(7) Function f_7

$$\min f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

s.t.

$$g_1(x) = 92 - 85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \geq 0$$
$$g_2(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \geq 0$$
$$g_3(x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 110 \geq 0$$
$$g_4(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 90 \geq 0$$
$$g_5(x) = 25 - 9.300961 - 0.0047062x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 \geq 0$$
$$g_6(x) = -20 + 9.300961 + 0.0047062x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \geq 0$$

with

$$78 \leq x_1 \leq 102,$$
$$33 \leq x_2 \leq 45,$$
$$27 \leq x_i \leq 45 (i = 3, 4, 5)$$

The optimal solution (20 times average) obtained in this article:

$$x^*: \begin{matrix} 78.00000000003647 & 33.00000583946766 \\ 29.98645470860217 & 44.99967834081389 \\ 36.77005736212263 & \end{matrix}$$

$$f_7(x^*): -30657.22185557836$$

In the particles iterative process, the f_7 function value evolution curve is shown in Fig.7.

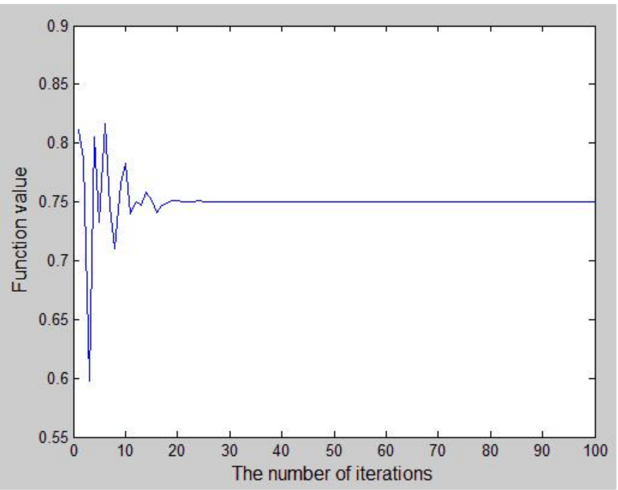

Fig.7 the evolution curve of the f_7 function value

(8) Function f_8

$$\min f(x) = x_1^2 + (x_2 - 1)^2$$

s.t.

$$h_1(x) = x_2 - x_1^2 = 0$$
$$-1 \leq x_1 \leq 1,$$
$$-1 \leq x_2 \leq 1$$

The optimal solution (20 times average) obtained in this article:

$$x^*: \begin{matrix} 0.70710677997036 & 0.49999999806761 \\ -0.70710677997036 & 0.49999999806761 \end{matrix}$$

$$f_8(x^*): 0.75000000021244$$

In the particles iterative process, the f_8 function value evolution curve is shown in Fig.8.

