Expert Systems with Applications 38 (2011) 6405–6411

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Cauchy mutation based on objective variable of Gaussian particle swarm optimization for parameters selection of SVM

Qi Wu [a,b,*], Rob Law [b]

[a] Jiangsu Key Laboratory for Design and Manufacture of Micro-Nano Biomedical Instruments, Southeast University, Nanjing 211189, China
[b] School of Hotel and Tourism Management, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ARTICLE INFO

Keywords:
Particle swarm optimization
Gaussian mutation
Cauchy mutation
Support vector machine

ABSTRACT

On the basis of the slow convergence of particle swarm algorithm (PSO) during parameters selection of support vector machine (SVM), this paper proposes a hybrid mutation strategy that integrates Gaussian mutation operator and Cauchy mutation operator for PSO. The combinatorial mutation based on the fitness function value and the iterative variable is also applied to inertia weight. The results of application in parameter selection of support vector machine show the proposed PSO with hybrid mutation strategy based on Gaussian mutation and Cauchy mutation is feasible and effective, and the comparison between the method proposed in this paper and other ones is also given, which proves this method is better than sole Gaussian mutation and standard PSO.

1. Introduction

Recently, particle swarm optimization (PSO) is widely researched as one of the evolutionary computation schemes (Kenedy & Eberheart, 1995). Moreover, many applications and improvements are also reported (AlRashidi & EL-Naggar, 2010; Coelho, 2010; Coelho & Lee, 2008; Fei, Wang, Miao, Tu, & Liu, 2009; Guo, Yang, Wu, Wang, & Liang, 2008; Kiranyaz, Ince, Yildirim, & Gabbouj, 2009; Lin, Ying, Chen, & Lee, 2008; Niu, Li, Li, & Liu, 2009; Shen, Shi, Kong, & Ye, 2007; Tang, Zhuang, & Jiang, 2009; Wu, 2010a, 2010b; Yang, Shao, & Luo, 2005; Yuan & Chu, 2007; Zhao & Yin, 2009). The PSO or the swarm intelligent mimics the social behavior of bird flocking or fish schooling to find an optimal of a certain function. This method consists of individuals called particles which are candidates of the optimal. These individuals group a kind of population called swarm. The algorithm updates these candidates based on their individual experiences, their group experiences and previous movements of the candidate. PSO can find a global optimal, at the same time, this method does not require gradient of the objective function but use values of the function itself. For multimodal functions, the first property of finding global optimal is useful. Many practical applications are involving the problem. The second one of using only values of the objective function avoids complicated procedure of calculation. As a result, the algorithm becomes simple and easy to implement. Therefore, these two points are promising advantages as same as the genetic algorithm. If PSO could utilize any local information without complicated calculation or direct calculation of the gradient, efficiency of the algorithm will be better without lack of the previous two advantages of PSO (Maeda & Kuratani, 2006).

The simultaneous perturbation method is a kind of stochastic gradient method. The scheme can obtain the local information without direct calculation of the gradient. Combination of PSO and the simultaneous perturbation optimization will yield interesting algorithm which has some advantages. From this point of view, we propose a novel particle swarm optimization integrating the simultaneous perturbation. To confirm the performance of simultaneous perturbation, the proposed algorithm is applied as parameters optimization algorithm of support vector machine. Much literatures focus on the integration of PSO and SVM (Fei et al., 2009; Guo et al., 2008; Hong, 2009; Lin et al., 2008; Shen et al., 2007; Tang et al., 2009; Wu, 2010a, 2010b; Wu & Law, 2010; Yuan & Chu, 2007; Zhao & Yin, 2009). PSO with its ability to handle combinatorially explosive problems appears to be very promising for the problem on hand. If PSO could utilize any local information without complicated calculation or direct calculation of the gradient, efficiency of the algorithm will be better. The simultaneous perturbation method is a kind of stochastic gradient method. The scheme can obtain the local information without direct calculation of the gradient. Combination of PSO and the simultaneous perturbation optimization will yield interesting algorithms. By the analysis on the probability density function, Cauchy distribution is similar to normal Gaussian distribution. On the plumb direction, Cauchy distribution is smaller than normal Gaussian distribution, while on the level direction, Cauchy distribution is broader than

* Corresponding author. Address: Jiangsu Key Laboratory for Design and Manufacture of Micro-Nano Biomedical Instruments, Southeast University, Nanjing 211189, China. Tel.: +86 25 51166581; fax: +86 25 511665260.

E-mail addresses: wuqi7812@163.com (Q. Wu), hmroblaw@inet.polyu.edu.hk (R. Law).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.08.069

normal distribution. Then, if Cauchy distribution is adopted as a mutate operator, the individuals shall improve themselves in biggish scope and discard local best solution easily. From this point of view, we propose a novel optimization algorithm using the simultaneous perturbation that integrate the Gaussian mutation and Cauchy mutation operators. However, these published papers pay little attention to the integration between Cauchy mutation and Gaussian mutation. Therefore, the simultaneous perturbation strategy based on Gaussian mutation and Cauchy mutation is proposed to improve the searching performance of PSO. An application in parameters selection of support vector machine is shown to confirm validity and properties of the algorithms.

