An Effective Adaptive Multi-objective Particle Swarm for Multimodal Constrained Function Optimization

Yongquan Zhou
College of Mathematics and Computer Science Guangxi University for Nationalities, Nanning 530006, China
Email: yongquanzhou@126.com

Shengyu Pei
College of Mathematics and Computer Science Guangxi University for Nationalities, Nanning 530006, China
Email:sunback_007@163.com

***Abstract* — This paper presents a novel adaptive multi-objective particle swarm optimization algorithm and with adaptive multi-objective particle swarm algorithm for solving objective constrained optimization problems, in which Pareto non-dominated ranking, tournament selection, crowding distance method were introduced, simultaneously the rate of crowding distance changing were integrated into the algorithm. Finally, ten standard functions are used to test the performance of the algorithm, experimental results show that the proposed approach is an efficient and achieve a high-quality performance.**

***Index Terms*—Particle Swarm Optimization Algorithm; Adaptability Multi-objective Optimization; Constrained optimization; Test functions**

I. INTRODUCTION

In recent years, using particle swarm algorithm (PSO) [1] for solving numerical optimization problems has received increasing attention and wide applications. To our knowledge, there is little published research work on constrained optimization so far. However, many practical problems in scientific research and engineering applications involve a number of constraints that the decision solutions need to satisfy. Therefore, solving objective constrained optimization problems based on particle swarm optimization has great value in real-world applications.

Generally, a constrained optimization problem (COP) [2] can be described as follows:

$$\begin{aligned} &\min y = f(x) \\ &subject\ to\ \ g(x) = (g_1(x),...,g_l(x)) \le 0, \\ &\quad h(x) = (h_{l+1}(x),...,h_m(x)) = 0, \\ &\quad x = (x_1, x_2,..., x_n) \in X \subseteq R^n, \\ &\quad X = \{(x_1, x_2,..., x_n) \mid l_i \le x_i \le u_i\}, \end{aligned} \tag{1}$$

$$l = (l_1, l_2,..., l_n), u = (u_1, u_2,..., u_n).$$

Where $x = [x_1, x_2,..., x_n]^T$ denotes the decision solution vector, l is the number of inequality constraints and m is the number of equality constraints, y is objective function. In a common practice, an equality constraint $h_j(x) = 0$ can be replaced by a couple of inequality constraints.

$$\begin{cases} h_j - \delta \le 0 \\ -h_j - \delta \le 0 \end{cases} \tag{2}$$

In recent years, particle swarm optimization algorithm made great progress in the applied research of multi-objective optimization. Its processing method is similar to multi-objective evolutionary algorithm (MOEA) [3]. Many successful strategies such as external archives based on MOEA are introduced into the multi-objective particle swarm optimization (MOPSO) [4]. On the other hand, MOPSO do not have to conduct fitness assignment so that simplify the algorithm. But it must choose an appropriate global best position for each particle from external archives. However, there is little published research work on constrained optimization based on multi-objective optimization algorithm. In this paper, an effective adaptive multi-objective particle swam algorithm (AMPSO) is proposed for constrained optimization problems by applying MPSO and employing the notion of the rate of crowding distance changing. In the AMPSO, Pareto non-dominated ranking, tournament selection, crowding distance method were introduced, simultaneously the rate of crowding distance changing were integrated into the algorithm. Simulation results based on some famous constrained test functions demonstrate the effectiveness and efficiency of the proposed AMPSO.

doi:10.4304/jcp.5.8.1144-1151

II. BASIC CONCEPTS

In many science and engineering disciplines, it is common to encounter a large number of constrained optimization problems. In this paper, we first give the relevant description of multi-objective optimization problems and the basic definitions related to multi-objective optimization. A multi-objective optimization problem (MOP) with n decision variables and m objective functions has the following form:

$$\min y = f(x) = (f_1(x), f_2(x),..., f_m(x)) \quad (3)$$

Where $x = (x_1, x_2,..., x_n) \in X \subset R^n$ is the decision vector, $y \in Y \subset R^m$ is the objective vector.