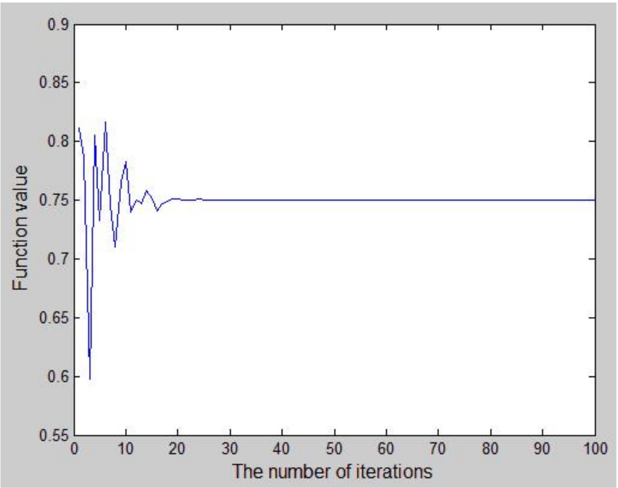

Fig.8 the evolution curve of the f_8 function value

5.2 Experiment Analysis

Numerical experiments are processing in MATLAB 7.0.1. In the calculation, set learning factors $c_1 = c_2 = 0.7$ for f_6 function, $c_1 = c_2 = 0.6$ for f_7 function, $c_1 = c_2 = 1.0$ for the remaining functions. Population size: f_1 ,f_2 ,f_3 ,f_4 ,f_5 ,f_6 ,f_7 ,f_8 for 50 particles, f_4 f_5 for 20 particles. Experiment iteration time will change with the scope and complexity of the problem. Set multiplier vector $\omega = 1, \upsilon = 1$, and $\beta = 0.7$ to measure the speed of convergence, and set constant magnification factor $\alpha = 2$, and the initial penalty factor σ settings: function f_4 , f_6 and f_7 are set to 200, the remaining functions are set as 20. For each test function, the algorithms are run 20 times, taking the optimal value, worst value, mean, mean square error. The results are shown in Table 1.

F	Optimal value
f_1	-6961.81387558016
f_2	1.17712434446771
f_3	-147.6669538913704
f_4	9761.567601684035
f_5	-0.09582504140719
f_6	-14.99905931197651
f_7	-30668.74234793204
f_8	0.74999999910799
Worst value Mean	Mean
-6960.57538397414	-6961.690026419559
1.17712434446771	1.17712434446771
-145.7188592406307	-147.3463377232924
10605.74154900046	10056.67707387459
-0.09582499088190	-0.09582503370020
-14.43006360512358	-14.76663551960394
-30625.85294512852	-30657.22185557836
0.75000000347674	0.75000000021244
square error	Iterations
0.38119967310441	200
0	150
0.67731369342482	300
286.8661296805540	100
1.302403541283220e-008	30
0.18495823750212	150
15.71720094738370	100
9.708100746526781e-010	100

Table 1 The operation results of the algorithm

Comparing a new algorithm with the Ref. [1,2,4,8,10], deterministic global optimization method were used for solving constrained optimization problems in Ref.[1,8]; in Ref.[2], the method of bi-objective fitness was used to deal with the constraints; in Ref.[4], external point method was used to handle the constraints, then the problem is solved by the adaptive velocity particle swarm optimization algorithm. The comparison of new algorithm and other papers is shown in Table 2-9.

The functions in this paper are common test functions for constraint optimization problems, in which f_5 is maximization problem that the objective function may be transformed into minimization problem by negation, and f_8 is the equality constrained optimization problem. It is difficult to achieve global optimization when using evolutionary algorithm for these test functions. It can be seen from Table 2-9 that the new algorithm on these constrained optimization problems basically achieved the optimal value, especially for f_4 function. The optimal solution found by the new algorithm is better than the solutions in other papers. Momentum particle swarm algorithm in the Ref. [5] will cause difficulty in the problem solving because the particles follow the best value of their own history and the best value of group. In many cases at the early stage, the particles fitness which violates the constraints more is better than the particles fitness which violates the constraints less, so that the particles may fly out of the feasible region.

Let IVN denote the Independent variable number, the following Table.2 -9 shows the results comparing the present paper with Ref [1-10].

Table 2. Compared results for f_1

F	IVN	Reference	Optimal value
f_1	2	[2]	-6961.81388
f_1	2	[4]	-6961.8138665
f_1	2	[10]	-6961.81388
f_1	2	This paper	-6961.81387558016

Table 3. Compared results for f_2

F	IVN	Reference	Optimal value
f_2	2	[4]	1.177643
f_2	2	[8]	1.177124327
f_2	2	This paper	1.17712434446771

Table 4. Compared results for f_3

F	IVN	Reference	Optimal value
f_3	3	[1]	-83.249728406
f_3	3	[4]	-147.6667
f_3	3	[8]	-83.249728406
f_3	3	This paper	-147.6669538913704

Table 5. Compared results for f_4

F	IVN	Reference	Optimal value
f_4	5	[1]	10122.493176362
f_4	5	[4]	10122.49323
f_4	5	[8]	10122.38112168
f_4	5	This paper	9761.567601684035

Table 6. Compared results for f_5

F	IVN	Reference	Optimal value
f_5	2	[2]	-0.095825
f_5	2	[4]	-0.0958250
f_5	2	[10]	-0.095825
f_5	2	This paper	-0.09582504140719