In this paper, we combine the Cauchy mutation strategies with Gaussian PSO, and put forward a new type of PSO, called CGPSO. Based on CGPSO, an application in parameters selection of support vector machine is executed. The rest of this paper is organized as follows. Standard PSO is described in Section 2. Hybrid PSO based on Cauchy and Gaussian mutation is derived in Section 3. The steps of the proposed PSO are arranged in Section 4. An application in parameter selection of SVM is given in Section 5. Section 6 draws the conclusions.

2. Standard particle swarm optimization

Similarly to evolutionary computation techniques, PSO uses a set of particles, representing potential solutions to the problem under consideration. The swarm consists of m particles; each particle has a position $X_i = \{x_{i1}, x_{i2}, \ldots, x_{id}\}$, and a velocity $V_i = \{v_{i1}, v_{i2}, \ldots, v_{id}\}$, where $i = 1, 2, \ldots, n$ and moves through a n-dimensional searching space. According to the global variant of the PSO algorithm, each particle moves towards its best previous position and towards the best particle g in the swarm. Let us denote the best previously visited position of the ith particle that gives the best fitness value as $p\_c_i = \{p\_c_{i1}, p\_c_{i2}, \ldots, p\_c_{id}\}$, and the best previously visited position of the swarm that gives best fitness as $\boldsymbol{p\_g} = \{p\_g_1, p\_g_2, \ldots, p\_g_n\}$.

PSO is inspired by swarms of insects (Kenedy & Eberheart, 1995). Since its introduction, PSO has gone through several improvements such as the inertial weighting factor, and fine-tuning. Often these changes come about as a result of numerous experiments and observations. Occasionally, intuitive improvements are not correct such as using a midpoint between the global and local best. This kind of failure is very instructive if the cause can be understood. This visualization method greatly enhances understanding, improves intuition, and explains this type of failure.

where:

v_{ij}^{k+1}	The particle's new velocity for the next generation
w	The inertial dampener. A measure of how much the particle "trusts" its own exploration
v_{ij}^{k}	The particle's current velocity
c_1r	A uniformly distributed random number from 0 to c_1. A measure of how much a particle "trusts" its neighborhood best velocity
p_c_{ij}	The neighborhood best position
"–"	The difference of two positions is the velocity that will transform the second position into the first position
x_{ij}^{k}	The current position
c_2r_2	A uniformly distributed random number from 0 to c_2. Independent from c_1r, a measure of how much a particle "trusts" the global velocity
p_g_j	The global best position
"+"	The transformation of a position using the velocity (yields a position)
x_{ij}^{k+1}	The particle's new "moved" position. The position of the next generation

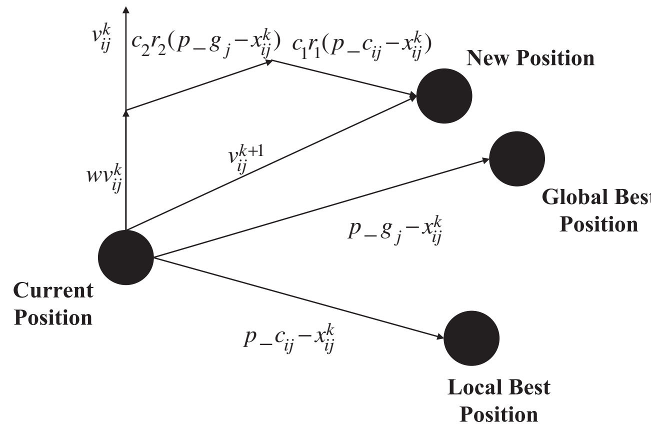

Fig. 1. Illustration of PSO.

The visualization of particle swarm optimization is illuminated in Fig. 1. This visualization method greatly enhances understanding, and improves intuition. The change of position of each particle from one iteration to another can be computed according to the distance between the current position and its previous best position, and the distance between the current position and the best position of swarm. Then the updating of velocity and particle position can be obtained by using the following equations:

$$v_{ij}^{k+1} = wv_{ij}^k + c_1r_1\left(p\_c_{ij} - x_{ij}^k\right) + c_2r_2\left(p\_g_j - x_{ij}^k\right) \tag{1}$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \tag{2}$$

where w is called inertia weight and is employed to control the impact of the previous history of velocities on the current one. Accordingly, the parameter w regulates the trade-off between the global and local exploration abilities of the swarm. A large inertia weight facilitates global exploration, while a small one tends to facilitate local exploration. A suitable value of the inertia weight w usually provides the balance between global and local exploration abilities and consequently results in a reduction of the number of iterations required to locate the optimum solution. $k = 1, 2, \ldots, K_{\max}$ denotes the iteration number, c_1 is the cognition learning factor, c_2 is the social learning factor, r_1 and r_2 are random numbers uniformly distributed in $[0,1]$.