Definition 1 (Pareto Dominance) [4]**.** A vector $u = (u_1, u_2,..., u_m)$ is said to Pareto dominate another vector $v = (v_1, v_2,..., v_m),$, denoted as $u \prec v$, if $\forall i \in \{1,..., m\}, u_i \le v_i \wedge \exists j \in \{1,..., m\}, u_j < v_j$

Definition 2 (Pareto Optimality) [4]. $x_u \in X$ is said to be Pareto optimal in X if and only if $\neg x_v \in X, v \prec u,$ where

$$u = f(x_u) = (u_1, u_2,..., u_m).$$

Definition 3 (Pareto Optimal Set) [4**]**. For a given $MOPf(x)$, the Pareto optimal set, denoted as ρ^*, is defined as

$$\rho^* = \{x_u \in X \mid \neg x_v \in X, v \prec u\}$$

Definition 4 (Pareto Front)[4][5]. For a given $MOPf(x)$, according to the Pareto optimal set, the Pareto front, denoted as ρf^*, is defined as

$$\rho f^* = \{u = f(x_u) \in \rho^*\}$$

Definition 5 (Crowding Distance) [6] to get an estimate of the density of solutions surrounding a particular solution in the population, we calculate the average distance of two points on either side of this point along each of the objectives. This quantity $i_{distance}$ serves as estimate of the size of the largest cuboids enclosing the point i without including any other point in the population (we call this the crowding distance).

III. ADAPTIVE MULTI-OBJECTIVE PARTICLE SWARM ALGORITHM

The c

A. The Basic PSO Algorithm

PSO is a stochastic global optimization method which is based on simulation of social behavior of bird flocks or fish schools. In contrast to other evolution algorithm, its principle is simple and easy to program.

The core operations of PSO are two updating equations of the velocity and position for each particle. The actual position and velocity of a particle is associated with a particular design vector x and v. Each particle's own best historical position is denoted by vector $pbest$, and the best historical position that the entire swarm has passed is denoted by vector $gbest$. The new velocity and position of each particle for the next generation are calculated as follows:

$$V_{i,j}(t+1) = \omega V_{i,j}(t) + c_1 \times rand() \times [pbest_{i,j} - x_{i,j}(t)] + c_2 \times rand() \times [gbest_{i,j} - x_{i,j}(t)] \quad (4)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + V_{i,j}(t+1), \ j = 1,2,..., d. \quad (5)$$

Where c_1 and c_2 are two positive constants called acceleration coefficients, ω is called the inertia factor, $rand()$ is random numbers uniformly distributed in the range of $[0,1]$.

B. Adaptive Multi-Objective PSO Algorithm

The optimal values of the control parameters regarding convergence rates and diversity are strong problem dependent, especially the inertia factor. It influences the trade-off between global and local exploration abilities. A larger inertia factor facilitates global exploration while a smaller inertia factor tends to facilitate local exploration to fine-tune the current search area. Therefore, an optimal inertia factor is necessary to improve performance of algorithm. There are a lot of published research works on that. Ren Zi-Hui, Wang Jian presented a new adaptive particle swarm optimization algorithm with dynamically changing inertia weight (DCWPSO) [7]. Hence, in this paper, we introduce crowding distance [6].

$$I(d_k) = I(d_k) + \frac{I(k+1).m - I(k-1).m}{f_m^{\max} - f_m^{\min}} \quad (6)$$

Where, $I(k).m$ refers to the m -th objective function value of the i -th individual in the set I.

Furthermore we present the rate of crowding distance changing value.

Definition 6 (The Rate of Crowding Distance Changing Value):

The current rate of crowding distance changing value k is defined as

$$k = \frac{MaxID - MeanID}{MaxID} \quad (7)$$

Where $MaxID$ and $MeanID$ are calculated as follows

$$MeanID = \frac{(\sum_{i=1}^{m}(p_{id} - x_{id})^2)}{m} \quad (8)$$

$$MaxID = \max(p_{id} - x_{id})^2 \quad (9)$$

Where, p_{id} is crowding distance of the optimal position of entire swarm, x_{id} is best historical position of particle.

We first calculate the crowding distance of each individual in the evolution process, then receive corresponding $MaxID$ and $MeanID$. Finally, we can determine what to do in the next step.

The value of the ω is adaptively adjusted by the mapping:

$$\omega = \begin{cases} (a_1 + |r|/2.0)|\ln k|, & |k| \le 1 \\ a_1 a_2 + |r|/2.0, & 0.05 \le |k| \le 1 \\ (a_2 + |r|/2.0)\dfrac{1}{|\ln k|}, & |k| < 0.05 \end{cases} \quad (10)$$

where a_1 =0.3, a_2 =0.2, r is random numbers uniformly distributed in the range of $[0,1]$ .Note that, with the mapping function, not purely with time or with the generation number. Hence, the adaptive inertia factor is expected to be changed efficiently according to the evolutionary states. It not only adjusts efficiently the global search and local search capabilities, but also maintains the diversity of population.

