental studies on 8 benchmark functions demonstrate that the HPSO holds good performance in solving both unimodal and multimodal functions optimization problems.

Key words. Space Transformation Search (STS), evolutionary algorithm, Particle Swarm Optimization (PSO), optimization.

1. Introduction

Particle swarm optimizer (PSO), which was firstly introduced by Kenedy and Eberhart in 1995[1,2], emulates the flocking behavior of birds to solve optimization problems. In PSO, each potential solution is considered as a particle. All particles have their own fitness values and velocities. These particles fly through the D-dimensional problem space by learning from the historical information of all the particles. A potential solution is represented by a particle that adjusts its position and velocity according to equation (1) and (2):

$$\upsilon_{id}^{(t+1)} = \omega\upsilon_{id}^{(t)} + c_1r_1(p_{id}^t - x_{id}^t) + c_2r_2(p_{gd}^t - x_{id}^t), \tag{1}$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + \nu_{id}^{(t+1)}, \tag{2}$$

where t is the time index, i is the particle index, and d is the dimension index. p_i is the individual best position. p_g is the known global best position. ω is the inertia weight described in [3]. c_1 and c_2 are the acceleration rates of the cognitive and social parts, respectively. r_1 and r_2 are random values different for each particle i as well as for each dimension d.The position of each particle is also updated in each iteration by adding the velocity vector to the position vector.

One problem found in the standard PSO is that it could easily fall into local optima in many optimization problems. One reason for PSO to converge to local optima is that particles in PSO can quickly converge to the best position once the best position has no change. When all particles become similar, there is a little hope to find a better position to replace the best position found so far. In this paper, a new hybrid PSO algorithm called HPSO is proposed. It avoids premature convergences and allows STS-PSO [4] to continue searching for the global optima

Received by the editors December 11, 2009 and, in revised form, March 2, 2010.
2000 *Mathematics Subject Classification.* 35R35, 49J40, 60G40.
This research was supported by Foundation CMG Funds and by the National Basic Research Program of China(973 program, No.: 2007CB310801).

by applying space transformation-based learning and to break away from local optimal with a new disturbing factor and a convergence monitor. Our HPSO has been tested on both unimodal and multi-modal function optimization problems. Comparison has been conducted among HPSO, standard PSO and STS-PSO. The rest of this paper is organized as follows: Section 2 presents the new HPSO algorithm. Section 3 describs the benchmark continuous optimization problems used in the experiments, and gives the experimental settings. Section 4 presents and discusses the experimental results. Finally, Section 5 concludes with a summary.

2. HPSO ALGORITHM

2.1. Space Transformation Search (STS). Many evolutionary optimization methods start with some initial solutions, called individuals, in an initial population, and try to improve them toward some optima solution(s). The process of searching terminates when some predefined conditions are satisfied. In some cases, the searching easily stagnates, when the population falls into local optima. If the stagnation takes places too early, the premature convergence of search is caused. Under these circumstances, the current search space hardly contains the global optimum. So it is difficult for the current population to achieve better solutions. However, Space transformation search, based on opposition learning method[5], originally introduced by Hui Wang [4], has proven to be an effective method to cope with lots of optimization problems. When evaluating a solution x to a given problem, we can guess the transformed solution of x to get a better solution x'. By doing this, the distance of x from optima solution can be reduced. For instance, if x is -10 and the optimum solution is 30, then the transformed solution is 40. But the distance of x' from the optimum solution is only 20. So the transformed solution x' is closer to the optimum solution. The new transformed solution X^* in the transformed space S can be calculated as follows:

$$x^* = k(a+b) - x, \tag{3}$$

where $x \in R$ within an interval of $[a, b]$ and k can be set as 0,0.5,1 or a random number within $[0, 1]$.