The nature of these equations is analogous to planetary bodies orbiting a pair of suns, but is inherently stable. Under these equations (and given appropriate coefficients), all of the particles in the swarm "gravitate" toward the best solution found so far. This produces a local search. It is inherently stable because as the particles get closer to the best solution, there is a weaker pull toward it, thus guaranteeing each particle converges (or orbits) near the best solution. This differs from the orbiting analogy since planetary bodies have a stronger pull toward the sun as they get nearer to it, thus are likely to be 'flung" away from the sun in a slingshot orbit.

Thus, the particle flies through potential solutions towards $p\_c_i^k$ and p_g^k in a navigated way while still exploring new areas by the stochastic mechanism to escape from local optima. Since there was no actual mechanism for controlling the velocity of a particle, it was necessary to impose a maximum value $V_{\max}$ on it. If the velocity exceeds the threshold, it is set equal to $V_{\max}$, which controls the maximum travel distance at each iteration to avoid this particle flying past good solutions.

The PSO algorithm is terminated with a maximal number of generations or the best particle position of the entire swarm can not be improved further after a sufficiently large number of generations.

The PSO algorithm has shown its robustness and efficacy in solving function value optimization problems in real number spaces.

3. Hybrid particle swarm optimization based on Cauchy and Gaussian mutation

3.1. Gaussian particle warm optimization

PSO have been used extensively for a variety of optimization problems and in most of these cases PSO has proven to have superior computational efficiency (AlRashidi & EL-Naggar, 2010; Coelho, 2010; Coelho & Lee, 2008; Fei et al., 2009; Guo et al., 2008; Kiranyaz et al., 2009; Lin et al., 2008; Niu et al., 2009; Shen et al., 2007; Tang et al., 2009; Wu, 2010a, 2010b; Yang et al., 2005; Yuan & Chu, 2007; Zhao & Yin, 2009). Further, PSO does not use any gradient-based information. It incorporates a flexible and well-balanced mechanism to adapt to the global and local exploration and exploitation abilities within a short computation time. Hence, this method is efficient in handling large and complex search spaces. PSO with its ability to handle combinatorially explosive problems appears to be very promising for the problem on hand. If PSO could utilize any local information without complicated calculation or direct calculation of the gradient, efficiency of the algorithm will be better without lack of the previous two advantages of PSO. The simultaneous perturbation method is a kind of stochastic gradient method. The scheme can obtain the local information without direct calculation of the gradient. Combination of PSO and the simultaneous perturbation optimization will yield interesting algorithms. From this point of view, we propose a novel optimization algorithm using the simultaneous perturbation. First, normal Gaussian operator is applied to the mutation of PSO.

The Probability density function of the single variable normal distribution is described as follows:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-(z-\mu)^2/(2\sigma^2)\right] \tag{3}$$

where μ is mean value, σ^2 is variance. For convenience, $N(\mu,\sigma^2)$ represents the normal distribution.

For multivariable normal distribution, The Probability density function can be described as follows:

$$p(\boldsymbol{z}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{C}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{C}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\right] \tag{4}$$

where $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)^{\mathrm{T}}$ represent stochastic vector composed of many stochastic variable, $\boldsymbol{\mu}$ is a vector of mean value, $\boldsymbol{C}$ is covariance matrix, n is dimension number of vector $\boldsymbol{z}$. $\boldsymbol{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ represents the multivariable normal distribution.

When the mean value vector is equal to $\mathbf{0}$, the corresponding normal distribution can be given as follows:

$$p(z) = \frac{1}{(2\pi)^{\frac{n}{2}}\prod_{i=1}^{n}\sigma_i} \exp\left[-\sum_{i=1}^{n} z_i^2/(2\sigma^2)\right] \tag{5}$$

If $z_1, z_2, \ldots, z_n$ are independent each other and obey the same distribution, viz., $\sigma_i = \sigma$, $i = 1,2,\ldots,n$, then we have the probability density function as follows:

$$p(z) = \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^n} \exp\left[-\sum_{i=1}^{n} z_i^2/(2\sigma^2)\right] \tag{6}$$

We know a hyper ellipsoid can be represent as follows:

$$\sum_{i=1}^{n}\left(\frac{z_i}{\sigma_i}\right)^2 = r, \quad (r \in R) \tag{7}$$

However, when $\sigma_i = \sigma (i = 1,2,\ldots,n)$, Eq. (6) denotes a hyper sphere. Therefore, the density curve with equal probability determined by Eq. (5) is a hyper ellipsoid, while that determined by Eq. (6) is a hyper sphere.