IV. Adaptive Multi-Objective PSO (AMPSO) algorithm

A. Constraints Treatment

Taking advantage of multi-objective optimization techniques to handle constraints is a new idea raised in recent years. Generally, constraints can be regards as one or more objectives. One objective is taking into account the original objective function $f(x)$, and the other one is considering the degree of constraint violation of an individual x, i.e., $G(x)$, where

$$G_j(x) = \begin{cases} \max\{0, g_j(x)\} & 1 \le j \le l \\ |h_j(x)| & l+1 \le j \le m \end{cases},$$

$$G(x) = \sum_{j=1}^{m} G_j(x). \quad (11)$$

In this paper, constraints can be regards as one objective $G(x)$, then $G(x)$ and $f(x)$ constitute two target vectors $F(x)$ = ($f(x)$, $G(x)$). In this way, constraint problems that contain n decision, one target function, l inequality constraints and $m-1$ equality constraints can be regarded as n decision vectors and two target vectors. Therefore, it avoids using penalty function and can get a Pareto optimal set that uniformly distributed and diversity. When $G(x) = 0$, we can get the Pareto optimal solution according to search results.

B. Adaptive Multi-Objective PSO (AMPSO)algorithm

So far, there are many mature approaches published search work. Laumanns presented a unified model for multi-objective evolutionary algorithms with elitism [8] that co-evolution of individual species and elite populations. De.betal presented a fast and elitist multi-objective genetic algorithm: NSGA-II[6]. The elitist strategy is defined as priority individual will not be excluded from the gene pool when poor individuals impact. In this paper, we use fast non-inferior classification for grading the whole population. First introduce the elite search strategy to expand the sample space, which can maintain the priority individual into the next generation, then get partial order results through non-inferior rank. The experiment show that this method can quickly converge to the area of non-inferior solutions.
Pseudo code of two operations as follow:

Pseudo code 1: Non-inferior classification

Initialize $S_p = \varnothing$. // the collection includes all individuals dominated by p .

Initialize $n_p = 0$. //The number of individuals dominated p

For each individual q in P

If p dominated q then

$S_p = S_p \cup \{q\}$

Else if q dominates p then

$n_p = n_p + 1$

If n_p =0 p_{rank} =1, $F_1 = F_1 \cup \{p\}$

Initialize $i = 0$ //Initialize the number of front

While $F_i \neq \varnothing$

$Q = \varnothing$

For each individual p in front F_i ;

For each individual q in S_p ;

$n_q = n_q$ -1

If n_q=0

q_{rank}=i+1, $Q = \varnothing \cup q$

$i = i + 1,\ F_i = Q$

Pseudo code 2: Crowding distance

Initialize $F_i(d_j)$=0. //The j crowding distance of Front F_i

For each objective function m

Sort (F_i, m).

$I(d_1) = \infty$ and $I(d_n) = \infty$

For k=2 to ($n-1$)

$$I(d_k) = I(d_k) + \frac{I(k+1).m - I(k-1).m}{f_m^{\max} - f_m^{\min}}.$$

We introduce a new particle update strategy, which choose the best position of entire swarm based on comparing Pareto dominated sorting with crowding distance. The whole procedure of AMPSO is described as follows:

Step 1. Let t=0, and randomly initialize swarm $P(t)$, evaluate these objectives value for all particles.

Step 2. Repeat until a stopping criterion is satisfied:

Step 2.1. Sorting-based Pareto dominance relationship in accordance with fitness (Pseudo code 1), calculate the crowding distance of each individual (Pseudo code 2), and choose in front of half of the individual through elitist strategy. Select particle's best Position *pbest* and the best position of the entire swarm *gbest*.

Step 2.2. Calculate the rate of crowding distance changing value, and determine w.

Step 2.3. Update the velocities and positions using Eq. (4) and Eq. (5).

Step 2.4. Merging new and old swarms $Q(t)$, Sorting-based Pareto dominance relationship in accordance with fitness (Pseudo code 1), calculate the crowding distance of each individual (Pseudo code 2), and choose in front of half of the individual through elitist strategy.

Step 2.5. Get partial order results through non-inferior rank, and let $t = t + 1$.

Step 3. Output Pareto optimal set.

V. SIMULATION TEST AND ANALYSIS

A. Parameter Setting

In this section, we will carry out numerical simulation based on some well-known constrained problems to investigate the performances of the proposed SMPSO. In simulation test, parameters of AMPSO are set as follows: swarm size $n = 200$, $c1 = c2$ =2.0, $tour\_size = 2$. The statistical simulation results of 30 independent run for each example.

