Acta Technica *62* (2017), No. 7A, 1459–1470 © 2017 Institute of Thermomechanics CAS, v.v.i.

Optimization for petric network quality management process of lean production mode oriented at customized products

Fei Shang[1], Xiaobo Nie[1]

Abstract. In order to enhance improvement for optimized performance for quality management process of lean production mode oriented at customized products, method for quality management process optimization based on Petri network mode optimization was proposed. First of all, process optimization for quality management of lean production mode oriented at customized products was conducted with design of schedule optimization model under Cloud environment. Besides, time-sequence model for optimal system of quality management process was described and optimization constraint was simplified with Petri network model. Dimensionality reduction for binary input of system state space was realized with Finite Automaton (DFA) to reduce complexity of the model. Eventually, validity of algorithm for was verified with simulation experiment.

Key words. Customized products, Lean production, Quality management, Process optimization

.

1. Introduction

Process calculation for quality management of lean production mode oriented at customized product that serves as a new business service mode is a new generation of high-speed network computation and service platform. With constant increase of its system scale and calculation performance, schedule optimization with high efficiency is a subject attracting great attention. Overhead of production process has become the primary one overhead for lean production and operation of production enterprises. High energy consumption can cause not only high computation overhead

[1]Department of industrial engineering, mechanics institute, Inner Mongolia university of technology , Hohhot, Inner Mongolia , 010051, China

but also instability of system and pollution of environment. Improving computing performance only doesn't conform to requirement for computing development of quality management process obviously. It is one of effective methods to combine computing performance and schedule optimization for quality management process of high-effective lean production mode.

High-effective optimization schedule for computing system of large-scale heterogeneous quality management process is a complex optimization problem with multiple targets, dynamic constrain, and real-time preference characteristics. On the one hand, as a commercial computing platform, that realizing service profit maximization is one of the targets is needed. However, customer need for different computing tasks is met at the same time. For instance, some customers tend to computing performance (the faster, the better), while some prefer to computing overhead (the more energy-saving, the better). On the other hand, complex computing system of quality management process has characteristics in heterogeneous computational node, dynamic change of available resources, and preference of real-time change of management demand and so on. In traditional meaning, simple schedule strategy and predefined single target optimization method based on decision preference cannot meet the requirement of its complex computing demand of multiple targets, dynamic condition of resource constrain, and real-time change of preference change. Therefore, there is significant meaning for researching optimization schedule method with multiple targets under computing environment of quality management process.

In order to improve performance for process optimization for quality management of lean production mode oriented at customized products for the Paper, process optimization for quality management of lean production mode oriented at customized products was designed with schedule optimization model under Cloud environment and optimization constraint was simplified with Petri network model. Dimensionality reduction for binary input of system state space was realized with Finite Automaton (DFA) to reduce complexity of the model. Validity of algorithm for is verified with simulation experiment.

2. Schedule optimization model under Cloud environment

System structure of task schedule for computing system of quality management process is shown in Fig. 1. It includes user's task, middleware of task schedule, and heterogeneous computer cluster. Middle ware of task schedule is core module in it and it includes DAG generation of computing task, decision for optimization schedule plan, and computing ask allocation. One important target for middleware of task schedule is to satisfy Server Level Agreement (SLA) of customer as possible and improve Quality of Service (QoS) of computing system at the same time.

When schedule optimization computing of multiple targets is oriented at production process, two computing models shown in the Fig.2 can be used for middleware of task schedule. Traditional computing mode that weighted sum of predefined preference is used for converting multiple optimization targets to single target optimization is shown in Fig. 2. Among others, preference weigh of optimization target needs to be preset and general optimization result is single solution. Diversified real-time

dynamic preference decision cannot be realized with infinite decision space. However, another optimization process that can be used is shown in Fig. 2. First of all, scheduling problem is defined as optimal problem with multiple targets. After optimal computing is conducted with high-effective optimal algorithm, Pareto optimal schedule solution set of entire decision space (eclectic non-inferior solution set for all optimization target) is obtained. Then real-time decision preference is generated according to dynamic constraint of resource, demand of system stability, profit preference of service system, and system management preference of computing system of current quality management process for Paret. Eclectic solution serves as final schedule scheme. Compared with the first method, this computing mode is more flexible in choosing space of decision and it fits the dynamic characteristic of complex large-scale computing system of quality management process. However, it has higher requirement for performance of optimization algorithm with multiple targets and it is core part. Effort is spared to analyze optimization algorithm with multiple targets for production process and performance under Cloud environment in Fig.2.

