energies

MDPI

Article

Velocity-Controlled Particle Swarm Optimization (PSO) and Its Application to the Optimization of Transverse Flux Induction Heating Apparatus

Youhua Wang [1,2], Bin Li [1,2], Liuxia Yin [1,2], Jiancheng Wu [1,2], Shipu Wu [1,2] and Chengcheng Liu [1,2,*]

1 State Key Laboratory of Reliability and Intelligence of Electrical Equipment, School of Electrical Engineering, Hebei University of Technology, Tianjin 300130, China; wangyi@hebut.edu.cn (Y.W.); lbin@hebut.edu.cn (B.L.); yinliuxia888@126.com (L.Y.); wjc254678@163.com (J.W.); necbmskk@outlook.com (S.W.)

2 Key Laboratory of Electromagnetic Field and Electrical Apparatus Reliability of Hebei Province, School of Electrical Engineering, Hebei University of Technology, Tianjin 300130, China

* Correspondence: 2016020@hebut.edu.cn; Tel.: +86-137-5269-4973

Received: 19 December 2018; Accepted: 30 January 2019; Published: 2 February 2019



Abstract: The main disadvantage of transverse flux induction heating (TFIH) is its resulting non-uniform temperature distribution on the surface of the strip at the inductor outlet. For obtaining a uniform temperature distribution, an improved particle swarm optimization (PSO) named velocity-controlled PSO (VCPSO) is proposed and applied to optimize this problem. Support vector machine (SVM) is adopted to establish a regression model to replace the complex and time-consuming coupling calculation process involved in TFIH problem. Simulation results of several test functions show that VCPSO performs much better than standard PSO (SPSO). Moreover, based on the existing research and experiments, the application of VCPSO combined with SVM to the TFIH problem achieves satisfactory results.

Keywords: transverse flux induction heating (TFIH); particle swarm optimization (PSO); velocity controlled PSO (VCPSO); support vector machine (SVM)

1. Introduction

Transverse flux induction heating (TFIH) technology has become more and more popular in both industry and domestic applications due to its fast, highly efficient, and high-quality heating abilities [1,2]. Among these applications, heating metal strips and sheets are very important examples; however, for this kind of application, the requirement for uniform temperature distribution on the surface of the strip or sheets is very high, and the main disadvantage of TFIH is its non-uniform temperature distribution on the surface of the heating target. In order to obtain a uniform temperature distribution, the input current, frequency, coil dimension, and structures need to be optimized.

Particle swarm optimization (PSO) is a good method for optimization of the TFIH problem. It was originally proposed and developed by Kennedy and Eberhart [3–6]. With the adopted simple concept, PSO can be implemented in a few lines of computer codes easily. It converges very fast and does not require that the function to be optimized must be differentiable [7,8]. Because of its obvious advantages, PSO has been applied to many fields. However, it also has the problem of low search precision and tending to fall into local extremum when solving some complex problems, which gives rise to substantial efforts to improve its optimization performance. In this paper, an algorithm called velocity-controlled PSO (VCPSO) is put forward, which improves the particles' distribution during the optimizing process. The improvement is reflected in two ways: the first is uniform initialization

of population; the second is velocity regulation according to the number of particles that exceed the feasible region during the optimizing process. Simulation test results of several typical functions reveal that VCPSO has better global search ability, higher search precision, and a faster convergence rate when compared with standard PSO (SPSO).

During the TFIH optimization process, the average relative error of the strip's temperature at the outlet is needed, since it relates to the optimization target. However, the calculation involved in this process is very complex and time-consuming. Generally, it may need 2 to 3 h to obtain one needed result. Thus, in this work, support vector machine (SVM) is used to establish a regression model between the applied current, frequency, coil size, and the average relative error of temperature of the TFIH. The 3D finite element method (FEM) package MagNet and ThermNet is adopted for calculating the temperature distribution of the strips heated by the TFIH system. After that, the analysis results are used to train the SVM, and the regression model of the TFIH system is obtained. Since the complex calculation process has been replaced by the regression model, thus the complexity has been reduced then the optimization efficiency can be improved greatly.

In this paper, the application of VCPSO combined with SVM to the TFIH problem is implemented by using the MATLAB platform (2015b, MathWorks, Natick, MA, USA). The whole optimization process is carried out successfully to get the optimal design variables. Comparison of results before and after the optimization shows that the temperature distribution of the strip after the optimization is more uniform and it can provide a theoretical reference for the optimal design of TFIH equipment.