Example 1[9].

$$\min f_1(x) = -4x_1 - 3x_2$$

$$s.t.\begin{cases} 2x_1 + 3x_2 - 6 \le 0 \\ -3x_1 + 2x_2 - 3 \le 0 \\ 2x_1 + x_2 - 4 \le 0 \\ 0 \le x_i \le 2, i = 1,2 \end{cases}$$

The Example 1 is a linear programming problem, the optimal value is $f_1(x^*) = -9.0$.

Example 2[9].

$$\min f_2(x) = -x_1 - x_2$$

$$s.t.\begin{cases} -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 \le 0 \\ -4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x_2 - 36 \le 0 \\ 0 \le x_1 \le 3 \\ 0 \le x_2 \le 4 \end{cases}$$

All constraints of example 2 are non-linear inequality, the optimal value is $f_2(x^*) = -5.5079$.

.

Example 3 [10].

$$\min f_3(x) = 5x_1 + 5x_2 + 5x_3 + 5x_4 - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$$

$$s.t.\begin{cases} 2x_1 + 2x_2 + x_{10} + x_{11} \le 10 \\ 2x_1 + 2x_3 + x_{10} + x_{11} \le 10 \\ 2x_2 + 2x_3 + x_{11} + x_{12} \le 10 \\ -8x_1 + x_{10} \le 0 \\ -8x_2 + x_{11} \le 0 \\ -8x_3 + x_{12} \le 0 \\ -2x_4 - x_5 + x_{10} \le 0 \\ -2x_6 - x_7 + x_{11} \le 0 \\ -2x_8 - x_9 + x_{12} \le 0 \\ 0 \le x_i \le 1, i = 1,...,9 \\ 0 \le x_j \le 100, j = 10,11,12 \\ 0 \le x_{13} \le 1 \end{cases}$$

The number of the decision vector is 13, the number of the constrained is 9, and the optimal value is $f_3(x^*) = -15$.

Example 4[10].

$$\min f_4(x)=5.3578547x_3^2+0.8356891x_1x_5+37.293239x_1-4.792.141$$

$$s.t.\begin{cases}0\le 85.334407+0.0056858x_2x_5+0.0006262x_1x_4-0.0022053x_3x_5\le 92\\ 90\le 80.51249+0.0071317x_2x_5+0.0029955x_1x_2+0.0021813x_3^2\le 110\\ 20\le 9.300961+0.0047026x_3x_5+0.0012547x_1x_3+0.0019085x_3x_4\le 25\\ 78\le x_1\le 102, 33\le x_2\le 45, 27\le x_i\le 45, (i=3,4,5)\end{cases}$$

The optimal value is $f_4(x^*)=-3.665.539$.

Example 5[10].

$$\min f_5(x)=3x_1+0.000001x_1^3+2x_2+0.000002/3x_2^3$$

$$s.t.\begin{cases}1000\sin(-x_3-0.25)+1000\sin(-x_4-0.25)+894.8-x_1=0\\ 1000\sin(x_3-0.25)+1000\sin(x_3-x_4-0.25)+894.8-x_2=0\\ 1000\sin(x_4-0.25)+1000\sin(x_4-x_3-0.25)+1294.8=0\\ x_4-x_3+0.55\ge 0\\ x_3-x_4+0.55\ge 0\\ 0\le x_i\le 1200, i=1,2\\ -0.55\le x_j\le 0.55, j=3,4\end{cases}$$

The optimal value is $f_5(x^*)=5126.4981$.

Example 6 [10].

$$\min f_6(x)=(x_1-10)^3+(x_2-20)^3$$

$$s.t.\begin{cases}(x_1-5)^2+(x_2-5)^2-100\ge 0\\ -(x_1-6)^2-(x_2-5)^2+82.81\ge 0\\ 13\le x_1\le 100\\ 0\le x_2\le 100\end{cases}$$

The optimal value is $f_6(x^*)=-6961.81388$.