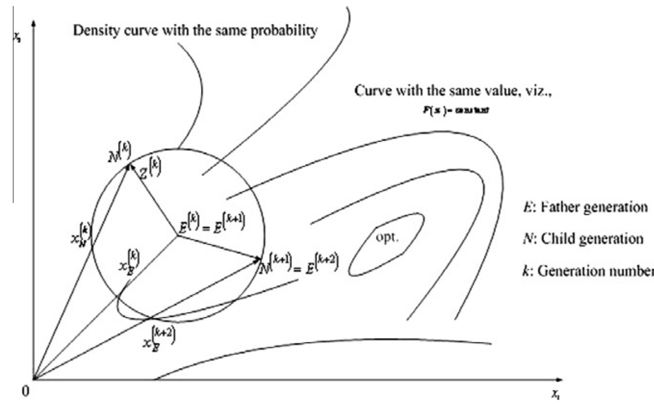

Fig. 2. One mutation process of two variable evolutionary strategies.

The geometry meaning of mutation process can be illustrated by Fig. 2 on the conditions of Eq. (6)

Aim at the disadvantage of the standard PSO, the adaptive mutation operator is proposed to regulate the inertia weight of velocity on the fitness value of object function. The normal mutation operator is considered to correct the direction of particle velocity at the same time. The adaptive mutation is highly efficient operator on the conditions of real number code. The solution quality is related tightly with the mutation operator. The aforementioned problem is addressed by incorporating adaptive mutation and normal mutation for the previous velocity of the particle. Thus, in the latest versions of the PSO, Eqs. (1) and (2) are changed into the following ones:

$$\begin{aligned} v_{id}^{k+1} = {} & (1-\lambda)w_{id}^k v_{id}^k + \lambda N(0,1)N(0,\sigma_i^k) + c_1 r_1 (p_{id} - x_{id}^k) \\ & + c_2 r_2 (p_{gd} - x_{id}^k) \end{aligned} \tag{8}$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \tag{9}$$

$$w_{id}^k = \beta(1 - f(x_i^k)/f(x_m^k)) + (1-\beta)w_{id}^0 \exp(-\alpha k^2) \tag{10}$$

$$\sigma_i^{k+1} = \sigma_i^k \exp(N(0,1) + N_i(0,\Delta\sigma)) \tag{11}$$

The first item of formula (10) $((1 - f(x_i^k)/f(x_m^k)))$ denotes the mutation based on the fitness function value $f(x_i^k)$. The particle with the bigger fitness mutate in a smaller scope, the one with the smaller fitness mutate in a bigger scope. The second item of formula (10) $(\exp(-\alpha k^2))$ represents the mutation based on the iterative variable (k). The particles mutate in big scope on the smaller iterative variable and search the local optimal value, the one do in small scope on the bigger iterative variable, search the optimal value in small space and gradually reach the global optimal value. The operator of normal Gaussian mutation correct the change of particle velocity is represented in formula (11). In the strategy of normal Gaussian mutation, the individual consists of explaining vector $v = (v_1, v_2, \ldots, v_n)$ and perturbing vector $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$. The perturbation vector mutate its value by (11) on the each iterative process as a controlling vector of explaining vector.

The adaptive and normal Gaussian mutation operators can restore the diversity loss of the population and improve the capacity of the global search of the algorithm.

3.2. Improvement based on Cauchy mutation

From the view point of probability density function, Cauchy distribution is similar to normal Gaussian distribution. On the plumb direction, Cauchy distribution is smaller than normal Gaussian distribution, while on the level direction, Cauchy distribution is broader than normal distribution. Hence, Cauchy distribution is adopted as a mutate operator. The individuals can improve themselves in biggish scope and discard local best solution easily. Compared with normal Gaussian distribution, the smaller central part of Cauchy distribution is its shortcoming. In view of the advantage of Cauchy distribution, Cauchy operator is integrated into the above Gaussian PSO.

The Cauchy density function can be defined as follows:

$$f_t(x) = \frac{1}{\pi}\frac{t}{t^2 + x^2}, \quad -\infty < x < +\infty \tag{12}$$

where $t > 0$ is proportion parameter, the corresponding distribution function is described as follows:

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x}{t}\right) \tag{13}$$

The version adopting Cauchy mutation operator based on Eqs. (8)–(11) can be represented as follows:

$$v_{id}^{k+1} = (1-\lambda)w_{id}^k v_{id}^k + \lambda(\eta_i \cdot N(0,\sigma_i^k)) + c_1 r_1 (p_{id} - x_{id}^k) + c_2 r_2 (p_{gd} - x_{id}^k) \tag{14}$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \tag{15}$$

$$w_{id}^k = \beta(1 - f(x_i^k)/f(x_m^k)) + (1-\beta)w_{id}^0 \exp(-\alpha k^2) \tag{16}$$

$$\sigma_i^{k+1} = \sigma_i^k \exp(N(0,1) + N_i(0,\Delta\sigma)) \tag{17}$$

where η_i is Cauchy distribution number.

η_i is not Gaussian distribution number but Cauchy distribution number. The number generated from Cauchy distribution is away from origin, has wider scope than that generated from Gaussian distribution. If Cauchy mutation substitutes Gaussian mutation to generate the next generation, the solution can dap the local optimum easily.