To be more specific, we put it in an optimization problem, Let $X = (x_1, x_2, x_n)$ be a solution in an n-dimensional space. Assume $f(X)$ is a fitness function which is used to evaluate the solution's fitness. According to the definition of the STS, $X^* = (x_1^*, x_2^*, x_n^*)$is the corresponding solution of X in the transformed search space. If $f(X^*)$ is better than $f(X)$, then update X with X^*; otherwise keep the current solution X. Hence, the current solution and its transformed solution are evaluated simultaneously in order to continue with the fitter one. The interval boundaries $[a_j(t), b_j(t)]$ is dynamically updated according to the size of current search space. The new dynamic STS model is defined by

$$X_{ij}^* = k[a_j(t) + b_j(t)] - X_{ij}, \tag{4}$$

$$a_j(t) = min(X_{ij}(t)), b_j(t) = max(X_{ij}(t)) \tag{5}$$
$$i = 1, 2, ..., PopSize, j = 1, 2, ...n$$

2.2. Modified Velocity Model. In the PSO, particles are attracted to their corresponding previous best particles $pbest_i$ and the global best particle $gbest$. With the movement of particles, particles are close to $pbest_i$ and $gbest$, and then $pbest_i - X_i$ and $gbest - X_i$ becomes small. According to the updating equation of velocity, the velocity of each particle become small. Once the $pbest_i$ or $gbest$ fall into local minima, all the particles will quickly converge to the positions of them.

The cognitive part and social part of each particle will be near to 0 because of $X_i = pbest_i = gbest$. As a result, the velocity of each particle tends to 0, and the updating equation of position is invalid. Finally, all the particles will be stagnate and hardly escape from local optima.

In order to avoid this situation, this paper proposes a new modified velocity model to perturb the position of particles by monitoring each $pbest_i$ and $gbest$. If the $pbest_i$ or $gbest$ has no changes in a predefined number of generations, it is considered to be trapped into local optima. To help it escape from local optima, we conduct a disturbance to the particle to help the trapped particle jump to another position accordingly, if the $Monitor\_pbest_i > T_1$, the cognitive part of PSO turns to be:

$$c_1 r_1 (p^t_{id} - d_1 x^t_{id}), \tag{6}$$

where $Monitor\_pbest_i$ records the number of times the $pbest$ did not change, and the T_1 is the predefined threshold, and d_1 is a random number within [0,1]. If the $Monitor\_gbest > T_2$, the social part of PSO turns to be:

$$c_2 r_2 (p^t_{gd} - d_2 x^t_{id}), \tag{7}$$

where $Monitor\_gbest$ records the number of times the $gbest$ did not change, and the T_2 is a predefined threshold, and d_2 is a random number within [0,1]; Accordingly the equation (1) can be modified to be:

$$v^{(t+1)}_{id} = \omega v^{(t)}_{id} + c_1 r_1 (p^t_{id} - d_1 x^t_{id}) + c_2 r_2 (p^t_{gd} - d_2 x^t_{id}) \tag{8}$$

3. Numerical Experiments

3.1. Test Functions. A comprehensive set of benchmark functions [6], including 8 different global optimization problems, have been chosen in our experimental studies. According to the properties, they are divided into two classes: unimodal functions ($f_1 - f_4$), multimodal functions ($f_5 - f_8$). All the functions used in this paper are to be minimized. The description of the benchmark functions and their global optimum(s) are listed in Table 2.

3.2. Experiment Setup. There are three variant PSO algorithms including the proposed HPSO used in the following experiments. The algorithms and parameters settings are listed below: The standard PSO (PSO);A space transformation search PSO (STS-PSO); Our Hybrid PSO (HPSO);

For PSO, STS-PSO and HPSO, $w = 0.72984$, $c_1 = c_2 = 1.49618$, and the maximum velocity V_{max} is set to the half range of the search space on each dimension. For all algorithms, the population size is set to 40 and the maximum number of evaluations is set to 200,000. The accuracy of functions $f_1 - f_4, f_7, f_8$ is set to 1e-15 and that of functions f_5, f_6 is set to 0. If the fitness value of the best fitness found by all particles so far (best fitness) reaches to the fixed accuracy, the current population is considered to obtain the global optimum, and then the algorithm is terminated. The probability of STS ps is set to 0.25. All the experiments are conducted 50 times with different random seeds, and the average results throughout the optimization runs are recorded.

TABLE 1. The main steps of HPSO.