2. Velocity-Controlled Particle Swarm Optimization

2.1. Principle of Particle Swarm Optimization

PSO is a kind of swarm intelligence algorithm [9]. It simulates the foraging behavior of birds. In a particle swarm optimizer, each particle represents a potential solution to a problem and adjusts its flying according to its own flying experience and its companions' flying experience [10]. Each particle is treated as a point in a D-dimensional space. The i-th particle is represented as $\boldsymbol{X}_i = (x_{i1}, x_{i2}, \ldots, x_{i\mathrm{D}})$. The velocity of particle i is represented as $\boldsymbol{V}_i = (v_{i1}, v_{i2}, \ldots, v_{i\mathrm{D}})$. The best previous position of any particle is recorded as $\boldsymbol{P}_i = (p_{i1}, p_{i2}, \ldots, p_{i\mathrm{D}})$. The best previous position of all particles is recorded as $\boldsymbol{P}_g = (p_{g1}, p_{g2}, \ldots, p_{g\mathrm{D}})$. The particles' velocity and position are updated through iterations according to the following equations:

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_1(p_{id}(t) - x_{id}(t)) + c_2 r_2(p_{gd}(t) - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

where c_1 and c_2 are usually two positive constants, in this article $c_1 = c_2 = 2$, r_1 and r_2 are two random numbers ranges from 0 to 1, whose values are recalculated every time Equation (1) is used; t represents the t-th iteration while $t + 1$ represents the $t + 1$-th iteration; w is the inertia weight [11], whose value is defined by Equation (3):

$$w = \frac{1}{2}\left(\cos\left((t/\mathrm{N}_{\max})^k \pi\right) + 1\right) \quad (3)$$

where $\mathrm{N}_{\max}$ represents maximum number of iterations, t represents the t-th iteration while $t + 1$ represents the $t + 1$-th iteration, k is a constant and its value is 0.5, 1, or 2 according to the distribution of population—when the distribution is too narrow, k equals 2; when it is too widely distributed k equals 0.5; otherwise k equals 1.

The performance of each particle is measured according to a pre-defined fitness function, which is determined by the problem to be solved. The inertia weight w is employed to control the impact of the previous history of velocities on the current velocity, thereby influencing the trade-off between global and local exploration abilities of the "flying points". A larger inertia weight w tends to facilitate a global search while a smaller inertia weight tends to facilitate a local search. Suitable selection of the

inertia weight w can provide a balance between global and local exploration abilities [12]. In SPSO, the inertia weight w is usually linearly decreased from a relatively large value to a small value through the course of the run, thus the algorithm tends to have more global search ability at the beginning of the run while having more local search ability near the end of the run. Besides, there is a maximum velocity allowed [12] in case that the particles' velocities are too large to exceed the feasible region.

2.2. Velocity-Controlled Particle Swarm Optimization

PSO has the advantage of simple principle and easy implementation, but it tends to fall into local extremum and have low search precision while solving some complex problems [13]. Considering this, VCPSO is put forward to improve the optimization performance. Details of the improvement are described as follows.

(1) For PSO, it is favourable to initialize the population as uniformly as possible to cover the entire search space, which can improve the PSO's global search ability and convergence rate. Thus, this paper proposes a method of uniform initialization. For D-dimensional optimization problem, each dimension is uniformly divided into m parts, so that uniformly distributed m^D points are combined in the whole search space.

(2) In a particle swarm optimizer, it is necessary to limit the particles' positions after updating them in case they exceed the feasible region. In the process of searching for the optimal solution, the number of particles that exceed the feasible region before limiting them can reflect the group's search status to some extent. If the number is too large, it means that the distribution of the particles is too scattered and the search scope is too large. Then it is necessary to reduce the velocities of the particles by speeding up the attenuation rate of the inertia weight and reducing the maximum velocity allowed. If the number is too small, it means that the particles maybe are gathering too much to do a full search. Then it is necessary to increase the velocities of the particles by slowing down the attenuation rate of the inertia weight and increasing the maximum velocity allowed.

In addition, considering that the optimal solution may be on the boundary of the feasible region and the algorithm converges in the later search stage, the process mentioned above is only implemented in the early search stage.

The specific iterations of VCPSO are as follows:

a. Give values of all parameters needed in the algorithm, including the number of uniformly division for each dimension m, range of search area ps, range of velocity vs, fitness function *adaptfunc*, the maximum number of iterations N, and so on.
b. Uniformly initialize the population's positions $\boldsymbol{X}_i$ ($i = 1, 2, \ldots, n$).
c. Randomly initialize population velocity within the range of velocity range.
d. Calculating particle fitness using initialized population position.
e. Initialize each particle's best previous position P_i ($i = 1, 2, \ldots, n$) and the best previous position of all particles P_g by using particle fitness calculated in d-th step.
f. Update each particle's velocity and position according to Equations (1) and (2).
g. If the current iteration number does not reach 2/3 of N, then go to step h, otherwise go to step g.
h. Count the number of particles, *num*, that exceed the feasible region after updating. If *num/n* is greater than 0.5, reduce the velocities of all particles by speeding up the attenuation rate of the inertia weight and reducing the maximum velocity allowed. If *num/n* is less than 0.05, increase the velocities of the particles by slowing down the attenuation rate of the inertia weight and increasing the maximum velocity allowed. If *num/n* is somewhere between 0.05 and 0.5, keep the current search status.

i. If a dimension of a particle exceeds the upper limit, the dimension information is replaced by the upper limit value, and the lower limit value is replaced by the lower limit value. Then calculate each particle's fitness.

j. Update each particle's best previous position P_i ($i = 1, 2, \ldots, n$) and the best previous position of all particles P_g.

k. Calculate the average fitness value of the top 50% particles in the population. If the difference between the average fitness values that were respectively calculated during the current iteration and last iteration is less than 0.01 or current iterations $t >$ N, stop the iteration; otherwise go to step f.

