Advance Journal of Food Science and Technology 7(10): 804-809, 2015
DOI:10.19026/ajfst.7.1977
ISSN: 2042-4868; e-ISSN: 2042-4876
© 2015 Maxwell Scientific Publication Corp.
Submitted: October 17, 2014 Accepted: December 18, 2014 Published: April 05, 2015

Research Article

Research on Demand Prediction of Fresh Food Supply Chain Based on Improved Particle Swarm Optimization Algorithm

He Wang
School of International Trade and Economics, Jiangxi University of Finance and Economics,
Nanchang 330013, China

Abstract: Demand prediction of supply chain is an important content and the first premise in supply management of different enterprises and has become one of the difficulties and hot research fields for the researchers related. The paper takes fresh food demand prediction for example and presents a new algorithm for predicting demand of fresh food supply chain. First, the working principle and the root causes of the defects of particle swarm optimization algorithm are analyzed in the study; Second, the study designs a new cloud particle swarm optimization algorithm to guarantee the effectiveness of particles in later searching phase and redesigns its cloud global optimization searching method and crossover operation; Finally, a certain fresh food supply chain is taken for example to illustrate the validity and feasibility of the improved algorithm and the experimental results show that the improved algorithm can improve prediction accuracy and calculation efficiency when used for demand prediction of fresh food supply chain.

Keywords: Demand prediction, fresh food, particle swarm optimization algorithm, supply chain management

INTRODUCTION

Nowadays, the lack of effective prediction system of supply chain demand makes it difficult to predict the practical food demand effectively and accurately and has become a crucial problem in Supply Chain Management (SCM) for many enterprises. Without effective and accurate prediction system it will makes great difficulties to guarantee the interests of the enterprises for long-term stability, particularly for supply chain management of fresh food products. For supply chain of fresh food products, various factors are involved in and may affect its demand prediction, so the factors selecting become crucial. And building an efficiency for SCDPS (Supply Chain Demand Prediction System) and using the scientific model to predict fresh food demand have close relationship with the final success of developing Supply Chain Partnership. Therefore, the research on building demand prediction system for fresh food supply chain and its scientific algorithm has great practical value and theoretical significance and it is a research hotspot in supply chain management and has great study priorities in this century (Amandle, 2010).

Up to now, the methods for predicting supply chain demand used by enterprises and researchers can be roughly divided into two categories: single prediction method and combination prediction method. The former methods are based on quantity of value, typical methods include grey prediction method, adopting neural network prediction method, time series prediction method and markov prediction method; The latter methods is based on the prediction results of multiple prediction methods and gets the final prediction results according to certain form rules. Apart from above, some other scholars also present some other prediction models, such as set pair analysis prediction and scenario analysis method and etc. the specific merits and defects of different models can be listed as follows:

- **Delphi prediction model:** Delphi model, sometimes called expert investigation model, can be used in profits prediction, sales prediction and technology prediction for long-term and new products. Its calculation flow mainly includes four steps:
    - Preparing opinion consultation table according to the goal and demand of prediction
    - Selecting some experts related to the prediction object as consultation persons
    - Consulting experts' evaluation repeatedly
    - **Making final prediction results:** The merit of Delphi model is which can drawing on the evaluation and prediction of the experts and masses and the method can be beneficial to a comprehensive prediction. But the defects of Delphi model are that the method is lack of objective prediction and the lower prediction reliability (Tebby, 2011)
- **Business personnel prediction model:** The model convenes all the business personnel of every logistics stage to estimate the demand of certain

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: http://creativecommons.org/licenses/by/4.0/).

logistics and then process the estimated data of everyone comprehensively and forms the final prediction results. The logistics stages include purchasing, planning, transporting and warehousing. The merit of this model is that the model know the best about the specific market dynamics and logistics demand conditions and the specific conditions and prediction results has close relationship with the practical situation. But the defects of this model is that and its prediction always neglects market demand change trend and the entire economic situation and the quality of some business personnel is low (Luong and Phien, 2012)

