**Mathematics and Computer Science**
2016; 1(3): 61-65
http://www.sciencepublishinggroup.com/j/mcs
doi: 10.11648/j.mcs.20160103.15

# Least Absolute Integral Method of Data Fitting Based on Algorithm of Simulated Annealing and Neural Network

**Maolin Cheng**

School of Mathematics and Physics, Suzhou University of Science and Technology, Suzhou, China

**Email address:**
ustscml@163.com

**To cite this article:**
Maolin Cheng. Least Absolute Integral Method of Data Fitting Based on Algorithm of Simulated Annealing and Neural Network. *Mathematics and Computer Science*. Vol. 1, No. 3, 2016, pp. 61-65. doi: 10.11648/j.mcs.20160103.15

**Received**: September 5, 2016; **Accepted**: September 18, 2016; **Published**: October 9, 2016

**Abstract:** There are many methods related to data fitting, and each method has its distinctive features. The article discusses the method of data fitting function under integral criterion. Since the estimate fitting parameters are complicated, the article combines algorithm of simulated annealing and neural network algorithm to solve the integral with neural network algorithm and solve the unknown parameters with simulated annealing algorithm. By case analog computation of household per capita consumption expenditure of urban and the rural residents in China, it proves that the combination of simulated annealing algorithm and neural network algorithm has strong reliability and high accuracy in terms of new method for least absolute integral data fitting.

**Keywords:** Data Fitting, Simulated Annealing, Neural Network, Algorithm, Least Absolute Integral Method

## 1. Introduction

There are n groups of observational data $(x_i, y_i), i = 0,1,\cdots.n,$ and they are the values of n nodes of $f(x)$.

The form of general linear regression model is $y_i = g(x_i, \beta) + \varepsilon_i$, where $g(x, \beta)$ is a fit function, which could be a linear function or a nonlinear function. $\beta = (\beta_1, \beta_2, \cdots, \beta_p)$ is a p-dimensional parameter vector and $\varepsilon$ is a random error variable which meets

$$\varepsilon \sim N(0, \sigma^2), \varepsilon_i = y_i - g(x_i, \beta)$$

There are three frequently-used $\varepsilon$ norms:

$$\|\varepsilon\|_1 = \sum_{i=0}^{n} |\varepsilon_i| \tag{1}$$

$$\|\varepsilon\|_2 = (\sum_{i=0}^{n} \varepsilon_i^2)^{1/2} \tag{2}$$

$$\|\varepsilon\|_\infty = \max_{0 \le i \le n} |\varepsilon_i| \tag{3}$$

Apparently, $\|\varepsilon\|_1 \to \min$ pertains to least absolute method; $\|\varepsilon\|_2 \to \min$ pertains to least square method; while $\|\varepsilon\|_\infty \to \min$ pertains to regression fitting under the least most rule.

Since we have regarded the observational data $(x_i, y_i), i = 0,1,\cdots.n, x_0 = a, x_n = b,$ as the points in the interval $[a,b]$ defined by $f(x)$, we assume $f(x)$、$g(x)$ are all continuous functions. Actually what we need is only integrabel and the fitting function $g(x)$ we are looking for are not discrete points. Due to $e(x) = f(x) - g(x)$, and the definition is $\|e\|_p = (\int_a^b |e(x)|^p)^{\frac{1}{p}} = (\int_a^b |f(x) - g(x)|^p)^{\frac{1}{p}}$.

Thus the parameters in $g(x)$ have been determined, let $\|e\|_p \to \min$.

In general, $p$ equal to 1, 2, or $\infty$. Since we hereby want to study the situation in case $p = 1$, namely

$$E(\beta) = \int_a^b |e(x)| dx = \int_a^b |f(x) - g(x, \beta)| dx$$

reaches the minimum. This method is referred to as least absolute integral method in this article and the geometric

meaning is the minimum area enclosed by curves of $f(x), g(x)$ within the interval of $[a,b]$. Since the $f(x)$ is unknown and only the values at the nodes are available, in addition, the integrand is an absolute value function, therefore the commonly used methods for seeking extreme value are invalid. In case the $f(x)$ is substituted by interpolating function, the calculation of the minimal value will be complicated. Especially, in case the $g(x)$ is an essential nonlinearity function, the calculation of the minimal value will be more complicated. Consequently, the article provides simulated annealing and neural network algorithms to find the solution to the integration with neural network algorithm [1-5]. It is feasible to integrate the functions which are complicated and hard to be integrated as well as the functions which only have a series of data without clear analytic expressions. The unknown parameters can be solved with revised simulated annealing algorithm [6-10]. The validity, reliability and accuracy of this method have also been proved by computation example.