Begin
n = dimensional size;
P = current population;
TP = the transformed population of P;
t = the generation index;
$[a_j(t), b_j(t)]$ = the interval boundaries of the j_{th} dimension in current population;
ps = the probability of STS;
$best\_fitness$ = the fitness value of the best particle found by all particles so far;
accuracy = fixed accuracy level;
MAX_{NE}=the maximum number of evaluation;
while($best_fitness > accuracy\&\&NEMAX_{NE}$);
if ($rand(0,1) < ps$);
update the dynamic interval boundaries $[a_j(t), b_j(t)]$incurrent population according to equation 5;
for $i = 1$ **to** $PopSize$
Calculate the transformed particle TP_i of P_i according to equation 4;
The velocity of TP_i keeps the same with P_i;
Calculate the fitness value of particle TP_i;
for end
select $PopSize$ fittest particles in P and TP as a new population;
Update *pbest*,*gbest* in the new population if needed;
else
for $i = 1$ **to** $PopSize$
if ($Monitor\_pbest_i >= T_1$)
$d_1 = random[0, 1]$;
else $d_1 = 1$;
if ($Monitor\_gbest >= T_2$), $d_2 = random[0, 1]$; **else** $d_2 = 1$;
Calculate the velocity of particle $P(i)$ according to equation 8;
Update the position of particles $P(i)$ according to equation 2;
Calculate the fitness value of particle $P(i)$;
update *pbest* if needed
if (*pbest* changed), $Monitor\_pbest_i = 0$;
else $Monitor\_pbest_i + +$;
for end
Update *gbest* if needed;
if (*gbest* changed), $Monitor\_gbest$=0;
else $Monitor\_gbest + +$;
while end
End

TABLE 2. minimum values of the function, and X Rn is these search space.

Test Function	Dim	X	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$f_2(x) = \sum_{i=1}^{n} \|x_i\| + \prod_{i=1}^{n} x_i$	30	[-10, 10]	0
$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i})$	30	[-100,100]	0
$f_4(x) = max\|x_i\|$	30	[-100, 100]	0
$f_5(x) = \sum_{i=1}^{n} (x_i + 0.5)$	30	[-100, 100]	0
$f_6(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi\ x_i) + 10)$	30	[-5.12,5.12]	0
$f_7(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30	[-32, 32]	0
$f_8(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\frac{x_i}{\sqrt{i}} + 1$	30	[-600, 600]	0

3.3. Experimental Results. Table 3 shows the comparison among PSO,STS-PSO, HPSO for function f_1 to f_8, where "Mean" indicates the mean best function values found in the last generation, and "NFC" stands for the average number of function calls over 50 trials.The convergence characteristics in terms of the best fitness value of the median run of each algorithm for each algorithm for each test function are presented in Figure 1. From the result, it is obvious that HPSO performs better than standard PSO and STS-PSO. The significant improvement achieved by HPSO can be contributed to the space transformation search and the disturbing factor. The space transformation search method adds the changing

TABLE 3. The comparison results among PSO, STS-PSO and HPSO.

Function	**PSO**		**STS-PSO**		**HPSO**	
	Mean	**NFC**	**Mean**	**NFC**	**Mean**	**NFC**
f_1	9.23e-16	39644	8.81e-16	57069.3	3.52e-16	22600
f_2	8.48e-16	105972	9.16e-16	90714.7	8.44e-16	39708
f_3	9.05e-16	44100	8.99e-16	57853.3	6.57e-16	23800
f_4	3.23e-06	200000	1.45e-04	200000	7.97e-16	48694
f_5	1.15	123560	0	15228	0	5210
f_6	46.3153	200000	56.746	200000	0	22200
f_7	1.41334	200000	9.59e-13	200000	9.56e-13	200000
f_8	2.3e-2	200000	9.34e-16	46400	7.66e-16	24200