Example 7 [10].

$$\min f_7(x)=(x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-10)^2+10x_5^6+7x_6^3+x_7^4-4x_6x_7-10x_6-8x_7$$

$$s.t.\begin{cases}127-2x_1^2-3x_2^4-x_3-4x_4^2-5x_5\ge 0\\ 282-7x_1-3x_2-10x_3^2-x_4+x_5\ge 0\\ 196-23x1-x22-6x62+8x7\ge 0\\ -4x_1^2-x_2^2+3x_1x_2-2x_3^2-5x_6+11x_7\ge 0\\ -10\le x_i\le 10, i=1,\ldots,7\end{cases}$$

The optimal value is $f_7(x^*)=680.600573$

Example 8[10].

$$\min f_8(x)=x_1^2+(x_2-1)^2$$

$$s.t.\begin{cases}x_2-x_1^2=0\\ -1\le x_i\le 1, (i=1,2)\end{cases}$$

The optimal value is $f_8(x^*)=0.75$.

Example 9[10].

$$\max f_9(x)=(100-(x_1-5)^2-(x_2-5)^2-(x_3-5)^2)/100$$

$$s.t.\begin{cases}(x_1-p)^2+(x_2-q)^2+(x_3-r)^2-0.0625\le 0\\ 0\le x_i\le 10, (i=1,2,3),\ p,q,r=1,\ldots,9\end{cases}$$

Feasible region is compose of 9^3 discrete sphere, vector (x_1,x_2,x_3) is feasible if and only if p,q,r meet these constraints, the optimal value is $f_9(x^*)=1$.

Example 10[11].
The welded beam design problem.

$$\min f_{10}(x)=1.10471x_1^2x_2+0.04811x_3x_4(14.0-x_2)$$

$$g_1(x)=\tau(x)-13000\le 0,$$
$$g_2(x)=\sigma(x)-30000\le 0,$$
$$g_3(x)=x_1-x_4\le 0,$$
$$g_4(x)=0.10471x_1^2+0.04811x_3x_4(14.0+x_2)-5.0\le 0,$$
$$g_5(x)=0.125-x_1\le 0,$$
$$g_6(x)=\delta(x)-0.25\le 0,$$
$$g_7(x)=6000-Pc(x)\le 0,$$
$$\tau(x)=\sqrt{(\tau')_2+2\tau'\tau''\frac{x_2}{2R}+(\tau'')^2},$$
$$\tau'=\frac{6000}{\sqrt{2}x_1x_2},\tau''=\frac{MR}{J},$$
$$M=6000(14+\frac{x_2}{2}),R=\sqrt{\frac{x_2^2}{4}+(\frac{x_1+x_3}{2})^2},$$
$$J=2\left\{\sqrt{2}x_1x_2[\frac{x_2^2}{12}+(\frac{x_1+x_3}{2})^2]\right\},$$
$$\sigma(x)=\frac{504000}{x_4x_3^2},\delta(x)=\frac{2.1952}{x_3^3x_4},$$
$$Pc(x)=64746.022(1-0.0282346x_3)x_3x_4^3$$
$$0.1\le x_1\le 2, 0.1\le x_2\le 10, 0.1\le x_3\le 10, 0.1\le x_4\le 2.$$

Example 10 is a engineering constrained problem, the optimal value is $f_{10}(x^*)=1.724852$.

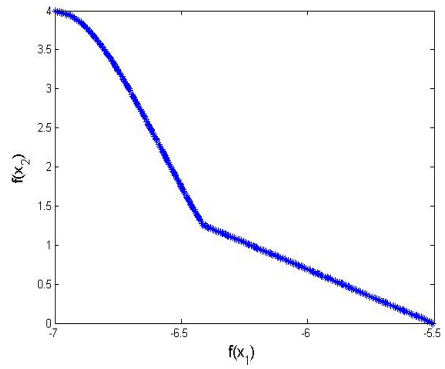

Figure 1. Non-dominated solutions with AMPSO on example 1.