# 2. Method and Model of Simulated Annealing and Neural Network Algorithm

### *2.1. Integrate with Neural Network Algorithm*

#### *2.1.1. Trigonometric Function Neural Network Model*

If $\beta$ is given (the $\beta$ optimizing process is derived from simulated annealing algorithm—refers to 2. 2), then,

$$E = \int_a^b |e(x)|\,dx = \int_a^b |f(x) - g(x)|dx$$

has been confirmed and here the calculation of integral is done by using trigonometric function neural network model. Since the integrand is an error function $|e(x)|$ and it has the fluctuation characteristic of little variation, it can be approximated by superimposed trigonometric functions. Therefore, the trigonometric function neural network model is mentioned in the article [11-15], the integral value is approximated by output integration of the neural network based on trigonometric functions basis funciton.

In consideration of possible cycles, the fluctuation series could be expressed by

$$\delta(x) = \sum_{k=0}^{m} (a_k \cos\frac{2\pi}{n}kx + b_k \sin\frac{2\pi}{n}kx) \quad , \ (n = 2m),$$

therefore, the hidden neuron excitation function is

$$c_k(x) = \cos(\frac{2\pi}{n}kx), s_k(x) = \sin(\frac{2\pi}{n}kx).$$

The number of hidden neurons is $m+1$.

The excitation matrix is

$$C(x) = (c_0(x), c_1(x), \cdots, c_m(x)),$$
$$S(x) = (s_0(x), s_1(x), \cdots, s_m(x)).$$

Set the weight matrix as $A = (a_0, a_1, \cdots, a_m), B = (b_0, b_1, \cdots, b_m)$.

Then the output of neural network:

$$\delta(x) = \sum_{k=0}^{m} [a_k c_k(x) + b_k s_k(x)] = AC^T + BS^T$$

The error function: $\rho(k) = |e(x_k)| - \delta(x_k), \quad (k = 0,1,\cdots,n)$.

The number of samples is $n+1$, $e(x_k) = f(x_k) - g(x_k)$, set the error matrix as $W = [\rho(0), \rho(1), \cdots, \rho(n)]$, the performance index is

$$J = \frac{1}{2}\sum_{k=0}^{n} \rho^2(k) = \frac{1}{2}\|W\|_2^2$$

where $\|\cdot\|_2^2$ is the square of Euclidean norm. Weight adjustment:

$$\Delta B = -\mu\frac{\partial J}{\partial B} = -\mu\frac{\partial J}{\partial \rho(k)}\frac{\partial \rho(k)}{\partial \delta(x_k)}\frac{\partial \delta(x_k)}{\partial B} = \mu\rho(k)S(x_k)$$

$$A(k+1) = A(k) + \Delta A = A(k) + \mu\rho(k)C(x_k)$$

$$B(k+1) = B(k) + \Delta B = B(k) + \mu\rho(k)S(x_k)$$

where $\mu$ is the learning rate and $0 < \mu < 1$.

#### *2.1.2. Convergence Conditions of Trigonometric Function Neural Network Algorithm*

Learning rate has significant influence on the convergence of trigonometric function neural network algorithm. Over low learning rate may slow down the convergence speed of trigonometric function neural network algorithm, while increasing the computation volume. By contrast, over high learning rate may lead to oscillation, rather than convergence. In order to ensure the absolute convergence of neural network algorithm, in the following, convergence condition of trigonometric function neural network algorithm is listed, acting as a theoretical basis for selection of learning rate.