- **Time series model:** This model make use of certain mathematical methods and the sale volume data of past periods or other level-one data with existing demand data to predict future demand and change trend. Mainstream time series methods for supply chain demand prediction has three kinds: index correction method, moving average method and epitaxial smoothing technique (Disney and Towill, 2006):
  - Index correction method uses actual data of past periods to predict final demand and the model is mainly applicable to predict the supply chain which has low seasonal change
  - **Moving average model:** The model assume that in time series each point moving averagely is arithmetic average or weighted average of a series of continuity points. In order to decrease seasonal influence or irregular effects, several data points are requires to be chosen when simple moving average model is used for predicting. The moving average method is widely used for its simple process flow and higher prediction results relative, but the model is only can be used for predicting the demand with no obvious descending or ascending of value change and cannot used for long-term prediction (Duc and Drezner, 2014)
  - **Epitaxial smoothing technique:** In this model the effects of trend value and seasonal fluctuation on prediction results are considered, so the model has higher prediction precision but is sensitive to seasonal change (Feng and Ma, 2013; Zhu and Tang, 2011)
- **BP neural network for supply chain demand prediction:** Lots of variables are involved in original BP neural network algorithm and the variables are used to confirm the structure of BP algorithm used and BP algorithm weighted the value of fixed structure (including node transformation function and connection weight). Based on the application experience, different prediction precision will affect the number of nodes and neuron of different network layer, so prediction system based on BP algorithm can be established dynamically as user required (Fang *et al*., 2011)

PSO (Particle Swarm Optimization) algorithm when used for supply chain demand prediction has fairly good prediction accuracy, but the method has the question of slow convergence speed which limits the practical application of the algorithm. The study improves PSO algorithm to overcome the defects of slow convergence speed of original BP neural network algorithm and presents a new cloud PSO algorithm for demand prediction of fresh food supply chain.

MATERIALS AND METHODS

Working principle of particle swarm optimization algorithm: Particle Swarm Optimization algorithm, based on swarm intelligence, is an evolutionary calculation method simulating flying behavior of birds. The working principle of PSO is the potential solution of each optimization problem like a bird in search space which is called "particle". Each particle in the algorithm has an adaptive value which is determined by optimization function and a speed vector which is determined by their flying distance and direction; all the particles should follow the search of existing optimal particle in solution space. PSO algorithm is started as a number of random particles and then through iteration to find optimal solution for each particle. In every iteration, each particle should update its optimal solution by follow and optimize the two extreme values, in which one extreme value is the best optimal solution which helps other particles to finds their current moment which is called individual best value. The other best solution is for the entire group to find their current moment, called best global value (Gao and Zhang, 2014).

Suppose that a group is formed by m particles forming in D-dimensional target search space, in the group the ith particle can be expressed as a D-dimensional vector and expressed as $\vec{x}_i = (x_{i1}, x_{i2}, \dots x_{id})$. In D-dimensional, the location of the ith particle search space is a potential solution. We can get its adaptive value by substituting $\vec{x}_i$ into a target function, measuring the weakness and strength of $\vec{x}_i$ according to the size of its adaptive value. The flying speed of the ith particle can be expressed as a D-dimensional vector also, expressed as $\vec{v} = (v_{i1}, x_{i2}, \dots v_{id})$ (Gao and Zhang, 2014) and $\vec{p}_{id} = (p_{i1}, p_{i2}, \dots p_{id})$ can be used to express the optimal location searched till now of the ith particle and $\vec{p}_{gd} = (p_{g1}, p_{g2}, \dots p_{gd})$ is used to express the optimal location searched till now of the entire particle swarm. Original PSO algorithm carries out the calculation of each particle by Eq. (1) and (2) followed (Mieach, 2012):