4. Support vector machine

It is difficult to confirm the optimal parameters of the SVM model. There exists crossover error in crossover validation method used commonly to determine penalty coefficient, controlling vector and kernel parameter. To overcome the shortage, particle swarm optimization (PSO), is applied to the parameter optimization of SVM (Vapnik, 1995). The literatures of SVM are arranged in the following part.

Suppose that we have a set of training samples $T = \{(\boldsymbol{x}_1, y_1), \ldots (\boldsymbol{x}_i, y_i), \ldots, (\boldsymbol{x}_l, y_l)\}$, where $\boldsymbol{x}_i \in \boldsymbol{R}^d$, $y_i \in R$. For an integer i, the regression model defines the relation between $\boldsymbol{x}_i$ and $f(\boldsymbol{x}_i)$ as: $f(\boldsymbol{x}_i) = \boldsymbol{w} \cdot \boldsymbol{x}_i + b$, $b \in R$, $i = 1, 2, \ldots, l$.

In order to estimate y_i for $\boldsymbol{x}_i$, we should solve the following constrained optimization problem.

$$\min_{\boldsymbol{w}, b, \xi^{(*)}, \varepsilon} \quad \tau(\boldsymbol{w}, \xi^{(*)}, \varepsilon) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C \cdot \left(v \cdot \varepsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*)\right) \tag{18}$$

$$\text{s.t.} \quad (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - y_i \leqslant \varepsilon + \xi_i, \tag{19}$$

$$y_i - (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \leqslant \varepsilon + \xi_i^*, \tag{20}$$

$$\xi^{(*)} \geqslant 0, \quad \varepsilon \geqslant 0. \tag{21}$$

where $\boldsymbol{w}$ and $\boldsymbol{x}_i$ are d dimensional column vectors, $C > 0$ is a penalty factor which controls the equilibrium between the complexity of model and training error, $v \in (0, 1]$ a parameter that controls the support vector numbers, ε that for controlling tube size, and $\xi^{(*)} = (\xi_1, \xi_1^*, \ldots, \xi_i, \xi_i^*, \ldots, \xi_l, \xi_l^*)^T$ that are slack variables guarantee the satisfaction of constraint condition.

Eq. (18) is a quadratic programming (QP) problem. By introducing Lagrangian multipliers, a Lagrangian function can be defined as

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\alpha}^{(*)}, \beta, \boldsymbol{\xi}^{(*)}, \varepsilon, \boldsymbol{\eta}^{(*)}) = {} & \frac{1}{2}\|\boldsymbol{w}\|^2 + C \cdot v \cdot \varepsilon + \frac{C}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ & - \beta \cdot \varepsilon - \sum_{i=1}^{l}(\eta_i \xi_i + \eta_i^* \xi_i^*) \\ & - \sum_{i=1}^{l}\alpha_i(\varepsilon + \xi_i + y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i - b) \\ & - \sum_{i=1}^{l}\alpha_i^*(\varepsilon + \xi_i^* - y_i + w \cdot x_i + b), \end{aligned} \tag{22}$$

where $\alpha_i^{(*)}$, β, $\eta_i^{(*)} \geqslant 0 (i = 1, \ldots, l)$ are Lagrangian multipliers. Differentiating the Lagrangian function (22) with respect to $\boldsymbol{w}$, b, ε, $\xi_i^{(*)}$, we get

$$\begin{cases} \nabla_w L = 0 \Rightarrow \boldsymbol{w} = \sum\limits_{i=1}^{l}(\alpha_i^* - \alpha_i)x_i \\ \nabla_b L = 0 \Rightarrow \sum\limits_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \\ \nabla_\varepsilon L = 0 \Rightarrow C \cdot v - \sum\limits_{i=1}^{l}(\alpha_i + \alpha_i^*) - \beta = 0 \\ \nabla_{\xi_i^{(*)}} L = 0 \Rightarrow \frac{C}{l} - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \end{cases}. \tag{23}$$

By substituting Eq. (23) into Eq. (22), we can obtain the corresponding dual form of function (18)

$$\max_{\alpha, \alpha^*} \quad W(\alpha^{(*)}) = -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)y_i, \tag{24}$$

$$\text{s.t.} \quad 0 \leqslant \alpha_i^{(*)} \leqslant \frac{C}{l}, \tag{25}$$

$$\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) \leqslant C \cdot v, \tag{26}$$

$$\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \tag{27}$$

Selecting the appropriate v, C and $K(\boldsymbol{x}, \boldsymbol{x}')$, we can construct and solve the optimal problem (24) by QP method to obtain the optimal solution $\bar{\boldsymbol{\alpha}}^{(*)} = (\bar{\alpha}_1, \bar{\alpha}_1^*, \ldots, \bar{\alpha}_l, \bar{\alpha}_l^*)^T$. The relation between $\boldsymbol{x}$ and $f(\boldsymbol{x})$ of the above standard v-SVM is expressed as

$$f(\boldsymbol{x}) = \sum_{i=1}^{l}(\bar{\alpha}_i^* - \bar{\alpha}_i)K(\boldsymbol{x}_i, \boldsymbol{x}) + b. \tag{28}$$

5. The parameters selection of SVM

The CGPSO is described in steps as follows:

Algorithm 1

Step (1) Data preparation: Training and testing sets are represented as Tr and Te, respectively.