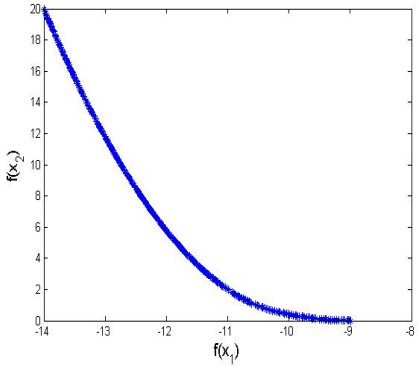

Figure 2. Non-dominated solutions with AMPSO on example 2

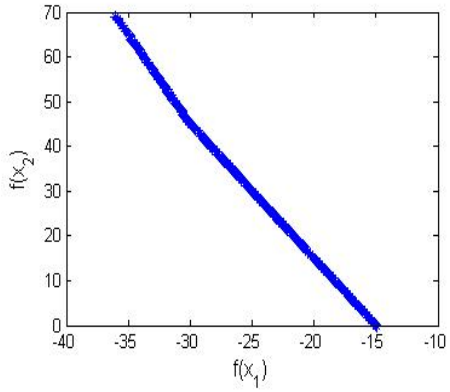

Figure 3. Non-dominated solutions with AMPSO on example 3

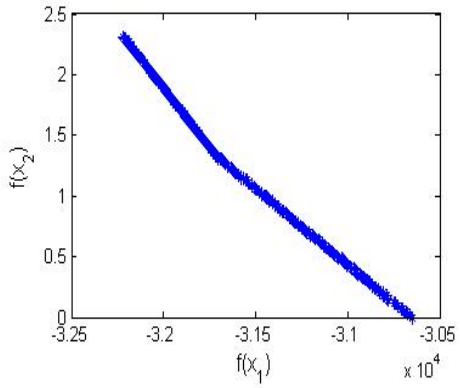

Figure 4. Non-dominated solutions with AMPSO on example 4.

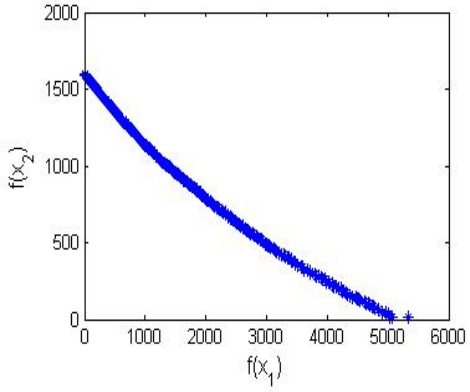

Figure 5. Non-dominated solutions with AMPSO on example 5.

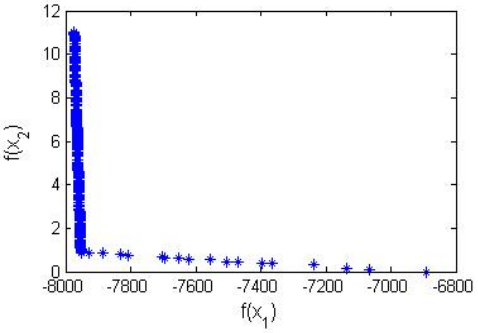

Figure 6. Non-dominated solutions with AMPSO on example 6.

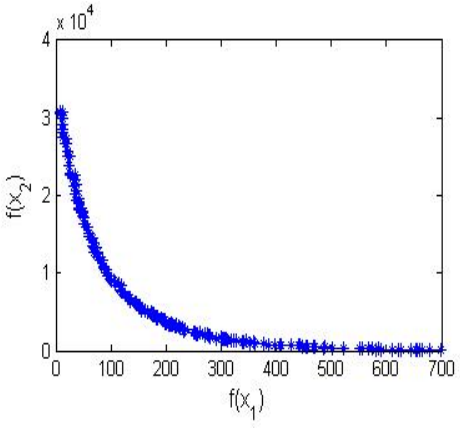

Figure 7. Non-dominated solutions with AMPSO on example 7

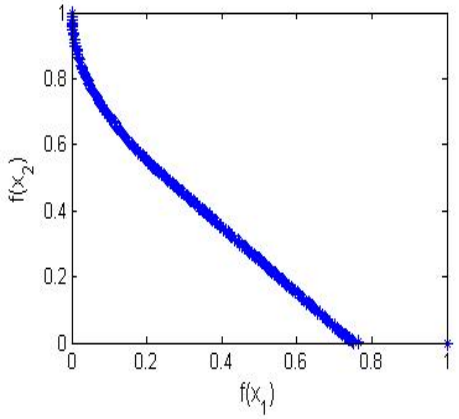

Figure 8. Non-dominated solutions with AMPSO on example 8.

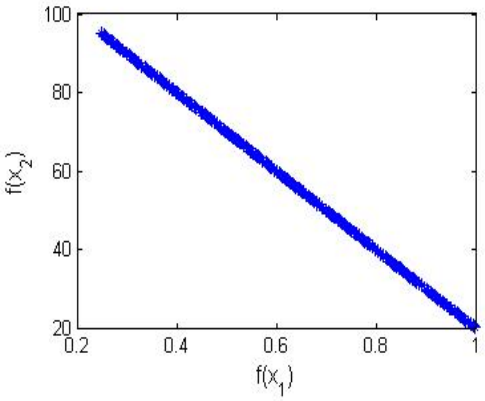

Figure 9. Non-dominated solutions with AMPSO on example 9.