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

$$x_{id}^{t+1} = x_{id}^{t} + \alpha v_{id} \qquad (2)$$

In Eq. (1) and Eq. (2), $i = 1,2,3......m, d = 1,2,3......D$, c_1 means learning factors c_1 and c_2 means nonnegative constants respectively; r_1 and r_2 means random numbers and their values should follow in the uniform distribution of (0, 1). In the Eq. (1), ϖ is called momentum coefficient and means nonnegative number, used for controlling the influence of previous speed on current speed; Generally speaking, the greater ϖ means the stronger global searching ability and the bigger influence of previous speed; and vice versa, the smaller ϖ means the weaker global searching ability and the less influence of previous speed. So the local minimum value can be able to be jumped out by adjusting the size of ϖ. The variable α is use and called constraints factor to control the weight of speed. According to specific problems, the condition for terminating iteration is selected as optimal location particle swarm or maximum iterations searched up until the desired minimum adaptive threshold is met (Mieach, 2012).

Designing cloud particle swarm optimization algorithm: Cloud PSO algorithm adopts updating form of basic PSO speed and position to carry out global optimization searching. However, after searching, the combination of cloud model and particles makes the particles obtain randomness and stabilization tendency, but part of the particles fail to carry out global searching within the whole searching scope due to losing the diversity of population. Hence, the thesis adopts cloud self-adaptive inertia weight for adjustment.

ICPSO (Improved Cloud Particle Swarm Optimization) guarantees the effectiveness of particles in later searching phase. The improvement measures make use of the crossover operator of cloud genetic algorithm to carry out crossover operation on the particles of cloud PSO, then making the particles after crossover carry out chaos immigration searching, thus making the part of particles falling into local extreme value achieve global best, improving the algorithm optimization accuracy.

In the basic PSO, assuming there are N particles forming a population in a D-dimensional target searching domain, X_i indicates the position of the ith particle, $X_i = (x_{i1}, x_{i2}, ...x_{id})$, P_i indicates the historical optimal position of the ith particle, P_g indicates the optimal position searched by the entire particle up till now, $P_i = (p_{i1}, p_{i2}, ...p_{id})$, $P_g = (p_{g1}, p_{g2}, ...p_{gd})$, V_i indicates the flight speed of the ith particle, $V_i = (v_{i1}, v_{i2}, ...v_{id})$, particles adjust their position and speed according to Eq. (3) and (4):

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_1 r_1 (p_{id}^{k} - x_{id}^{k}) + c_2 r_2 (p_{gd}^{k} - x_{gd}^{k}) \qquad (3)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad (4)$$

Speed adjustment rules are shown in Eq. (5):

$$\begin{cases} v_{id} = v_{\max}, & v_{id} \succ v_{\max} \\ v_{id} = -v_{\max}, & v_{id} \le v_{\max} \end{cases} \qquad (5)$$

In the basic PSO algorithm, the weight is calculated according to iterative formula; as the larger inertia weight ω helps jumping out of local optimization, for global optimization searching, the smaller inertia weight ω helps local optimization, accelerating algorithm convergence, thus clustering particle swarm via evolutionary strategy of different weights.

Assuming the size of particle swarm as N, in the t times of iteration, the fitness value of particle X_i is f_i, and the average fitness of particles is Eq. (6):

$$f_{avg} = \frac{1}{N} \sum_{i=1}^{N} f_i \qquad (6)$$

Averaging the fitness value that fitness value is superior to f_{avg} to obtain f'_{avg}, averaging the fitness value that fitness value is inferior to f_{avg} to obtain f''_{avg}, the fitness value of optimal particle is f_{min}, dividing the group into 3 subgroups, adopting different weights to generate strategy which is:

- If f_i is superior to f'_{avg}, the fitness value of particles is smaller, closer to optimal solution, thus adopting smaller inertia weight, evolutionary strategy adopting “social model”, accelerating global convergence speed, ω is valued as 0.2.
- If f_i is inferior to f''_{avg} and the fitness value of particles is larger, evolutionary strategy of optimal solution adopts “cognitive model”, making the particles with poor performance accelerate the rate of convergence, further, adopting larger inertia weight, ω is valued as 0.9.
- If f_i is superior to f''_{avg} but inferior to f'_{avg}, the fitness value of particles at this time is moderate, inertia weight adopting cloud self-adaptive inertia weight, evolutionary strategy adopting “complete model”.