Set $\mu$ as the learning rate, Lyapunov function is defined by

$$V(k) = \frac{1}{2}\rho^2(k)$$

Then, $\Delta V(k) = \frac{1}{2}\rho^2(k+1) - \frac{1}{2}\rho^2(k)$,

$$\rho(k+1) = \rho(k) + \Delta\rho(k)$$
$$= \rho(k) + \left[\frac{\partial \rho(k)}{\partial A}\right]^T \Delta A + \left[\frac{\partial \rho(k)}{\partial B}\right]^T \Delta B.$$

Moreover,

$$\Delta\rho(m)=\left[\frac{\partial\rho(k)}{\partial A}\right]^T\Delta A+\left[\frac{\partial\rho(k)}{\partial B}\right]^T\Delta B$$
$$=-\mu\rho(k)\left[\frac{\partial\rho(k)}{\partial A}\right]^T\frac{\partial\rho(k)}{\partial A}$$
$$-\mu\rho(k)\left[\frac{\partial\rho(k)}{\partial B}\right]^T\frac{\partial\rho(k)}{\partial B}$$
$$=-\mu\rho(k)\left[\left\|\frac{\partial\rho(k)}{\partial A}\right\|_2^2+\left\|\frac{\partial\rho(k)}{\partial B}\right\|_2^2\right],$$

where $\|\cdot\|_2^2$ is the square of Euclidean norm, so that:

$$\Delta V(k)=\Delta\rho(k)[\rho(k)+\frac{1}{2}\Delta\rho(k)]$$
$$=\rho^2(k))\left[\left\|\frac{\partial\rho(k)}{\partial A}\right\|_2^2+\left\|\frac{\partial\rho(k)}{\partial B}\right\|_2^2\right]$$
$$\times\left[-\mu+\frac{1}{2}\mu^2\left\|\frac{\partial\rho(k)}{\partial A}\right\|_2^2+\left\|\frac{\partial\rho(k)}{\partial B}\right\|_2^2\right].$$

In order to make the neural network algorithm convergent, there shall be $\Delta V(k)<0$, and the following inequality must be true:

$$-\mu+\frac{1}{2}\mu^2\left\|\frac{\partial\rho(k)}{\partial A}\right\|_2^2+\left\|\frac{\partial\rho(k)}{\partial B}\right\|_2^2<0$$

As $\mu>0$, i.e.:

$$\mu<\frac{2}{\left\|\frac{\partial\rho(k)}{\partial A}\right\|_2^2+\left\|\frac{\partial\rho(k)}{\partial B}\right\|_2^2}.$$

As

$$\frac{\partial\rho(k)}{\partial A}=\frac{\partial\rho(k)}{\partial\delta(x_k)}\frac{\partial\delta(x_k)}{\partial A}=-C(x_k),$$
$$\frac{\partial\rho(k)}{\partial B}=\frac{\partial\rho(k)}{\partial\delta(x_k)}\frac{\partial\delta(x_k)}{\partial B}=-S(x_k).$$

Hereby

$$\left\|\frac{\partial\rho(k)}{\partial A}\right\|_2^2+\left\|\frac{\partial\rho(k)}{\partial B}\right\|_2^2$$
$$=\|-C(x_k)\|_2^2+\|-S(x_k)\|_2^2$$
$$=m+1.$$

So that, when learning rate is set to be $0<\mu<\frac{2}{m+1}$, the neural network algorithm would be convergent.

### *2.1.3. Calculation of Integral Based on Trigonometric Function Neural Network Algorithm*

Since we know the weight of neural network $a_k, b_k$ and the integral value will be:

$$E=\int_a^b|e(x)|dx\approx\int_a^b\delta(x)dx=\int_a^b\sum_{k=0}^{m}(a_k\cos\frac{2\pi}{n}kx+b_k\sin\frac{2\pi}{n}kx)dx$$
$$=\int_a^b[a_0+\sum_{k=1}^{m}(a_k\cos\frac{2\pi}{n}kx+b_k\sin\frac{2\pi}{n}kx)\ ]dx$$
$$=(b-a)a_0+\int_a^b\sum_{k=1}^{m}(a_k\cos\frac{2\pi}{n}kx+b_k\sin\frac{2\pi}{n}kx)\ dx$$
$$=(b-a)a_0+\frac{n}{2\pi}\sum_{k=1}^{m}\frac{1}{k}\left\{\begin{matrix}a_k[\sin(\frac{2\pi kb}{n})-\sin(\frac{2\pi ka}{n})]\\+b_k[\cos(\frac{2\pi ka}{n})-\cos(\frac{2\pi kb}{n})]\end{matrix}\right\}.$$