Flow charts are shown in Figure 1.

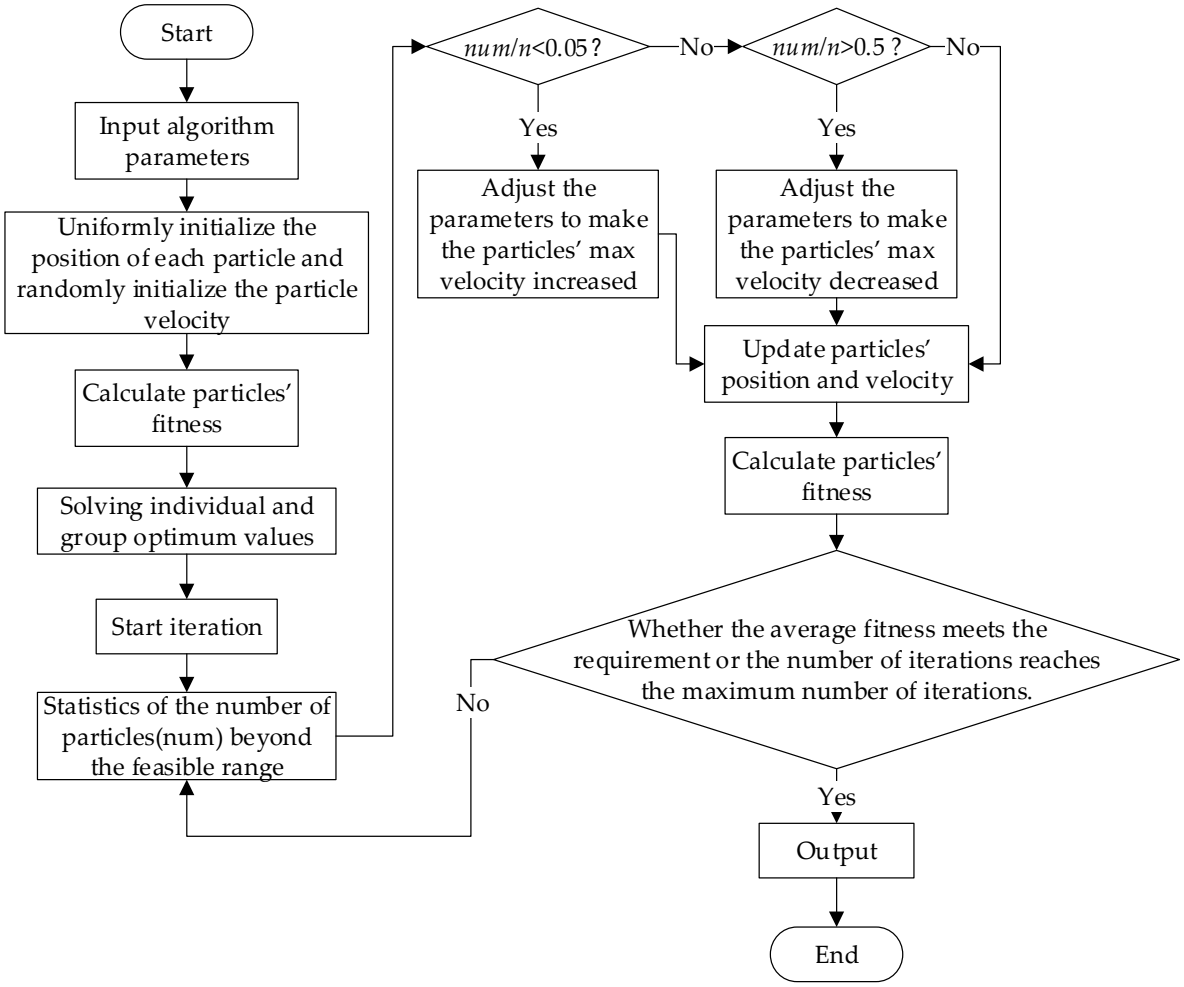


Figure 1. Flow chart of velocity-controlled particle swarm optimization (VCPSO) algorithm.