Table 7. Compared results for f_6

F	IVN	Reference	Optimal value
f_6	13	[2]	-15
f_6	13	[4]	-14.9999999
f_6	13	[10]	-15
f_6	13	This paper	-14.99905931197651

Table 8. Compared results for f_7

F	IVN	Reference	Optimal value
f_7	5	[2]	-30655.539
f_7	5	[4]	-30665.53677
f_7	5	[10]	-30655.539
f_7	5	This paper	-30668.74234793204

Table 9. Compared results for f_8

F	IVN	Reference	Optimal value
f_8	2	[2]	0.749
f_8	2	[4]	0.7500000
f_8	2	[10]	0.749
f_8	2	This paper	0.74999999910799

6 Conclusion

This paper proposes a hybrid particle swarm optimization algorithm based on the multiplier penalty function to solve constrained optimization problems. During the search process, this method sets the particles position of the previous generation as the optimum of particle individual history and the optimum of group of the previous generation as the optimum of group, which makes each flight of the particle be affected only by the flight of the previous generation particle. This is conducive to trace changes of constraint violations extent, and makes all of the particles gradually approaching the feasible region, and eventually finds the optimal solution in the feasible region. The particle swarm optimization algorithm has no requirement for the optimization function. The numerical experiments show that the new algorithm has the ability to find an optimal value under constraint condition. The parameter adjustment of the new algorithm is more difficult for the optimization problem of different levels of complexity, especially the difference in the number of iterations and the number of particles is large, but the solution speed of the particle swarm algorithm is affected by the number of iterations and the size of the particle swarm.

Acknowledgements: The research was supported by NSFC (No.11071282), and Supported by the Open Project Program of Key Laboratory of Intelligent Computing and Information Processing of Ministry of Education, Xiangtan University, China (No.2011ICIP07). Many thanks for Prof. Sheng Baohuai for his helpful suggestions.

References:

[1] P. P. Shen, K. C. Zhang, Global optimization of signomial geometric programming using linear relaxation, *Appl. Math. Comput.* No.150, 2004, pp. 99-114.

[2] H. Y. Lu, W. Q. Chen, Self-adaptive velocity particle swarm optimization for solving constrained optimization problems, *Journal of Global Optimization.* Vol.41, No.3, 2008, pp. 427-445.

[3] C. Gong, Z. L. Wang, *Proficient in MATLAB optimization.* Beijing: publishing house of electronics industry, 2010.

[4] Y. L. Gao, H. R. Li, Hybrid Particle Swarm Algorithm of Nonlinear Constraint Optimization Problems, *Mathematica Numerica Sinica.* Vol.32, No.2, 2010, pp. 135-146.

[5] R. X. Ma, Y. Liu , Z. Qin, X. Wang, Momentum Particle Swarm Optimizer for Constrained Optimization, *Journal of System Simulation.* Vol.22, No.11, 2010, pp. 2485-2488.

[6] K. E. Parsopoulos, M. N. Vrahatis, Particle Swarm Optimization Method for Constrained Optimization Problems, *Intelligent Technologies-Theory and Applications: New Trends in Intelligent Technologies, USA: IEEE.* No.12, 2002, pp. 214-220.

[7] X. H. Hu, R. C. Eberhart, Y. H. Shi, Engineering optimization with particle swarm, *Proceedings of the 2003 IEEE Swarm Intelligence Symposium, USA: IEEE.* 2003, pp. 53-57.

[8] P. P. Shen, H. W. Jiao, A new rectangle branch-and-pruning approach for generalized geometric programming, *Applied Mathematics and Computation.* No.183, 2006, pp. 1027-1038.

[9] Y. H. Shi, R. C. Eberhart, Empiricalstudy of particle swarm optimization, *Proc of Congress on Computational Intelligence, Washington DC, USA.* 1999, pp. 1945-1950.

[10] H. Y. Lu, W. Q. Chen, Dynamic-objective particle swarm optimization for solving constrained optimization problem, *J. Comb. Optim.* No.12, 2006, pp. 409-419.

[11] G. T. Pulido, C. A. C. Coello, A constraint-handling mechanism for particle swarm optimization, *IEEE 2004 Congress on Evolutionary Computation.USA: IEEE.* 2004, pp. 1396-1403.