### *2.2. Estimated Parameters of Simulated Annealing Algorithm*

#### *2.2.1. The General Steps of Simulated Annealing Algorithm*

The simulated annealing algorithm i.e. SA algorithm [16-20] was put forwarded by Kirkpatrick and the annealing process of solid was made analogy with combinatorial optimization problems. The purpose of this method is to seek the optimal solution of function or approximate optimal solution by utilizing the method of achieving minimum energy balance during annealing process. The evaluation function is equivalent to energy $E$ while the predefined one set of $\beta$ parameters correspond to a physical state. The control parameters during the whole optimizing process are equivalent to the temperature $T$ during annealing process, while the SA algorithm can be inverted for optimizing. The concrete steps of algorithm include:

(1) Set the initial temperature $T_0$ and minimum temperature $T_f$. Provide the ranges of each parameter and select an initial model $\beta_0=(\beta_{01},\beta_{02},\cdots,\beta_{0p})$ randomly within this range. The corresponding objective function value $E(\beta_0)$ will be calculated.

(2) Disturb the current model parameters to generate a new model. Let $\beta=(\beta_{01}+\tau_1,\beta_{02}+\tau_2,\cdots,\beta_{0p}+\tau_p)$. Similarly, the corresponding objective function value $E(\beta)$ will be calculated, such will get $\Delta E=E(\beta)-E(\beta_0)$.

(3) If $\Delta E<0$, the new model can be accepted; if $\Delta E>0$, the new model $\beta$ will be accepted based on probability $P=\exp(-\Delta E/T)$ ,where $T$ is the temperature. In case of the model being accepted, set $\beta_0=\beta, E(\beta_0)=E(\beta)$.

(4) At the temperature $T$, reiterate the disturbance and acceptance processes for certain times, i.e. , repeat the steps of (2) and (3).

(5) Decrease the temperature $T$ slowly.

(6) Repeat steps (2) and (5) until $T<T_f$ or $E(\beta)<\varepsilon$, where the $\varepsilon$ is the predetermined smaller positive number.

(7) The final solution obtained is the most optimal solution, i.e., the parameter values we are seeking for.

#### *2.2.2. The Improvement of Simulated Annealing Algorithm*

During the application of the algorithm, to get the parameters faster and more accurately following improvement could be made:

(1) Model disturbance

The author analyzes disturbance improvement strategy for simulated annealing algorithm and puts forward an algorithm which intensifies the local search ability. The algorithm drawlessonsfroms the idea of non-uniform mutation in genetic algorithm and generates new model parameters after disturbing the current model parameters with non-uniform mutation strategy, namely:

$$\beta_i' = \beta_i + \varphi_i(\beta_{i,\max} - \beta_{i,\min})$$
$$\text{or} \quad \tau_i = \varphi_i(\beta_{i,\max} - \beta_{i,\min}),$$
$$\varphi_i = r(1 - t/N)^{\lambda} \operatorname{sgn}(r - 0.5).$$

where $\beta_i$ is No. $i$ parameter in current model, $\beta_i'$ is No. $i$ parameter after disturbance, $\varphi_i$ is No. $i$ parameter disturbance factor, $\beta_{i,\min}, \beta_{i,\max}$ are values range of No. $i$ parameter, $r$ is a random number within $(0,1)$, $t$ is current temperature, $N$ is the maximum iteration(related to maximum temperature and minimum temperature), $\lambda$ is the constant to confirm non-uniformity and sgn is a sign function.

The above equation has showed that the search range is relatively broad in case of high temperature; however, the search range narrows gradually with the decrease of the temperature. The value of $\varphi_i$ decreases gradually with the increase of iterations (i.e. the decrease of the temperature) thus the search range also narrows, which intensifies the local search ability. In the process of actual computation, the value of $\lambda$ must be controlled reasonably—it is prone to be fallen into local extremum in case the value is too big. Generally speaking it is better to be located within $[2,5]$. The new accepted model will get closer to the actual model to be solved with the decrease of the temperature. Therefore, intensifying local search is more favorable to improve the convergence performance of the algorithm.

(2) Cooling method

As for annealing plan, the practice shows that the changing pattern of $T$ exponentially is more suitable to the nature of annealing. In consideration of the fact that temperature drops very fast at the beginning then slow down at last in annealing method, the author adopts the modified index curve which is more suitable to the decreasing characteristic, namely

$$T(k) = \xi T_f + T_0 \alpha^k$$

where $T_0$ is the initial temperature, $\xi$ is the given constant and $0 < \xi < 1$, $\alpha$ is the speed factor, $k$ is the iterations. With the increase of $k$, the temperature drops very fast at the beginning then slow down at last. Such the large-scaled rough search is utilized at the beginning and then refined search is utilized locally at last.