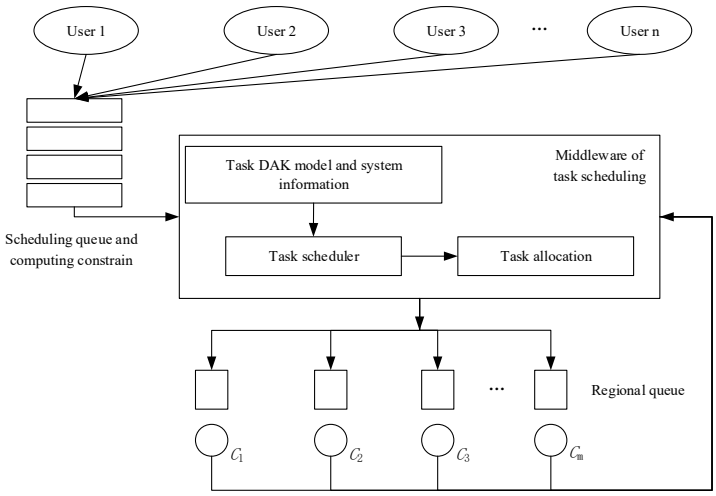

Fig. 1. System structure for task schedule under cloud environment

3. Solution to Petri network based on infinite automaton

3.1. Time sequence of OTC system- conflicting model of Petri net

Petri network is a directed weighed bipartite graph consisting of two kinds of nodes, including place, transition, connection, and token and it is shown in the Fig.3.

In above time sequence model, $P = \{p_1, \cdots, p_5\}$ indicates place, $t = \{t_1, t_2, t_3\}$ indicates transition, and black point in the Fig. 3 indicates token and they are in corresponding problem area, strategic pass, and produced product respectively.

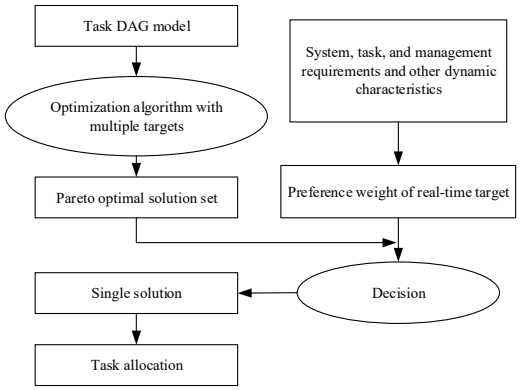

Fig. 2. Decision process of schedule middleware

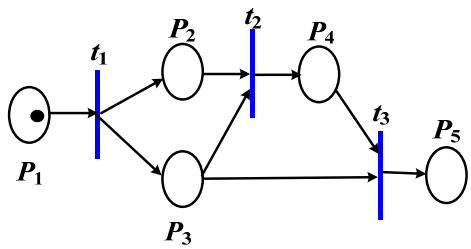

Fig. 3. Petri network model

Especially, it is always regarded that weight of each link arc belongs to {0, 1} and there are not multiple sides. Under this situation, relation of place-transition is presented by the occurrence matrix and it allocates a non-negative integer (weight) for each input arc (from transition to place) and output arc (from place to transition). And incidence matrix of input arc and output arch of each car can be presented as: $\hat{B}^+ \in \{0,1\}^{n\times m}$ and $\hat{B}^- \in \{0,1\}^{n\times m}$, and it satisfies:

$$\hat{B} = \hat{B}^+ - \hat{B}^- . \tag{1}$$

Petri network model of undirected OTC system can be constructed based on state space model. Transition rule of Petri network can be defined with constraint, and G_2 can be defined as:

$$G_2 = diag\left(\hat{B}^-, \cdots, \hat{B}^-\right) \in \boldsymbol{R}^{mS\times mS} . \tag{2}$$