The generating algorithm of cloud self-adaptive inertia weight ω adopts Eq. (7) to (11) successively:

$$Ex = f_{avg} \tag{7}$$

$$En = (f'_{avg} - f_{\min}) / c_1 \tag{8}$$

$$He = En / c_2 \tag{9}$$

$$En' = normnd\,(En, He) \tag{10}$$

$$\omega = 0.9 \sim 0.5 \times e^{\frac{-(f_i - Ex)^2}{2(En')^2}} \tag{11}$$

With the decrease of the fitness of cloud particle, we can know from limit theorem, $0 \prec \frac{-(f_i - Ex)^2}{2(En')^2} \prec 1$, thus making $\omega \in [0.4, 0.9]$; we can know from the value formula of ω, ω will be decreased with the decrease of fitness value of cloud particle, thus realizing optimal particle obtaining smaller value of ω (Mieach, 2012).

Crossover operation of cloud particle: Under the condition of large scale of population, while solving multimodal function, as cloud particle algorithm, on the condition of searching large scale of population, in later phase of searching, is easy to fall into local optimization, thus failing to find global optimal solution, the reason of which is that this part of particles have lost the diversity of population, crossover operator in cloud genetic algorithm is adopted to carry out particle crossover operation for such part of particles losing diversity, for the purpose of generating new offspring particles to substitute the position of old particles.

The main idea of crossover is to introduce the crossover operation in cloud genetic algorithm in the process of basic cloud particle swarm iteration; users define a crossover probability to give it to the cloud particle and the probability is irrelevant to the fitness value of cloud particle. In the process of iteration, only looking for crossover probability, choosing a certain quantity of particles to put them into a particle pool; these cloud particles crossover with each other randomly and generate filial generation, with quantity unchanged, substituting parent particles with filial generation particles to guarantee the unchanged particles quantity of population.

After the crossover cloud particles complete initialization, update the position and speed of particles according to Formula 1 and 3; the crossover strategy is:

- Sorting all the cloud particles according to high and low fitness in each iteration process, making half of particles with high fitness to directly enter the next generation and the left half of particles enter crossover-waiting area.
- Randomly generating crossover position $C_i \in [1, D]$, carrying out crossover on the $C_i \sim D$ position of the i^th^ particle and the $N \times 3/2 + 1 - i$ ^th^ particle; while crossover, choosing certain crossover probability and the crossover probability in the cloud genetic algorithm is adopted here.

Use a and b to indicate the pointer of the chosen two parental individuals; calculation formula using cloud genetic operator crossover is shown in Eq. (12):

$$\begin{cases} Child_1(X_j = p_j \cdot parent_1(X_j) + (1.0 - p_j) \cdot parent_2(X_j) \\ Child_2(X_j = p_j \cdot parent_2(X_j) + (1.0 - p_j) \cdot parent_1(X_j) \\ Child_1(V) = \dfrac{parent_1(V) + parent_2(V)}{|parent_1(V) + parent_2(V)|} |parent_1(V)| \\ Child_2(V) = \dfrac{parent_1(V) + parent_2(V)}{|parent_1(V) + parent_2(V)|} \end{cases} \tag{12}$$

Building chaos immigration: In the theory of biological evolution, one of the effective methods for maintaining population diversity is immigration. Bring in certain scale of excellent individuals of the same species from outside the population to substitute bad ones in the original population and participate in the mating and reproduction of the population to guarantee the quality of the population and avoid genetic lesions and recession due to inbreeding.

As the features of normal cloud model are randomness and stabilization tendency. Randomness can keep the diversity of individuals, thus avoiding searching falling into local extremum; and stabilization tendency can well protect superior individuals, thus carrying out self-adaptive positioning for global extremum.

Single cloud genetic algorithm is low in calculation efficiency while reactive power optimization and slow in rate of convergence, thus bringing in chaos immigration. The specific operation is to carry out chaos immigration searching on part of individuals after the population completes one evolution (choosing, crossover, mutation) to obtain optimal individual with larger fitness value, leading population evolution. As the point near optimal point is lower, 3 individuals whose fitness value closer to optimal individual participate in chaos immigration searching.