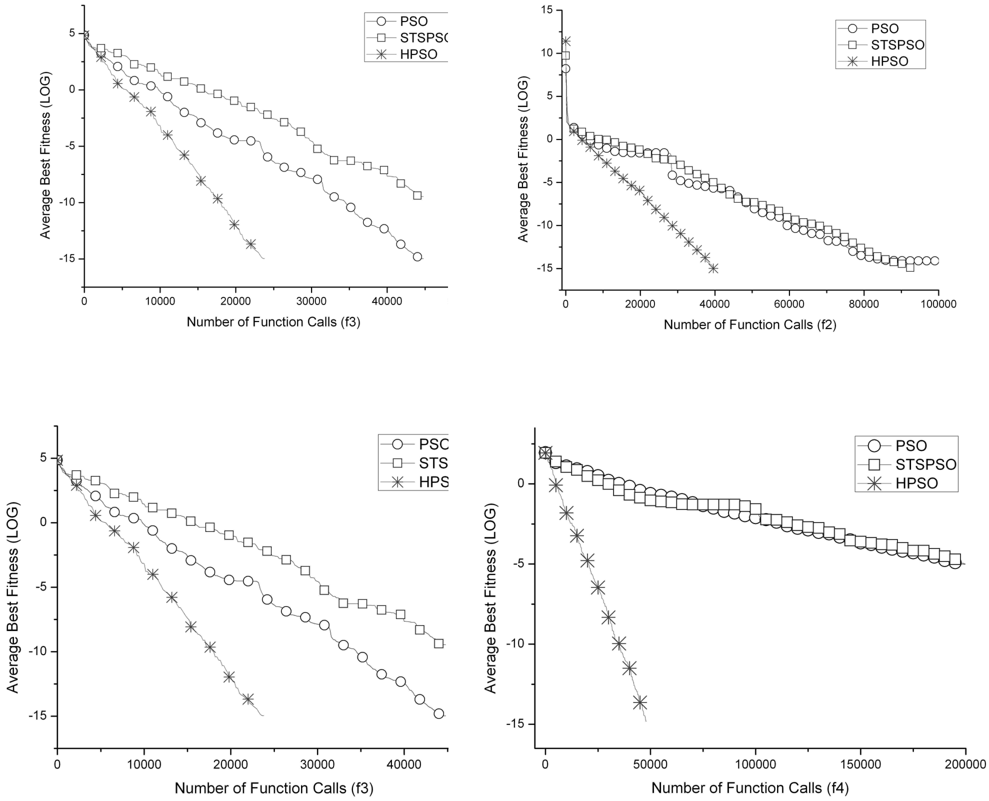

FIGURE 1. Performance comparision among PSO, STS-PSO and HPSO on f_1-f_8. The horizontal axis is the average number of funtion calls and the vertical axis is the average best fitness value(log) over 50 trials.

probability of particles, and the modified velocity model proposed in this paper improves the accuracy of convergence. Therefore, HPSO gets better solutions than the standard PSO and STS-PSO.