Inclusion relation among constraints is focused here for reducing constraint amount and the following lemma can be obtained:

Theorem 1: considering $\hat{x}^{v_s} \in \mathcal{M}$, $\mathcal{M} = \left\{\eta \in \{0,1\}^n \left| \mathbf{1}^T \cdot \eta = 1\right.\right\}$, $s = 1, \cdots, S$, $x^{\sum v_s}(0) \in \{0,1\}^n$, in case the former two formulas in formula (1) satisfy it, the third formula is also true.

Prove: according to formula (4)- formula (10), it can be obtained that the fol-

lowing formula is true for $s = 1, \cdots, S$, $0 \le k \le M-1$:

$$\hat{x}^{v_s}(k+1) = \hat{A}\hat{x}^{v_s}(k) + (\hat{B}^{+} - \hat{B}^{-})\hat{u}^{v_s}(k)\,. \tag{3}$$

Besides, it can be obtained from the former two formulas in formula (1):

$$0 \le \hat{x}^{v_s}(k) - \hat{B}^{-}\hat{u}^{v_s}(k) \le \hat{A}\hat{x}^{v_s}(k) + (\hat{B}^{+} - \hat{B}^{-})\hat{u}^{v_s}(k)\,. \tag{4}$$

As for $s = 1, \cdots, S$, $0 \le k \le M-1$, it can be obtained from formula (14) and formula (11):

$$\begin{aligned} \mathbf{0} &\le A\hat{x}^{v_s}(k) + (\hat{B}^{+} - \hat{B}^{-})\hat{u}^{v_s}(k) \\ &= \hat{A}_a\hat{x}^{v}(k) + \hat{B}_a\hat{u}^{v}(k) = \hat{x}^{v_s}(k+1) \le \mathbf{1} \end{aligned} \tag{5}$$

It can be obtained according to (3-5)

$$\begin{aligned} &0 \le \hat{x}^{v_s}(k) - \hat{B}^{-}\hat{u}^{v_s}(k) \le \hat{A}\hat{x}^{v_s}(k) + (\hat{B}^{+} - \hat{B}^{-})\hat{u}^{v_s}(k) \\ &\le \sum_{s=1}^{S}\{\hat{A}\hat{x}^{v_s}(k) + \hat{B}^{v_s}\hat{u}(k)\} \end{aligned} \tag{6}$$

As for $s = 1, \cdots, S$, $0 \le k \le M-1$, it can be obtained according to formula (1) so as to conclude that:

$$\sum_{s=1}^{S}\hat{u}_{c1}^{v_s}(k) + \hat{u}_{c2}^{v_s}(k) \le 1, \sum_{s=1}^{S}u_{d1}^{v_s}(k) + u_{d2}^{v_s}(k) \le 1\,. \tag{7}$$

As for $0 \le k \le M-1$, $\hat{u}_{c1}^{v_s}$ and $\hat{u}_{c2}^{v_s}$ are strategic pass of convergence region, $\hat{x}_C^{v_s}$ and $\hat{u}_C^{v_s}$ are strategic pass of diverging area. It can be obtained from formula (4):

$$\hat{x}_C^{v_s}(k+1) = \hat{x}_C^{v_s}(k) + \hat{u}_{c1}^{v_s}(k) + \hat{u}_{c2}^{v_s}(k) - \hat{u}_C^{v_s}(k)\,. \tag{8}$$

As for $0 \le k \le M-1$, it can be obtained according to formula (8) and the third formula of formula (1):

$$\sum_{s=1}^{S}x_C^{v_s}(k+1) \Leftrightarrow \sum_{s=1}^{S}\{x_C^{v_s}(k) + \hat{u}_{c1}^{v_s}(k) + \hat{u}_{c2}^{v_s}(k) - \hat{u}_C^{v_s}(k)\} \le 1\,. \tag{9}$$

Therefore, it can be obtained in combination with formula (4) and formula (1):

$$\sum_{s=1}^{S}\{\hat{u}_{d1}^{v_s}(k) + \hat{u}_{d2}^{v_s}(k)\} \le \sum_{s=1}^{S}x_d^{v_s}(k)\,. \tag{10}$$

Therefore, lemma 1 is proved to be true. Necessary initial condition of lemma 1 is $\hat{x}^{v_s}(0)$ and $x^{\sum v_s}(0)$, because the former two formulas of formula (10) must be true in initial time.