# 3. Example Application

At present, China economic is rapid growth. Researching on developing trend of household per capita consumption expenditure of urban and the rural residents in China is of vital significance. What the increasing regularity of household per capita consumption expenditure of urban and the rural residents in China and what kind of development tendency will be presented in the future? The article chooses Gaussian model based on the characteristics of growth for household per capita consumption expenditure of urban and the rural residents in China during 1991-2010. The information refers to Table 1.

The form of model:

$$g(x,\beta) = \hat{y} = a_1 e^{-(\frac{x-b_1}{c_1})^2} + a_2 e^{-(\frac{x-b_2}{c_2})^2}$$

***Table 1.*** *Information about household per capita consumption expenditure of urban and the rural residents in China and relevant outcome.*

| Time (year) | $x$ | $y$ (Yuan) | Predicted value | Relative error (%) | Time (year) | $x$ | $y$ (Yuan) | Predicted value | Relative error (%) |
|---|---|---|---|---|---|---|---|---|---|
| 1991 | 1 | 1453.81 | 1409.893 | 3.0208 | 2001 | 11 | 5309.01 | 5356.252 | -0.8899 |
| 1992 | 2 | 1671.73 | 1806.887 | -8.0849 | 2002 | 12 | 6029.88 | 5842.035 | 3.1152 |
| 1993 | 3 | 2110.81 | 2295.847 | -8.7662 | 2003 | 13 | 6510.94 | 6461.218 | 0.7637 |
| 1994 | 4 | 2851.34 | 2843.249 | 0.2838 | 2004 | 14 | 7182.1 | 7201.305 | -0.2674 |
| 1995 | 5 | 3537.57 | 3388.491 | 4.2142 | 2005 | 15 | 7942.9 | 8045.821 | -1.2958 |
| 1996 | 6 | 3919.47 | 3865.783 | 1.3698 | 2006 | 16 | 8696.55 | 8981.852 | -3.2806 |
| 1997 | 7 | 4185.64 | 4235.898 | -1.2007 | 2007 | 17 | 9997.47 | 10001.8 | -0.0433 |
| 1998 | 8 | 4331.61 | 4508.229 | -4.0774 | 2008 | 18 | 11242.85 | 11102.01 | 1.2527 |
| 1999 | 9 | 4615.91 | 4737.806 | -2.6408 | 2009 | 19 | 12264.55 | 12280.74 | -0.1320 |
| 2000 | 10 | 4998.0 | 4999.384 | -0.0277 | 2010 | 20 | 13471.45 | 13536.67 | -0.4842 |

By utilizing the method in the article, the trigonometric function neural network model will take performance index $J = 10^{-6}$. The number of hidden layers of neural network $m+1 = 11$, the learning rate $\mu = \frac{1.2}{m+1} = 0.1091$, the sample training rate is $\{x_k \mid e(x_k), k = 0,1,\cdots,20\}$.

The parameter setting of simulated annealing algorithm is: initial temperature $T_0 = 100$, minimum temperature $T_f = 0.001$, the length of Markov chains is 10, temperature drop factor $\alpha = 0.90$,the non-uniformity degree constant of model disturbance $\lambda = 2.5$, $[\beta_{i,\min}, \beta_{i,\max}] = [10^{-1}, 10^{6}]$ and the $\xi = 0.5$ in cooling method.

By software programming with Matlab, we get

$$\beta = (\beta_1, \beta_2, \cdots, \beta_p) = (a_1, b_1, c_1, a_2, b_2, c_2)$$
$$= (48763.1424, 46.8122, 23.6843, 1374.7020, 6.3931, 4.1466)$$

namely

$$g(x, \beta) = \hat{y} = 48763.1424 e^{-(\frac{x-46.8122}{23.6843})^2} + 1374.7020 e^{-(\frac{x-6.3931}{4.1466})^2}$$

# 4. Summary

The simulated annealing and neural network algorithm has been applied in data fitting least square integral method. The numerical simulation results show that the method is feasible and valid. The simulated results of simulated annealing neural network algorithm are desirable by judging from the test amount . However, the major defect of the algorithm is the contradiction between the precision of solution and long solving time. The solving time will increase greatly with the increase of inverting parameters. In addition, the scope of variation interval of each parameter has relatively great influence on the convergence rate of algorithm. Consequently, how to confirm effectively the ranges of parameters and some initial parameters of the algorithm so as to increase the convergence rate of algorithm is still to be further studied and discussed.