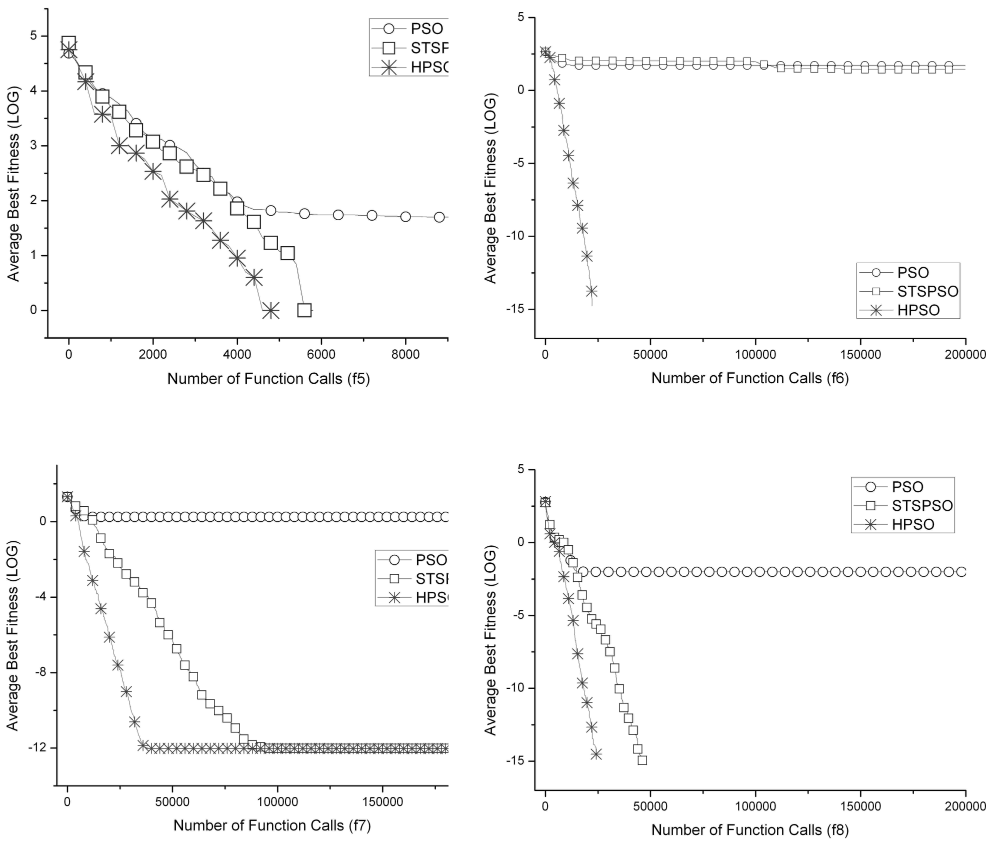

Figure 1. Performance comparison among PSO, STS-PSO and HPSO on f_1-f_8. The horizontal axis is the average number of function calls and the vertical axis is the average best fitness value(log) over 50 trials(con't).

4. Conclusion

The idea of HPSO is to improve PSO based on space transformation search method and a new velocity model with convergence monitor to help avoid local optima and accelerate the convergence of PSO. The new proposed velocity model is to monitor the changes of fitness values of each $pbest_i$ and $gbest$. If a $pbest_i$ or $gbest$ has no improvements in a predefined generations, it is considered to fall into local minima and at the same time we present some disturbances to these particles to break away the local optima. By combining these methods , HPSO is able to find better solutions than other improved PSO. HPSO has been compared with the standard PSO, STS-PSO on both 4 unimodal functions and multimodal functions. The results have shown that HPSO has faster convergence rate on those simple unimodal functions and superior global search ability on those multimodal functions compared to other PSO.

However, according to no free lunch theory[7], in some cases, this hybrid PSO can still not avoid premature convergence, in some more complex problems. This will be an important work to continue. Besides the 8 multimodal functions, more test functions will be selected in further work.

References

[1] J. Kennedy, and R. C. Eberhart.: Particle swarm optimization. *In: Proceedings of IEEE International Conference on Neural Networks* (1995) 1942-1948.

[2] R. C. Eberhart and Y. Shi.: Comparison between genetic algorithms and particle swarm optimization. *In: Proceedings of the 7th Annual Conference on Evolutionary Programming* (1998) 69-73.

[3] Y. Shi and R. C. Eberhart.: A modified particle swarm optimization, *In: Proceedings of IEEE Congress Evolutionary Computation* (1998) 69-73.

[4] H. Wang, Z. J. Wu, Y. Liu.: Space Transformation Search: A New Evolutionary Technique. *Genetic and Evolutionary Computation* (2009) (in press).

[5] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama.: Opposition-Based differential evolution,*In: Proceedings of IEEE Congress Evolutionary Computation*, Vol.12 (2008) 64-79.

[6] X. Yao, Y. Liu and G. Lin.: Evolutionary programming made faster, *IEEE Transaction on Evolutionary Computation*, Vol. 3 (1999) 82-102.

[7] D. H. Wolpert, and W. G. Macready.: No free lunch theorems for optimization. *IEEE Transaction on Evolutionary Computation,* Vol.1 (1997) 67-82.

Center for Computational Geoscienes and Mathematics, Faculty of Science, Xi'an Jiaotong University, Xi'an,710049, P.R. China and Department of Chemical and Petroleum Engineering, Schulich School of Engineering, University of Calgary, Calgary 2500, Canada

E-mail: zhachen@ucalgary.ca and soyu@ucalgary.ca

State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, P.R. China

E-mail: zjwu9551@sina.com and wanghui_cug@yahoo.com.cn

Department of Chemical and Petroleum Engineering, Schulich School of Engineering, University of Calgary, Calgary 2500, Canada

E-mail: zhachen@ucalgary.ca and hzhong@ucalgary.ca

URL: http://schulich.ucalgary.ca/chemical/JohnChen