This paper chooses five test functions whose global optimums are already known in order to test the optimization performance of VCPSO. Expressions of the five test functions are shown in Equations (4)–(8). Test results of VCPSO, SPSO, and Gaussian quantum-behaved PSO (GQPSO) [14] are shown and compared in Table 1. The whole procedure runs in MATLAB. The data are the average of 100 times' results. "non" in Table 1 means that algorithm did not find the global optimum within the maximum number of iterations.

$$f_1(\boldsymbol{X}) = 30 + \sum_{i=1}^{3} [\frac{1}{10}(x_i - 20)^2 - 9\cos\frac{2\pi}{5}x_i], 1 \le x_i \le 39, i = 1, 2, 3 \quad (4)$$

$$f_2(\boldsymbol{X}) = \sum_{i=1}^{10} x_i^2, -100 \le x_i \le 100, i = 1, 2, \cdots, 10 \tag{5}$$

$$f_3(\boldsymbol{X}) = \sum_{i=1}^{10} [x_i^2 - 10\cos(2\pi x_i) + 10], -5.12 \le x_i \le 5.12, i = 1, 2, \cdots, 10 \tag{6}$$

$$f_4(\boldsymbol{X}) = \frac{1}{4000}\sum_{i=1}^{10} x_i^2 - \prod_{i=1}^{10} \cos(x_i/\sqrt{i}) + 1, -600 \le x_i \le 600, i = 1, 2, \cdots, 10 \tag{7}$$

$$f_5(\boldsymbol{X}) = \frac{\sin^2\sqrt{x_1^2 + x_2^2 - 0.5}}{[1 + 0.001(x_1^2 + x_2^2)]^2} + 0.5, -100 \le x_i \le 100, i = 1, 2 \tag{8}$$

Table 1. Comparison of test results.

Algorithm	Test Function	Population	Running Time (s)	Average Iterations	Optimal Solution
SPSO	f_1	125	0.15	75	3.0014
	f_2	1024	2.01	100	5.7421×10^{-3}
	f_3	1024	2.02	100	non
	f_4	1024	2.74	100	0.4833
	f_5	100	0.02	8	3.49620×10^{-3}
GQPSO	f_1	125	0.49	100	9.3634
	f_2	1024	1.21	98	3.9626×10^{-21}
	f_3	1024	1.99	98	19.151×10^{-24}
	f_4	1024	2.43	100	0.4608×10^{-6}
	f_5	100	0.05	100	0.07
VCPSO	f_1	125	0.10	72	3.0006
	f_2	1024	1.07	53	0
	f_3	1024	1.68	89	0.6502×10^{-3}
	f_4	1024	1.79	60	0
	f_5	100	0.01	6	0

SPSO, standard particle swarm optimization; GQPSO; VCPSO, velocity-controlled PSO.

From Table 1, it is concluded that VCPSO has better global search ability, higher search precision and a faster convergence rate than SPSO while solving the same problem. The improvements proposed in this paper effectively improve the optimization performance of PSO. In the other hand, VCPSO is much better than GQPSO in running time and average iterations. The optimal solution of VCPSO is also better than that of GQPSO in general.

3. Support Vector Machine

SVM was proposed by Vapnik and Cortes in 1995 [15]. As a kind of machine learning algorithm based on statistical learning theory, SVM has an intuitive geometric interpretation and good generalization ability [16]. It can effectively overcome the "curse of dimensionality" and avoid the local optimal solution. It can not only be applied to machine learning problems such as function fitting, but also has many unique advantages in solving small sample, nonlinear, and high-dimensional pattern recognition problems. SVM has become a hot research area in the field of machine learning [17–20].

Assuming that the sample generated from a certain probability distribution is $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, the regression problem is to find the appropriate real value function to fit these sample points and establish the corresponding relationship between input and output. Support Vector Machine (SVM) can solve regression problem as looking for a pipeline that can hold all the sample points in the sample set, and the central line of the pipeline is the regression estimation function $f(x)$.

Using ε-insensitive loss function

$$|y_i - f(x_i)| = \max\{0, |y_i - f(x_i)| - \varepsilon\} \tag{9}$$

To measure the error between the observed value y and the predicted value $f(x)$, that is, when the error between the observed value y and the predicted value $f(x)$ of $\boldsymbol{X}$ point does not exceed the small positive number given in advance, it is considered that the predicted value $f(x)$ at that point is lossless relative to the observed value y, although sometimes the predicted value $f(x)$ and the observed value y may not be completely equal. Similar to support vector classification, the following problems need to be solved:

$$\begin{aligned} &\min \tfrac{1}{2}\|w\|^2 \\ &\text{s.t. } (w{\cdot}x_i + b) - y_i \le \varepsilon, i = 1, 2, \ldots, n \\ &y_i - (w{\cdot}x_i + b) \le \varepsilon, i = 1, 2, \ldots, n \end{aligned} \tag{10}$$

By introducing relaxation variable ξ, ξ' and penalty parameter C, the above problems are transformed into:

$$\begin{aligned} &\min \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i') \\ &\text{s.t. } (w{\cdot}x_i + b) - y_i \le \varepsilon + \xi_i, i = 1, 2, \ldots, n \\ &y_i - (w{\cdot}x_i + b) \le \varepsilon + \xi_i', i = 1, 2, \ldots, n \\ &\xi_i, \xi_i' \ge 0, i = 1, 2, \ldots, n \end{aligned} \tag{11}$$

In order to reduce the number of parameters of SVM, the radial basis function (RBF) is used in this paper, the equation is as follows

$$K(x_i, x) = \exp(-\gamma{\cdot}\|x_i - x\|) \tag{12}$$

where γ is a constant greater than zero.