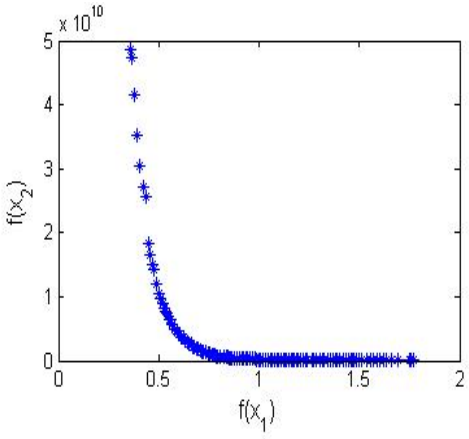

Figure 10. Non-dominated solutions with AMPSO on example 10.

TABLE I. STATISTICAL RESULTS OF AMPSO AND OTHER METHODS FOR EXAMPLE 1-2

Function/ Optimal Value	Status	Methods		
		BPSO	PSO-CA	AMPSO
Example 1/ -9	Best	-8.999700	-8.999900	-9
	Mean	-8.999500	-8.999800	-9
Example 2/ -5.5079	Best	-5.507000	-5.507800	-5.508013
	Mean	-5.506800	-5.507500	-5.508013

TABLE II. STATISTICAL RESULTS OF AMPSO AND OTHER METHODS FOR EXAMPLE 3-9

Function/ Optimal Value	Status	Methods							
		SAFF	SMES	RY	IS-PAES	FSA	IRY	CW	AMPSO
Example 3/ -15	Best	-15.000	-15.000	-15.000	-15.000	-14.999	-15.000	-15.000	-15.000
	Mean	-15.000	-15.000	-15.000	-14.494	-14.993	-15.000	-15.000	-15.000
	Worst	-15.000	-15.000	-15.000	-12.446	-14.980	-15.000	-15.000	-15.000
	St.dev	0	0	0.0E+00	9.3E-01	4.8E-03	1.3E-13	1.3E-14	0
Example 4/ -30665.539	Best	-30665.50	-30665.539	-30665.539	-30665.539	-30665.538	-30665.539	-30665.539	-30665.539
	Mean	-30665.20	-30665.539	-30665.539	-30665.539	-30665.467	-30665.539	-30665.539	-30665.539
	Worst	-30663.30	-30665.539	-30665.539	-30665.539	-30664.688	-30665.539	-30665.539	-30665.539
	St.dev	4.9E-01	0	2.0E-05	0	1.7E-01	2.2E-11	8.0E-12	2.3E-11
Example 5/ 5126.4981	Best	5126.989	5126.599	5126.497	NA	5126.4981	5126.4981	5126.4981	5126.4981
	Mean	5432.080	5174.492	5128.881	NA	5126.4981	5126.4981	5126.4981	5126.4981
	Worst	6089.430	5304.167	5142.472	NA	5126.4981	5126.4981	5126.4981	5126.4981
	St.dev	3.9E+03	5.0E+01	3.5E+00	NA	0	6.2E-12	1.513E-12	8.0E-12
Example 6/ -6961.81388	Best	-6961.800	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814	-6961.814
	Mean	-6961.800	-6961.284	-6875.940	-6961.813	-6961.814	-6961.814	-6961.814	-6961.813
	Worst	-6961.800	-6952.482	-6350.262	-6961.810	-6961.814	-6961.814	-6961.814	-6961.811
	St.dev	0	1.9E+00	1.6E+02	8.5E-05	0	6.4E-12	1.8E-12	6.2E-05
Example 7/ 680.600573	Best	680.64	680.632	680.630	680.630	680.630	680.630	680.630	680.630
	Mean	680.72	680.643	680.656	680.631	680.636	680.630	680.630	680.630
	Worst	680.87	680.719	680.763	680.634	680.698	680.630	680.630	680.630
	St.dev	5.9E-02	1.6E-02	3.4E-02	8.1E-04	1.5E-02	4.6E-13	4.7E-13	5.7E-12
Example 8/ 0.75	Best	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750
	Mean	0.750	0.75	0.750	0.750	0.750	0.750	0.750	0.750
	Worst	0.750	0.75	0.750	0.751	0.750	0.750	0.750	0.750
	St.dev	0	1.5E-04	8.0E-05	2.6E-04	0	9.6E-10	0.0E-00	1.3E-12
Example 9/ 1	Best	-1.000	-1.000	-1.000	NA	-1.000	-1.000	-1.000	-1.000
	Mean	-1.000	-1.000	-1.000	NA	-1.000	-1.000	-1.000	-1.000
	Worst	-1.000	-1.000	-1.000	NA	-1.000	-1.000	-1.000	-1.000
	St.dev	0	0	0.0E+00	NA	0	9.6E-10	0.0E+00	0

We analyze from the example 8 and decide to choose $\max(G(x))$ as the second objective functions, Fig.9 shows that it can converge quickly the Pareto-optimal set in this way.