3.2. Finite automaton method

Node in finite automaton indicates active discrete dynamics model in discrete time.

Assumption 1: finite automaton provides linked directed graph where two ends of each arc connect to some nodes and there is at least one input arc and one output arc for each node. In case the assumption is true, it can be marked with A.

According to relation for input arc and output arc of finite automaton, implicit structure form of system is given as shown in Fig. 4 [14]:

$$E\xi(k+1) = F\xi(k), \xi(k) \in \{0,1\}^{n'+m'}, \xi(0) \in \Xi_0(\delta_0) \tag{11}$$

Where, $\Xi_0(\delta_0) = (\eta \in \{0,1\}^{n'+m'} | 1^T\} \cdot \eta = 1$, $E, F \in \{0,1\}^{n' \times (n'+m')}$, $\mathrm{E}\eta = \delta_0 \in \mathcal{M}_D$, $\mathcal{M}_D = \{\eta \in (0,1)^{\mathrm{n'}} | 1^{\mathrm{T}} \cdot \eta = 1\}$. And the following theorem can be obtained:

Theorem 2: With designated $\delta_0 \in \mathcal{M}_D$, considering implicit system (11), it satisfies assumption $\mathcal{A}$ and permutation matrix $P \in (0,1)^{(n'+m') \times (n'+m')}$ satisfies $EP = \left[I_{n'} \tilde{E} \right]$, where $\tilde{E} \in \{0,1\}^{n' \times m'}$. Then, under transformation of coordinates:

$$\begin{bmatrix} \hat{x}_D \\ \hat{u}_D \end{bmatrix} := \begin{bmatrix} I_{n'} & \tilde{E} \\ O_{m' \times n'} & I_{m'} \end{bmatrix} P^{-1} \xi \tag{12}$$

Implicit system (12) is equal to the following state formula:

$$\begin{cases} \hat{x}_D(k+1) = \hat{A}_D \hat{x}_D(k) + \hat{B}_D \hat{u}_D(k) \\ \hat{C}_D \hat{x}_D(k) + \hat{D}_D \hat{u}_D(k) \le \hat{G}_D \\ \hat{x}_D(k) \in R^{n'}, \hat{u}_D(k) \in \{0,1\}^{m'} \\ \hat{x}_D(0) = \Gamma_D \delta_0, \hat{u}_D(0) \in \{0,1\}^{m'} \end{cases} \tag{13}$$

Where $\hat{A}_D := \tilde{F}_a$, $\hat{B}_D := -\tilde{F}_a \tilde{E} + \tilde{F}_b$, $\hat{C}_D := -I_{m'}$, $\hat{D}_D := -\tilde{E}$, $\hat{G}_D := \mathbf{0}_{\mathrm{n'}}$, $\Gamma_D := \mathbf{I}_{m'}$, $[\tilde{F}_a \tilde{F}_b] := FP$.

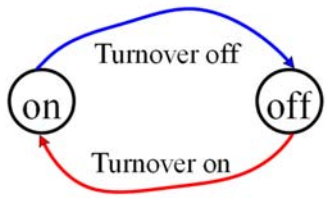

Fig. 4. Finite automaton

Considering problems of path and route, schedule module of product produced with Petri network based on formula (1) can be presented as:

$$\begin{cases} \hat{v}^{v_s}(k+1) = \hat{A}\hat{x}^{v_s}(k) + B\hat{u}^{v_s}(k) \\ \hat{C}\hat{x}^{v_s}(k) + D\hat{u}^{v_s}(k) \le G \\ \hat{x}^{v_s}(k) \in R^n, \hat{u}^{v_s}(k) \in \{0,1\}^m \\ \hat{x}^{v_s}(0) = \hat{x}_0^{v_s}, \hat{u}^{v_s}(0) \in \{0,1\}^m \end{cases} \tag{14}$$