Step (2) Particle initialization and PSO parameters setting: Generate initial particles comprised of C, v and a. Set the PSO parameters including number of particles, particle dimension, number of maximal iterations, error

Table 1
PDT factors of mold characteristics.

Mold characteristics	Unit	Expression	Weight
Structure complexity (SC)	Dimensionless	Linguistic information	0.90
Model difficulty (MD)	Dimensionless	Linguistic information	0.70
Wainscot gauge variation (WGV)	Dimensionless	Linguistic information	0.70
Cavity number (CN)	Dimensionless	Numerical information	0.80
Mold size (height/diameter) (MS)	Dimensionless	Numerical information	0.55
Form feature number (FFN)	Dimensionless	Numerical information	0.55

Table 2
Learning and testing patterns for the CGPSO-SVM.

No.	Molds	Input data						Desired PDT (h)
	Name	SC	MD	WGV	CN	MS	FFN	
1	Global handle	L	L	L	4	3.10	3	23.0
2	Water bottle lid	H	L	H	4	0.56	7	45.5
3	Medicine lid	H	M	VL	4	1.50	6	37.0
4	Footbath basin	VL	VL	VL	1	0.50	3	10.0
5	Litter basket	L	M	H	1	2.10	12	42.5
6	Plastic silk	L	M	M	1	7.10	4	29.5
7	flower	M	H	L	1	0.50	15	48.0
8	Dining chair	H	VL	L	2	8.07	2	30.0
9	Spindling	H	L	L	1	0.45	5	24.5
10	bushing	VH	H	M	1	0.30	7	49.0
...	Three-way pipe	...	...	...	...	...	...	...
59	Hydrant shell	H	L	L	10	5	10	59.0
60	Paper-lead pulley Winding tray	M	M	VH	12	7.9	2	69

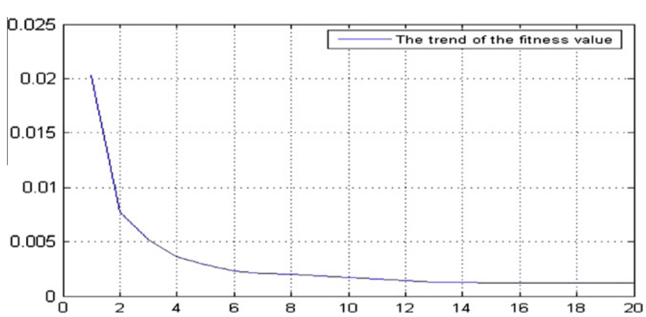

Fig. 3. The change trend of the mean fitness value of each generation.

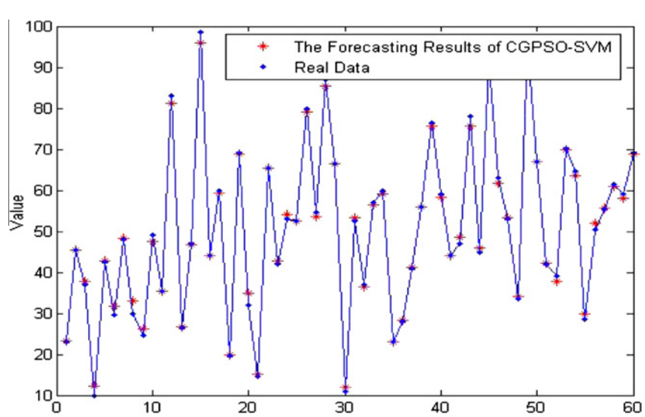

Fig. 4. The estimating results from CGPSO-SVM model.

limitation of the fitness function, velocity limitation, and inertia weight for particle velocity, normal Gaussian distribution, the perturbation momentum; the coefficient of controlling particle velocity attenuation, adaptive coefficient, increment coefficient. Set iteration $i = 0$. and perform the training process from step3–10.

Step (3) Set iteration $i = i + 1$.

Step (4) Go to step 2 of Algorithm 2 and get the support vector $a^{(*)}$. Compute the estimating result by (28).

Step (5) Compute the fitness function value of each particle. Take current particle as individual extremum point of every particle and do the particle with minimal fitness value as the global extremum point.

Step (6) Adopt the adaptive mutation operator by formula (16) and normal mutation operator by formula (17) to manipulate particle velocity.

Step (7) Update the particle position by formula (14) and (15) and form new particle swarms.

Step (8) Compute the fitness function value of each particle according to the updated position.

Step (9) Compare the current optimal particle with last generational optimal particle and update global and personal best.

Step (10) Particle manipulations: each particle moves to its next position using formula (14) and (15).