These figures and tables show that AMPSO can search local Pareto-optimal set, and the convergence and ability to find a diverse set of solutions. Especially, table 1 show that the best solutions obtained by AMPSO are better than the above mentioned approaches. For example 1, a typical solution found by our algorithm is: $x = (1.5,1.0)$ with $f(x) = 9$. For example 2, a typical solution found by our algorithm is: x = (2.329520, 3.178493) with $f_2(x)$ =-5.508013. For example 10, a typical solution found by our algorithm is: $x = (0.812500, 0.437500, 42.09845, 1.766366 + 002)$ with $f_{10}(x)$ = 6059.7143. Figures demonstrate the abilities of AMPSO in converging to the true front and in finding diverse set of solutions in the front.

VI. CONCLUSION

This paper has introduced a novel constraint-handling method — adaptive multi-objective particle swarm algorithm. In AMPSO, Pareto non-dominated ranking, tournament selection, crowding distance method were introduced, simultaneously the rate of crowding distance changing was integrated into the algorithm. Simulation results based on some well-known constrained problems and demonstrate the effectiveness, efficiency and robustness of the AMPSO.

ACKNOWLEDGMENT

This work is supported by Grants 60461001 from NSF of China and the project Supported by Grants 0832082, 0991086 from Guangxi Science Foundation.

REFERENCES

[1] Kennedy J, Eberhart R C. Particle swarm optimization [A]. Proceedings of IEEE International Conference on Neural Networks [C]. Piscataway, NJ: IEEE, 1995: 39-43.
[2] Shang Rong-Hua, Jiao Li-Chen, MaWen-Ping. Immune Clonal Multi-Objective Optimization Algorithm for Constrained Optimization. Journal of Software [J]. 2008, 19(11): 2943-2956.
[3] Coello C A C. An Updated survey of evolutionary multi-objective optimization techniques: State of the art and future trends. In: Proceedings of the 1999 Congress on Evolutionary Computation, Washington D C [J]. 1999:3-13.
[4] Parsopoulos K E, Vrahatis M N. Particle swam optimization method in multi-objective problems. Proceedings of the ACM Symposium on Applied Computing. [C]. 2002: 603-607.
[5] Coello C.A.C, et al. Evolutionary algorithms for solving multi-objective problems. New York: Kluwer Academic [M]. 2002.
[6] Deb K, Pratap A, Agarwal S,et al. A fast and elitist multi-objective genetic algorithm: NSGA-Ⅱ. IEEE Trans on Evolutionary Computation [J]. 2002, 6(2): 182-197.
[7] Ren Zi-Hui, Wang Jian. New Adaptive Particle Swarm Optimization Algorithm with Dynamically Changing Inertia Weight. Computer Science [J].2009, 36(2):227-256.
[8] Laumanns M, Zitzler. A unified model for multi-objective evolutionary algorithms with elitism. In Congress on Evolutionary Computation (CEC 2000) [J]. 2000, 46-53.
[9] Gao Li-Li, Liu Hong, Li Tong-Xi. Particle Swarm Based on Cultural Algorithm for Solving Constrained Optimization Problems. Computer Engineering [J]. 2008, 34(5):179-181.
[10] Thomas P. Runarsson, Xin Yao. Stochastic Ranking for Constrained Evolutionary Optimization. IEEE Transactions on Evolutionary Computation [J]. 2000, 4(3): 284-294.
[11] He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Design Problems [J].2007, 31(9-11): 1001-1011.
[12] Carlos A. Coello Coello. The Use of a Multi-objective Optimization Technique to Handle Constraints. Civil Engineering and Environmental Systems [J]. 2000, 17(4):319-346.

Yangquan Zhou Ph. D, Professor. His received the B.S. degree in mathematics from Xianyang Normal University, Xianyang, China. In 1983. the M.S. degree in computer from Lanzhou University , Lanzhou, China. in 1993, and Ph.D. in computation intelligence from Xidian University, Xian, China. in 2006. His current research interests including computation intelligence, neural network.

Shengyu Pei received the M.S degree in computer from Guangxi University for Nationalities, Nanning, China. He current research interests including computation intelligence and applies.