Where, $\hat{A} := I_m$, $\hat{B} := -\hat{B}^- + \hat{B}^+$, $\hat{C} := -I_m$, $\hat{D} := \hat{B}^-$, $G := 0_n$, $\hat{B}^- \in \{0,1\}^{n\times m}$, $\hat{x}_0^{v_s} \in \mathcal{M}$, $\mathcal{M} = \{\eta \in \{0,1\}^n\}$. Considering discrete dynamics mode for location of produced product, path layout graph of assumption 1 shown in Fig. 1 is satisfied and system is realized in accordance with equation (13).

3.3. Schedule algorithm model

Cost function of OTC routing problem is mainly researched in the Section. $\delta_F(k) \in (0,1)$ indicates mark of transit time. $\delta_F(k) = 1$ indicates that all tasks are completed. Then all produced product returns to parking lot at the time of k. Otherwise, $\delta_F(k) = 0$ is true. Besides, designated region of produced product s indicates L_s. At the same time, $x_L^{v_s}(k) = 1$ indicates produced product s occurs in designated region at the time of k. Otherwise, . As for , there is vector that satisfies:

$$\begin{cases} \hat{x}_L^{v_s}(k) = g_L^{v_s T}\hat{x}^{v_s}(k) \\ g_L^{v_s} := [g_{Ll_1}^{v_s T} g_{Ll_2}^{v_s T} \cdots g_{Ll_{22}}^{v_s T}]^T \in R^n \end{cases} \tag{15}$$

$$g_{Ll_ij}^{v_s} := \begin{cases} 0, \ l_ij \ne L_p \\ 1, \ l_ij = L_p \end{cases} \tag{16}$$

Where, $g_{Ll_ij}^{v_s}$ is the j element of $g_{Ll_i}^{v_s}$. The following rule is considered here:

$$\sum_{k=0}^{M} \delta_F(k) = 1\,. \tag{17}$$

$$\sum_{s=1}^{S} \hat{x}_L^{v_s}(k) \ge S, 0 \le k \le M\,. \tag{18}$$

Formula (7) indicates that transit time is less than or equal to forecast level M. According to formula (17-18), it can be presented as the following inequation:

$$\begin{cases} -g_3^T\hat{x}^v(k) + S\sum_{t=0}^{k}\delta_F(t) \le 0 \\ g_3^T\hat{x}^v(k) - \varepsilon\sum_{t=0}^{k}\delta_F(t) \le S - \varepsilon \end{cases} \tag{19}$$

$$0 \le k \le M, g_3^T := [g_L^{v_I T} \cdots g_L^{v_s T}]^T\,. \tag{20}$$

Where, ε is a small enough normal number. According to formula (17) and formula (19), the following target function of minimum transit time can be obtained:

$$\min J \ s.t \ J := \sum_{k=0}^{M} k\delta_F(k)\,. \tag{21}$$

$\hat{x}_M^v(0) \in R^{nS(M+1)}, \hat{u}_M^v(0) \in R^{mSM}$ and $\delta_{FM}(0) \in R^{(M+1)}$ can be defined as respectively:

$$\begin{cases} \hat{x}_M^v(0) := [\hat{x}^v(0)^T, \hat{x}^v(1)^T, \cdots, \hat{x}^v(M)^T]^T \\ \hat{u}_M^v(0) := [\hat{u}^v(0)^T, \cdots, \hat{u}^v(M-1)^T]^T \\ \delta_{FM}(0) := [\delta_{FM}(0)^T, \delta_{FM}(1)^T, \cdots, \delta_{FM}(M)^T]^T \end{cases} \tag{22}$$