Step (11) Stop condition checking: if stopping criteria (maximum iterations predefined) are met, go to step 3. Otherwise, go to the next step.

Step (12) To avoid overtraining, we observe the validation accuracy curve, and stop training when the iteration has the best validation accuracy during the training process.

Table 3
Comparison of estimating results from three models.

Model	1	2	3	4	5	6	7	8	9	10	11	12
Real value	69	59	61.5	55.5	50.5	28.5	64.5	70	39	42	67	95.5
PSOv-SVM	70.4	59.2	62.4	58.1	53.8	31.7	64.8	71.3	38.9	43.9	68.5	94.3
GPSOv-SVM	68.9	57.7	60.9	56.9	52.2	30.2	63.2	69.8	37.4	42.4	66.9	92.7
CGPSO-SVM	68.9	57.9	61.0	56.3	51.9	29.9	63.5	69.8	37.7	42.3	66.9	93.3

Table 4
Error statistic of three models.

Model	MAE	MAPE	MSE
PSOv-SVM	1.4917	0.0307	3.3158
GPSOv-SVM	1.1000	0.0213	1.8417
CGPSO-SVM	0.8667	0.0169	1.1417

Step (13) End the training procedure, output the optimal parameters C, v and a.

On the basis of the v-SVRM model, we can summarize a regression estimating algorithm as the follows:

Algorithm 2

Step (1) Initialize the original data by normalization and fuzzification, and form training set.
Step (2) Select the kernel function $K(x,x')$, call Algorithm 1 and get the optimal parameters combination vector (C,v,a), solve the optimization problem (24) and obtain the parameters $a^{(*)}$.
Step (3) For a new regression estimation, extract influencing characteristics and form a set of input variables x.
Step (4) Compute the estimating result $f(x)$ by (28).

6. Application

To illustrate this time estimation method, the design of plastic injection mold is studied. Injection mold is a kind of single-piece-designed product and the design process is usually driven by customer orders. Many product development projects involve the design process of injection mold and the pre-estimating time is meaningful for the planning, scheduling and optimization of the whole product development process.

In the following text, we construct a CGPSO-SVM to estimate the mold design time. We place emphasis on the mold characteristics associated with design time. Some characteristics with large influencing weights are gathered to develop a time factor list, as shown in Table 1.

In our experiments, 60 sets of molds with corresponding design time are selected from past projects in a typical company. The detailed characteristic data and design time of these molds compose the corresponding patterns, as shown in Table 2. We train the CGPSO-SVM with 48 patterns, and the others are used for testing.

The initial parameters of the CGPSO are given as follows: the maximal iterative number: $k_{\max} = 100$; inertia weight $w^0 = 0.9$; positive acceleration constants $c_1, c_2 = 2$; the standard error of normal Gaussian distribution $\Delta\sigma = 0.5$; the adaptive coefficient $\beta = 0.8$; increment coefficient $\lambda = 0.1$; the fitness accuracy of the normalized samples is equal to 0.0002; the coefficient of controlling particle velocity attenuation $\alpha = 2$.

To evaluate forecasting capability of CGPSO-SVM, some evaluation indexes, such as mean absolute error (**MAE**), mean absolute percentage error (**MAPE**) and mean square error (**MSE**), are utilized to handle the estimating results of CGPSO-SVM (where the CGPSO is used to optimize parameters of the standard SVM), GPSO-SVM (where the GPSO is used to optimize parameters of the standard SVM), and PSO-SVM (where standard PSO is used to optimize parameters of standard SVM). The optimal combinational parameters are obtained by Algorithm CPSO, viz., $C = 795.69$, $v = 0.95$ and $\sigma = 0.47$. The change trend of the mean fitness value of each generation is shown in Fig. 3. It is obvious that the CGPSO is convergent. Then the CGPSO is applied to seeking the optimal parameters SVM. Fig. 4 illuminates the estimating results provided by CGPSO-SVM.

To verify the capability of CGPSO-SVM, the models (GPSO-SVM and PSO-SVM) are selected to deal with the above product design time series. Their results are shown in Table 3.

The indexes **MAE**, **MAPE** and **MSE** are employed to evaluate the estimating capability of three models, as shown in Table 4. To represent the error trend well, the estimating results of the last 12 injection molds are used to analyze the forecasting performance of the above models. It is dear that the parameters optimized by CGPSO are of better choice to construct SVM model for the design of plastic injection mold than the ones by PSO and GPSO.

The indexes **MAE**, **MAPE** and **MSE** provided by CGPSO-SVM are also better than those of GPSO-SVM. Since CGPSO adds a Cauchy mutation on the basis of GPSO, the run time of CGPSO-SVM is close to that of GPSO-SVM, that of PSO-SVM being the shortest.