# References


[1] Zhang X. D, J. N. Yu, L. P. Guo, J. G. Zhang, Q. H. Ding (2011). Prediction of modeling of nonlinear times Series based on Improved BP neural network. *Journal of Mianyang Normal University*, 30: 84-88.

[2] Gao X. J. (2011). Application of BP neural network in prediction of the liver cirrhosis's cure. *Mathematical Theory and Applications,* 31: 20-23.

[3] Gavin J. J. B. Bowden, G. C. Nixon, H. R. Dandy, M. H. Maier (2006). Forecasting chlorine residuals in a water distribution system using a general regression neural network. *Mathematical and Computer Modelling,* 44: 69-484.

[4] Hu J. X, J. Zeng (2010). A Fast learning algorithm of global convergence for BP-neural network. *Journal of Systems Science and Mathematical Sciences*, 30: 604-610.

[5] Jia H. P. (2012). GRNN neural network in the application of power system load forecasting. *Electronic Design Engineering*, 20: 14-16.

[6] Alrefaei M. H, A. H. Diabat (2009). A simulated annealing technique for multi-objective simulation optimization. *Applied Mathematics and Computation*, 215: 3029-3035.

[7] Anagnostopoulos K. P, L. Kotsikas (2010). Experimental evaluation of simulated annealing algorithms for the time-cost trade-off problem. *Applied Mathematics and Computation,* 217: 260-270.

[8] Chen J, Y. Liu (2011). Traffic prediction research of neural network combined simulated annealing algorithm. *Computer Engineering and Design,* 32: 2138-2145.

[9] Gao S. (2002). Research on annealing strategy in Simulated Annealing Algorithm. *Aeronautical Computer Technique*, 32: 20-25.

[10] Gu Y. X, B. W. Xiang, G. Z. Zhao (2005). An improved simulated annealing algorithm for global optimization problems with continuous variables. *Systems Engineering Theory & Practice,* 25: 103-177.

[11] Shi H., Y. L. Wang, F. L. Weng (2011). Composite prediction model of BP neural networks and fuzzy time series and its application. *Journal of Computer Application*, 31: 90-92.

[12] Xiao H., S. D. Liu, X. Y. Huang, L. Jin (2010). A study on neural network ensemble forecast model based on kernel principal component analysis. *Computer Simulation*, 27: 163-167.

[13] Qiu Q. R, T. Yu (2011). BP neural network forecast model based on principal component analysis for the real estate price of prediction. *Journal of Hunan University of Arts and Science (Natural Science Edition)*, 23: 24-27.

[14] Li H. X, C. W. Li (2009). Application of artificial neural network in predicting the number of graduate students. *Mathematics in Practice and Theory*, 39: 27-33.

[15] Feng Z. Y, M. H. Wang (2011). Essential analysis on the generalization problem of feedforward neural network used in economic forcasting. *Journal of Shaanxi University of Science & Technology (Natural Science Edition)*, 29: 108-111.

[16] Guo D. L, H. M. Xia, Y. Q. Zhou (2010). Hybrid simulation annealing and evolution strategy algorithm in non-linear parameter estimation application. *Mathematics in Practice and Theory*, 40: 91-98.

[17] Hao Z, F. Hong (2009). Using genetic/simulated annealing algorithm to solve disassembly sequence planning. *Journal of Systems Ensnaring and Electronics,* 20: 906-912.

[18] Jin L. X. H. W. Tang, B. Li, M. J. Ji, X. Z. Zhu (2005). A simulated annealing algorithm for continuous functions and its convergence properties. *Mathematica Numerica Sinica*, 27: 19-30.

[19] Liu J. J, M. F. Chen, Z. P. Ye (2010).The estimation of the non-linear model parameter based on combination of damped least-squares method and simulated annealing method. *Journal of Jinggangshan University (Natural Science),* 31: 10-14.

[20] Michael A. H, M. Brasel, J. Morig, F. Tusch, P. Werner, algorithms for minimizing mean flow time in an open shop. *Mathematical and Computer Modelling*, 48: 1279-1293.