Its dual form is

$$\begin{aligned} \min \quad & \tfrac{1}{2}\sum_{i,j=1}^{n}(\alpha_i - \alpha_i')(\alpha_j - \alpha_j')K(x_i, x_j) \\ & +\varepsilon\sum_{i=1}^{n}(\alpha_i - \alpha_i') - y_i\sum_{i=1}^{n}(\alpha_i - \alpha_i') \\ \text{s.t.} \quad & \sum_{i=1}^{n}(\alpha_i - \alpha_i') = 0 \\ & \alpha_i, \alpha_i' \in [0, C] \end{aligned} \tag{13}$$

The final decision Equation (14) is obtained by solving the above dual problem.

$$f(x) = \sum_{i=1}^{n}(\alpha_i' - \alpha_i)K(x_i, x) + b \tag{14}$$

4. Transverse Flux Induction Heating Optimization Based on VCPSO and SVM

TFIH is composed of two iron cores and two coils, the coil is placed inside of the iron core named a heater, one heater is placed above the strip and another heater is placed below the strip, two heaters on each side are placed parallel to each other, as shown in Figure 2. By injecting alternating current into the coil of the heater, an alternating magnetic field will be produced in the iron core and strip and thus eddy current will be produced in the strip due to the alternating magnetic field. The eddy current generates heat, and thus the strip will be heated. During the induction heating process, the strip moves at a constant speed between the two heaters. The temperature distribution on the surface of the strip at the outlet of TFIH is the most important parameter to represent the performance of the TFIH. In Figure 2, the arrow shows the movement direction of the strip.

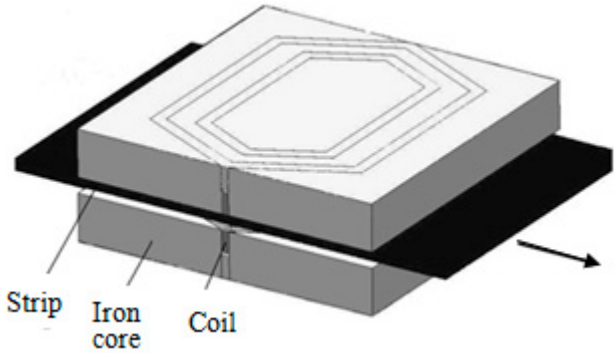


Figure 2. Model of transverse flux induction heating (TFIH) equipment.

In our previous work, both the current and frequency have been found to influence the temperature distribution of the strip. Generally, the higher the input current and its frequency, the higher the temperature distributed on the strip. Furthermore, the size of the coil plays the most important role to determine the temperature distribution of the strip as it can affect the eddy current distribution on the strip. According to our work, the hexagonal coil has the ideal coil structure for the TFIH system [21]. In this paper, the length of the coil along the width of the strip has been taken into the consideration, and the parameter of l is shown in Figure 3. In order to measure the uniformity of the temperature distribution of the strip, the average relative error of temperature is adopted in this paper. The optimization model of TFIH system is obtained and described as follows:

$$\begin{aligned} \min \quad & F(\mathbf{X}) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{T_i - T_a}{T_a}\right| \\ \text{s.t.} \quad & I_{\min} \le x_1 \le I_{\max} \\ & f_{\min} \le x_2 \le f_{\max} \\ & l_{\min} \le x_3 \le l_{\max} \end{aligned} \tag{15}$$

where $i = 1, 2, \ldots, n$. n represents the number of sampling points of the strip surface temperature at the outlet of heater.

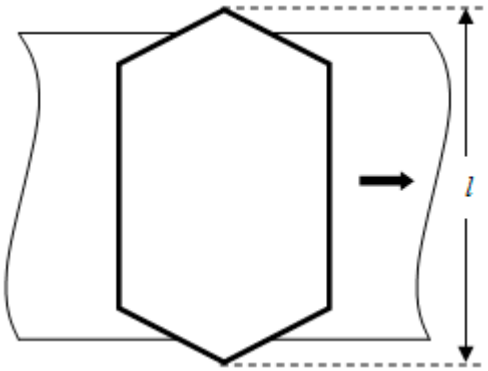


Figure 3. A hexagonal TFIH coil.

In Equation (15), $F(\boldsymbol{X})$ is the objective function that represents the average relative error of temperature. T_i is the temperature value at each sampling point. T_a is the average temperature of all sampling points.

$$T_a = \frac{1}{n}\sum_{i=1}^{n} T_i \tag{16}$$

$\boldsymbol{X} = (x_1, x_2, x_3)$ represents three design variables, x_1 is the current, x_2 is the frequency, x_3 is the length of the coil along the width of the strip which has been shown in Figure 3. $I_{\min}$ equals 300 A, $I_{\max}$ equals 1500 A; $f_{\min}$ equals 500 Hz, $f_{\max}$ equals 1500 Hz; $l_{\min}$ equals 380 mm, $l_{\max}$ equals 420 mm.