This study presents an improved PSO-based approach, capable of searching for the optimal parameter values for SVM. Comparison of the obtained results with those of other approaches demonstrates that the developed CGPSO+SVM approach has a better estimating accuracy than others tested. Results of this study were obtained with an RBF kernel function. However, other kernel parameters can also be optimized using the same approach. Experimental results obtained from practical datasets, other public datasets and real-world problems can be tested in the future to verify and extend this approach.

7. Conclusions

In this paper, CGPSO-SVM is applied to estimate product design time. The CGPSO does not need to consider the analytic property of the generalization performance measure and can determine multiple hyper-parameters at the same time. In the CGPSO-SVM approach, CGPSO is used to select suitable parameters of SVM, which avoids over-fitting or under-fitting of the SVM model occurring because of the improper determining of these parameters. CGPSO is a new optimization method, which not only has strong global search capability, but also is very easy to implement. So it is very suitable for parameters selection of SVM. The real data sets are used to investigate its feasibility in estimating product design time. The experimental results indicate that the CGPSO-SVM method can achieve greater estimating accuracy than GPSO-SVM and PSO-SVM models.

However, as one of the limitations of the study, the complexity of the proposed CGPSO has not been fully explored, which is sure to fall into the scope of our future research.

Acknowledgements

This research was partly supported by the National Natural Science Foundation of China under Grant 60904043, a research grant funded by the Hong Kong Polytechnic University, China Postdoctoral Science Foundation (20090451152), Jiangsu Planned Projects for Postdoctoral Research Funds (0901023C) and Southeast University Planned Projects for Postdoctoral Research Funds.

References

AlRashidi, M. R., & EL-Naggar, K. M. (2010). Long term electric load forecasting based on particle swarm optimization. *Applied Energy, 87*(1), 320–326.

Coelho, L. S. (2010). Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Systems with Applications, 37*(2), 1676–1683.

Coelho, L. S., & Lee, C. S. (2008). Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches. *International Journal of Electrical Power & Energy Systems, 30*(5), 297–307.

Fei, S. w., Wang, M. J., Miao, Y. b., Tu, J., & Liu, C. L. (2009). Particle swarm optimization-based support vector machine for forecasting dissolved gases content in power transformer oil. *Energy Conversion and Management, 50*(6), 1604–1609.

Guo, X. C., Yang, J. H., Wu, C. G., Wang, C. Y., & Liang, Y. C. (2008). A novel LS-SVMs hyper-parameter selection based on particle swarm optimization. *Neurocomputing, 71*(16–18), 3211–3215.

Hong, W. C. (2009). Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model. *Energy Conversion and Management, 50*(1), 105–117.

Kenedy, J., & Eberheart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (pp. 1942–1948).

Kiranyaz, S., Ince, T., Yildirim, A., & Gabbouj, M. (2009). Evolutionary artificial neural networks by multi-dimensional particle swarm optimization. *Neural Networks, 22*(10), 1448–1462.

Lin, S. W., Ying, K. C., Chen, S. C., & Lee, Z. J. (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications, 35*(4), 1817–1824.

Maeda, Y., & Kuratani, T. (2006). Simultaneous Perturbation Particle Swarm Optimization. In *IEEE Congress on Evolutionary Computation* (pp. 672–676).

Niu, D., Li, J., Li, J., & Liu, D. (2009). Middle-long power load forecasting based on particle swarm optimization. *Computers & Mathematics with Applications, 57*(11–12), 1883–1889.

Shen, Q., Shi, W. M., Kong, W., & Ye, B. X. (2007). A combination of modified particle swarm optimization algorithm and support vector machine for gene selection and tumor classification. *Talanta, 71*(4), 1679–1683.

Tang, X., Zhuang, L., & Jiang, C. (2009). Prediction of silicon content in hot metal using support vector regression based on chaos particle swarm optimization. *Expert Systems with Applications, 36*(9), 11853–11857.

Vapnik, V. (1995). *The nature of statistical learning*. New York: Springer.

Wu, Q. (2010a). Power load forecasts based on hybrid PSO with Gaussian and adaptive mutation and Wν-SVM. *Expert Systems with Applications, 37*(1), 194–201.

Wu, Q. (2010b). A hybrid-forecasting model based on Gaussian support vector machine and chaotic particle swarm optimization. *Expert Systems with Applications, 37*(3), 2388–2394.

Wu, Q., & Law, R. (2010). Complex system fault diagnosis based on a fuzzy robust wavelet support vector classifier and an adaptive Gaussian particle swarm optimization [J]. *Information Sciences, 180*(23), 4514–4528.

Yang, S. M., Shao, D. G., & Luo, Y. J. (2005). A novel evolution strategy for multiobjective optimization problem. *Applied Mathematics and Computation, 170*(2), 850–873.

Yuan, S. F., & Chu, F. L. (2007). Fault diagnostics based on particle swarm optimization and support vector machines. *Mechanical Systems and Signal Processing, 21*(4), 1787–1798.

Zhao, H. B., & Yin, S. (2009). Geomechanical parameters identification by particle swarm optimization and support vector machine. *Applied Mathematical Modelling, 33*(10), 3997–4012.