Chaos is the nondeterministic and unpredictable motion state obtained by certain linear equation; variable appearing chaos state in nonlinear equation is called chaotic variable. Mapping the potential solution of the function to be optimized into chaotic channel; through chaotic iteration, we can make optimization searching process have the ability to jump out of local optimal solution. Such searching mechanism is called chaotic local searching.

Chaotic motion has ergodicity, randomness and regularity, which can traverse all the states without repetition according to self rule within certain range. Compared with other power systems generating chaotic change, Logistic mapping is simple and small in calculated quantity, thus mapping is as shown in Eq. (13):

$$x_{k+1} = \mu x_k (1 - x_k) \qquad 0 \le x_0 \le 1 \tag{13}$$

In which, μ is controls parameter, x_k is variable, $k = 0,1,2,...$. When $\mu = 4$, the sequence generated by Logistic mapping appears chaotic state feature.

To sum up, chaos is the inherent randomness possessed by certain nonlinear system, which has the following characteristics:

- Extremely sensitive to initial conditions; in the long run, the future behavior of chaotic system is unpredictable.
- Chaos shows strange attractor in state space, having the self-similar structure of infinite nest in the finite phase space geometry.
- The frequency spectrum of chaotic system is continuous, similar to that of noise, having continuous broadband.
- Lyapunov index of chaotic system has one number larger than 0 at least.

While Cloud PSO carries out reactive power optimization, later phase of iteration generates "sluggishness" motion, making each cloud particle lost population diversity, so each control variable obtained by the reactive power optimization of the algorithm is easy to fall into local solution region, therefore, in the later phase of searching, cloud particles adopt chaos immigration operation, making cloud particles with small population diversity able to participate in reactive power optimization.

The specific steps for building immigration population:

- Initialization of chaotic variable. Iteration times k = 0, carry out linear transformation on the value of each variable of 3 individuals after cloud genetic iteration, mapping decision variable x^k_j according to Formula 11, making each variable of each individual become the value among (0, 1), as the initial value of chaotic variable, current solution recorded as $x_c = (x_{10}, x_{20}, ..., x_{n0})$, cx^k_j is calculated according to Eq. (14):

$$cx_j^k = \frac{x_j^k - x_{\min,\,j}}{x_j^k - x_{\max,\,j}} \qquad j =,1,2,...,\ n \tag{14}$$

- Generate n (0, 1) distributed chaotic variable new sequence λ_j with Formula 11, $j =,1,2,...,\ n$, n is the number of control variable, setting chaos searching times as N, generation kept by optimal individual in chaotic searching being N_1.
- Transforming chaotic variable to allowable solution space $cx_j^k = a_j + \lambda_j^k (b_j - a_j)$, in which a_j and b_j are the upper and lower limits of corresponding control variables and calculating the value of current target function f (x^k).
- If $f(x^k) \prec f^*$, keep existing optimal solution, current solution and corresponding optimal target function value, f^* being existing optimal value.
- If f^* keeps unchanged after the searching of N_1 step, finish chaotic searching and return to cloud genetic iteration; otherwise, continue to search until meet the times of chaotic searching.

After adopting chaos immigration operator, make cloud particle swarm be able to keep good population diversity within searching domain and let more particles be able to participate in searching motion.

RESULTS AND DISCUSSION

With C programming language, the improved PSO algorithm is realized in the study. Based on investigation of certain fresh food product enterprises, taking its demand of four quarters in the past 10 years from 2004-2013 for experimental sample to carry out demand prediction and model application research. Considering the prediction algorithm research is the focus of this paper, so the prediction factor analysis is omitted here and the specific factor used in this study can see the reference.