For the temperature distribution, the 3D FEM is adopted. In this paper the MagNet and ThermNet are used. In the performance analysis, the gap between the strip and heating coil equals 2 mm, and the thickness of strip is 2 mm and the width of strip is 400 mm. The strip moves at the speed of 0.05 m/s. Because TFIH is symmetrical in both horizontal and vertical directions, 1/4 model of TFIH is chosen for simulation.

In order to realize the regression prediction of transverse flux induction heating, the data of temperature average relative error under different current, frequency, and coil size obtained by finite element simulation are used as training samples of support vector machine. The training data are shown in Table 2.

Table 2. Training samples.

Input Current (A)	Input Frequency (Hz)	Coil Length (mm)	Average Relative Error of Temperature (%)
400	550	380	0.74
700	550	380	1.18
1000	550	380	1.45
1300	550	380	1.68
400	850	380	0.71
...	...	...	...
1300	1150	410	8.53
400	1450	410	7.25
700	1450	410	8.39
1000	1450	410	8.54

The flow chart of TFIH problem prediction using SVM is shown in Figure 4.

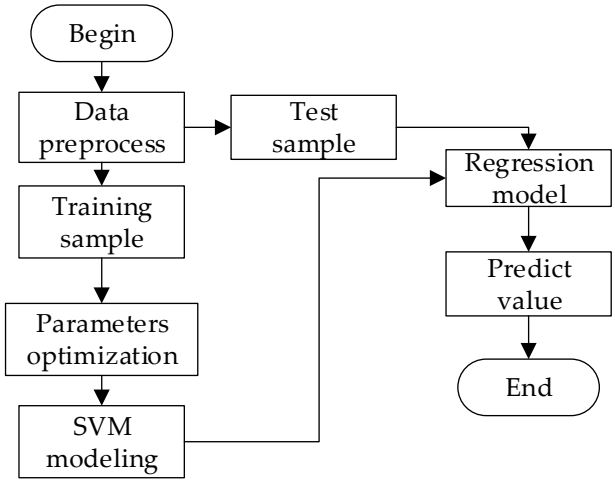


Figure 4. Flow chart of SVM used for the TFIH problem prediction.

The process of SVM modeling for TFIH is described as follows:

1. Original data processing: in order to improve the prediction accuracy, first of all to the original data format, and data will be normalized to [–1, 1]. The normalization has three purpose: (1) to solve the problem of dimension, and if the magnitude difference between different data of the same attribute is too large, the change of large number will cover up the change of decimal number; (2) to improve the convergence speed—the speed of solution can be accelerated after data normalization; (3) to optimize the parameters in the following steps—the larger the input, the better. It is likely that the value of the parameters will exceed the optimum range, and after normalization, the optimum range of the parameters can basically cover its optimum value.

2. Training samples and test samples: after data processing, a part of the original data is selected as training samples to establish a regression prediction model, and the rest as test samples to test the accuracy of the obtained prediction model.
3. Parameter optimization: the parameters here refer to a series of parameters involved in the modeling process, such as penalty factor, and parameters in the kernel function. When different parameters are selected, the prediction models will be different and the accuracy of the models will be different. Therefore, it is necessary to optimize the parameters to obtain the prediction model with the highest accuracy.
4. SVM modeling: the process of solving the corresponding convex quadratic programming problem.
5. Regression model: the regression prediction model of transverse flux induction heating problem can be obtained by the above process.
6. Finally, the predictive model is used to predict the test sample and get the predictive value. The comparison between predicted and simulated values is shown in Table 3. From the comparison of data in Table 3, it can be seen that the regression prediction model of transverse flux induction heating established by support vector machine has high fitting accuracy, and can be used to replace the finite element numerical calculation to analyze the transverse flux induction heating problem.

Table 3. Prediction results of average relative error of strip temperature at outlet of device by support vector machine.

Input Current (A)	Input Frequency (Hz)	Coil Length (mm)	Average Relative Error of Temperature (%) Finite Element Calculate Value	Average Relative Error of Temperature (%) SVM Predict Value
700	1450	380	2.41	2.47
400	550	390	0.86	1.00
700	550	400	3.91	3.56
1300	850	410	7.34	7.27

Since the calculation of nonlinear coupled electromagnetic thermal problems involved in TFIH is time consuming, the analysis results obtained by using the MagNet and ThermNet simulation is used to train SVM, thus a regression model of the TFIH problem can be obtained to replace the time consuming calculation process. The regression model is applied in the process of optimizing the TFIH problem with VCPSO. The whole process is implemented in MATLAB.

$\boldsymbol{X}$ values before and after the optimization respectively as shown in Table 4.

Table 4. $\boldsymbol{X}$ values before and after the optimization.

Optimization Stage	Input Current (A)	Input Frequency (Hz)	Coil Length (mm)
Before optimization	1000	500	400
After optimization	1197	1333	389

TFIH parameters are normally selected via trial-and-error according to the temperature requirement from strip heating treatment. $\boldsymbol{X}$ values before optimization were given by experience on the basis of many simulation analysis, the "Before optimization" parameters in Table 4 are typical of the settings in TFIH equipment.

Figures 5 and 6 show the temperature distribution on the surface of strip at the outlet of TFIH device before and after the optimization, respectively, based on the MagNet and ThermNet with the selected $\boldsymbol{X}$ values. The left boundary in both Figures 5 and 6 show the temperature distribution of the strip at the outlet. It can be seen that after the optimization, the temperature distribution on the surface of the strip after optimization is more uniform and also the average temperature rise is higher as well,

as shown on the left edge of Figure 6. The temperature distribution of half width strip at the outlet is illustrated in Figure 7. As shown, the peak to peak temperature before the optimization is about 260 °C while that after optimization is about 60 °C.

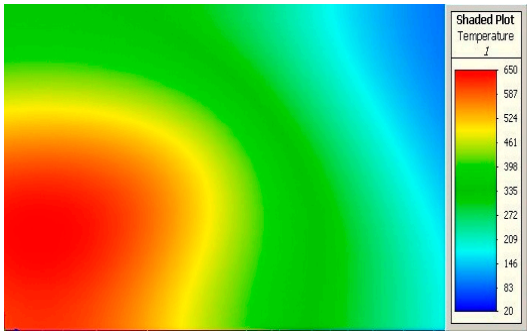


Figure 5. Before optimization.

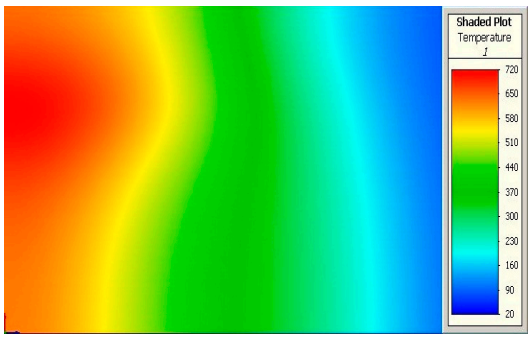


Figure 6. After optimization.

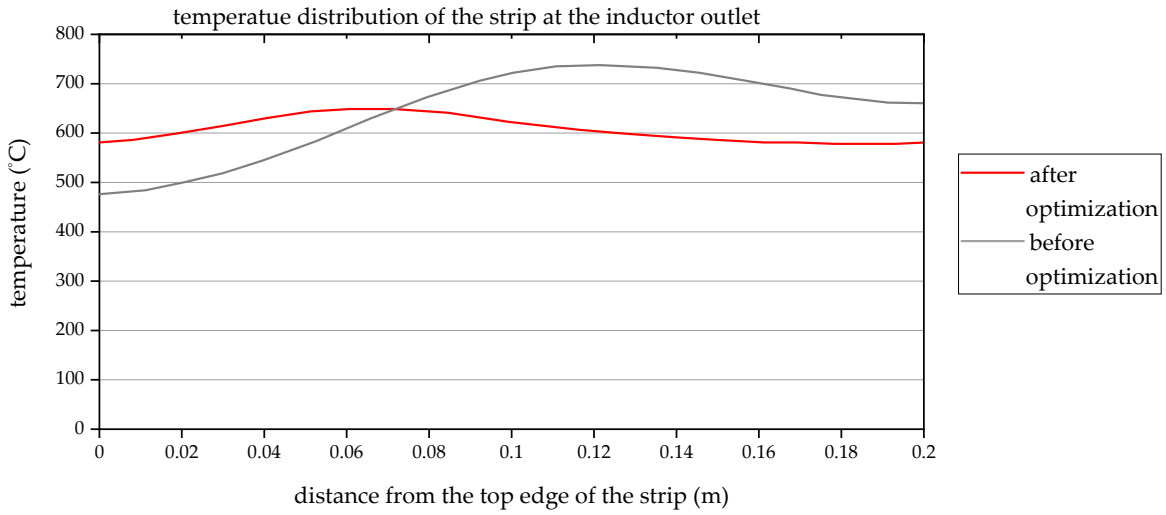


Figure 7. Comparison of temperature distribution before and after optimization.

5. Conclusions

In this paper, VCPSO is proposed based on the analysis of the distribution of the particles during the PSO optimizing process. A number of experiments have been done to compare the performance of VCPSO with SPSO. It was found that VCPSO has better global search ability, higher search precision and a faster convergence rate when compared with SPSO. Furthermore, the VCPSO combined with SVM is adopted to optimize a TFIH system. By using the proposed new method, the temperature distribution of the strip at the outlet became quite uniform. However, the number of particles increasing exponentially with the increase of the optimization parameters (mentioned in Section 2.2) is the main

disadvantage for the proposed VCPSO. Although VCSPO can achieve better optimization results, it needs more research.