Table 1: Some prediction results of the improved algorithm

Year	Quarter	Actual demand (unit)	Model prediction demand (unit)	Relative error (%)
2004	1	6603	6809	3.11
2005	2	6898	7001	1.49
2006	3	7139	7287	2.07
2007	4	7698	7764	0.09
2008	1	7976	8243	3.34
2009	2	8689	8776	1.00
2010	3	8876	8981	1.18
2011	4	9108	9309	2.20
2012	1	9801	9986	1.89
2013	2	10189	10279	0.08

Table 2: The prediction performance comparison of different algorithms

	Moving average algorithm	Ordinary BP algorithm	The improved algorithm
The overall prediction error	15.00%	5.31%	2.37%
The prediction error of the first quarter	16.33%	6.76%	3.99%
The prediction error of the second quarter	14.78%	4.99%	1.76%
The prediction error of the third quarter	15.11%	4.92%	2.07%
The prediction error of the fourth quarter	13.79%	4.58%	1.67%
Time consumption (sec)	10	462	10

Part of experimental prediction results can be seen in Table 1 and 2. In which, some prediction results realized by the improved algorithm in the study were chosen on a random and the specific results can seen in Table 1 and 2 shows the realization performance of different algorithm, including moving average algorithm (time series method) (Disney and Towill, 2006), ordinary BP neural network algorithm (Fang *et al*., 2011) and the improved algorithm in the paper. The experiment is carried out through the same PC calculation platform in the practical application of above algorithms. The parameters of PC platform can be listed as follows. Hardware part: Dell Poweredge R710, E5506 processor, 4G memory, 500G hard disk; Software part: C programming language environment, operating system is Windows XP.

CONCLUSION

This study, through improving PSO algorithm making use of the advantage in high system prediction accuracy, also overcoming the actual defect in poor algorithm convergence, advances a new system supply chain demand predication model based on BP neural network algorithm and the experimental results show that the prediction results of the improved model are satisfactory. The model in this study has the following superiorities compared with other methods:

- Through self-study on samples involving in the comparison, BP neural network algorithm structure can be decided, repeatedly iterating according to the criterion of optimal training, constantly adjusting BP neural network algorithm structure, until reaching a relatively stable status, thus, the utilization of that method eliminates many human factors, helping to ensure the objectiveness of the evaluation results.
- High accuracy, able to make system error reach the requirement of any accuracy with convergence.
- Good dynamics, self-study and dynamic tracking ability will be stronger with the progress of time and the increase of samples involved in comparison. Hence, there is certain practical application value in that method.

REFERENCES

Amandle, S., 2010. A new method demand prediction for supply chain management. J. Manage. Sci., 13(1): 29-37.

Disney, S.M. and D.R. Towill, 2006. On the equivalence of control theoretic, differential, and difference equation approaches to modeling supply chains. Int. J. Prod. Econ., 101(4): 194-208.

Duc, K. and R. Drezner, 2014. The quantifying bullwhip effect in a simple supply chain: The impact of forecasting, lead times and information. Manage. Sci., 46(2): 46-53.

Fang, S., X. Liu and W. Xiong, 2011. Multi-robot task allocation based on BP neural network algorithm. Ind. Eng. J., 19(5): 124-133.

Feng, Y. and J. Ma, 2013. Nonlinear method research on demand forefoundry for supply chain. J. Beijing Inst. Tech. Soc. Sci. Edn., 10(5): 82-86.

Gao, G. and X. Zhang, 2014. An novel PSO algorithm for teaching evaluation. J. Educ. Manage., 18(12): 1134-1146.

Luong, L. and M. Phien, 2012. Measure of bullwhip effect in supply chains: the case of high order autoregressive demand process. Eur. J. Oper. Res., 183(2): 197-209.

Mieach, S., 2012. An effective clustering algorithm with PSO. J. Manage. Technol., 11(4): 24-31.

Tebby, J., 2011. Analyzing the state space property of echo state networks for chaotic system prediction. J. Bus. Manage., 10(11): 1412-1417.

Zhu, J. and M. Tang, 2011. A measure of bullwhip effect in supply chains with a mixed autoregressive-moving average demand process. Eur. J. Oper. Res., 182(11): 652-662.