Author Contributions: All authors contributed to this work through collaboration. Y.W. is the main author of this manuscript; B.L. and L.Y. assisted to establish the velocity-controlled particle swarm optimization (VCPSO); J.W., S.W., and C.L. provided some useful suggestions in the construction of paper. All authors revised and approved the publication.

Funding: This research was funded by National Natural Science Foundation of China grant number 51877065.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pang, L.L.; Wang, Y.H.; Chen, T.G. New development of traveling wave induction heating. *IEEE Trans. Appl. Supercond.* **2010**, *20*, 1013–1016. [CrossRef]
2. Wang, Y.H.; Wang, J.H.; Pang, L.L.; Ho, S.L.; Fu, W.N. An advanced double-layer combined windings transverse flux system for thin strip induction heating. *J. Appl. Phys.* **2011**, *109*, 07E511. [CrossRef]
3. Eberhart, R.C.; Dobbins, R.W.; Simpson, P.K. *Computational Intelligence PC Tools*; Academic Press: Boston, MA, USA, 1996.
4. Eberhart, R.C.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
5. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
6. Kennedy, J. The particle swarm: Social adaptation of knowledge. In Proceedings of the IEEE International Conference on Evolutionary Computation, Indianapolis, IN, USA, 13–16 April 1997; pp. 303–308.
7. Ciuprina, G.; Ioan, D.; Munteanu, I. Use of intelligent-particle swarm optimization in electromagnetics. *IEEE Trans. Magn.* **2002**, *38*, 1037–1040. [CrossRef]
8. Fukuyama, Y.; Takayama, S.; Nakanishi, Y.; Yoshida, H. In A Particle Swarm Optimization for Reactive Power and Voltage Control in Electric Power Systems considering voltage security assessment. In Proceedings of the Conference on Systems, Man, and Cybernetics, Tokyo, Japan, 12–15 October 1999.
9. Senthil Arumugam, M.; Rao, M.V.C.; Chandramohan, A. A New and Improved Version of Particle Swarm Optimization Algorithm with Global-Local Best Parameters. *J. Knowl. Inf. Syst.* **2008**, *16*, 324–350. [CrossRef]
10. Li, Z.Q.; Zheng, H.; Peng, C.M. Particle Swarm Optimization Algorithm Based on Adaptive Inertia Weight. In Proceedings of the 2010 2nd International Conference on Signal Systems, Dalian, China, 5–7 July 2010; pp. 454–457.
11. Shi, Y.; Eberhart, R.C. A modified particle swarm optimizer. In Proceedings of the IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
12. Shi, Y.; Eberhart, R.C. Parameter selection in particle swarm optimization. In Proceedings of the International Conference on Evolutionary Programming, Berlin/Heidelberg, Germany, 25–27 March 1998; pp. 591–600.
13. Shi, Y.; Eberhart, R.C. Empirical study of particle swarm optimization. In Proceedings of the 1999 Congress of Evolutionary Computation, Washington, DC, USA, 6–9 July 1999; pp. 1945–1950.
14. Coelho, L.d.S. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Syst. Appl.* **2010**, *37*, 1676–1683. [CrossRef]
15. Fu, K.; Yang, X.G.; Wang, Y.H. The inter generation statistical character self-feedback improved genetic algorithm based on SVM model. In Proceedings of the 1st International Conference on Bioinformatics and Biomedical Engineering, Wuhan, China, 6–8 July 2007; pp. 618–651.
16. Cristinini, N.; Shawe-taylor, J. *An Introduction to Support Vector Machines and other Kernel-Based Learning Methods*; Cambridge University Press: Cambridge, UK, 2000.
17. Hadchum, P.; Wongthanavasu, S. Facial Age Estimation Using a Hybrid of SVM and Fuzzy Logic. In Proceedings of the 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Hua Hin, Thailand, 24–27 June 2015.
18. Wu, J.; Yang, H. Linear Regression-Based Efficient SVM Learning for Large-Scale Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 2357–2369. [CrossRef] [PubMed]

19. Tyagi, D.; Verma, A.; Sharma, S. An Improved Method for Face Recognition using Local Ternary Pattern with GA and SVM classifier. In Proceedings of the 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Noida, India, 14–17 December 2016; pp. 421–426.
20. Zhu, H.; Liu, X.; Lu, R.; Li, H. Efficient and Privacy-Preserving Online Medical Prediagnosis Framework Using Nonlinear SVM. *IEEE J. Biomed. Health Inform.* **2017**, *21*, 838–850. [CrossRef] [PubMed]
21. Sun, Y.; Wang, Y.H.; Yang, X.G.; Pang, L.L. A novel coil shape for transverse flux induction heating. *Chin. J. Electr. Eng.* **2014**, *29*, 85–90.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).