Under this situation, target function can be re-written as:

$$\min_{\delta_{FM}(0)} J \ s.t \ J := G_O\delta_{FM}(0)\,. \tag{23}$$

Where, $\hat{x}_M^v(0)$can be defined as:

$$\hat{x}_M^v(0) = \hat{A}_M\hat{x}^v(0) + \hat{B}_M\hat{u}_M^v(0)\,. \tag{24}$$

Where, matrix G_O can be defined as:

$$G_O := [0, 1, \cdots, M]^T \in R^{(M+1)}\,. \tag{25}$$

$$\hat{A}_M = \left[1, \hat{A}, \hat{A}^2, \cdots, \hat{A}^M\right]^T\,. \tag{26}$$

$$\hat{B}_M := \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \hat{B} & 0 & \cdots & 0 \\ \hat{A}\hat{B} & \hat{B} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ \hat{A}^{M-1}\hat{B} & \cdots & \hat{A}\hat{B} & \hat{B} \end{bmatrix}. \tag{27}$$

Constrains (1), (17), (19) can be presented as:

$$C_M\hat{x}_M^v(0) + D_M\hat{u}_M^v(0) + E_M\delta_{FM}(0) \le FM\,. \tag{28}$$

4. Experiment analysis

Simulation experiment is used in the paper to verify and evaluate performance of proposed algorithm (MOMA). During experiment process, extensible heterogeneous simulation environment ① consisting of Intel AMD and other 31 treatment institutes were established. Several kinds were selected from 32 kinds of computing material randomly to be computing material for each experiment. The main purpose of the

experiment is to verify convergence performance of multiple targets and computation complexity for the algorithm in the Paper. Because optimization algorithm with multiple targets for optimization of performance and production process oriented at Cloud environment is rare, the classic multiple targets algorithm MOEA was improved in the research experiment. Then it was compared with the algorithm in the Paper.

S index and C index are used for evaluation index of convergence performance of algorithm. Convergence performance of optimization solution set χ' is measured mainly based on $S(\chi')$ for S index. $S(\chi')$ indicates volume of hypercube for corresponding target space of χ'. Because number for optimization solution obtained for each algorithm is different, average $S(\chi')$ is used in the Paper and it indicates quantity of optimal solution obtained with $S(\chi')$ to divide the algorithm. Convergence index C can be defined as:

$$C(\chi'', \chi') := \frac{|\{\alpha'' \in \chi''; \exists \alpha' \in \chi' : \alpha' \geq \alpha''\}|}{|\chi''|}. \tag{29}$$

Where, $\chi', \chi'' \in \chi$ are two solution sets of decision variable. Value for C is selected from 0 to 1. When $C(\chi'', \chi') = 1$, it means that solution in χ'' is dominated by or equal to χ'. When $C(\chi'', \chi') = 0$, it means all solutions in χ'' cannot be dominated by or equal to χ'. In contrast experiment, $C(\chi'', \chi')$ and $C(\chi', \chi'')$ were taken into consideration in the Paper. Experiment results are shown in Fig. 5 to Fig. 7. Data for confidence interval of average value and average value of 95% in 50 times of experiments are shown in the figures.

Experiment result of schedule simulation for DAG task graph with different scale is shown in Fig.5. It can be found from that figures that average value of C (MOMA, MOEA) is always more than that of C (MOMA, MOEA) and confidence intervals of C (MOMA, MOEA) are all more than that of C (MOMA, MOEA). It indicates that superior optimal solution is generated with MOMA algorithm under different task quantity (dimensionality of corresponding problems) and the performance is more stable. Corresponding different task quantity (32, 64, 128, 256), solution in MOEA optimal solution set is dominated by MOEA. Combining both of them, it can be known that MOMA algorithm has superior convergence performance compared with MOEA algorithm.

Experiment result of schedule simulation for different quantity of processor with certain task number is shown in Fig. 6. As for corresponding different processor quantity (4, 8, 16, 32), average rates for solution in MOEA optimal solution set dominated by solution of MOMA are of 72%, 75%, 71%, and 40% respectively. Average rates for solution in MOMA dominated by solution in MOEA are of 7%, 12%, 11%, 34% respectively. Average index S of MOMA algorithm is less than that of MOEA. Therefore, it can be known that MOMA algorithm with different quantity of processor has superior convergence performance compared with MOEA algorithm.

In schedule simulation experiment of DAG task graph with different parallel factors, the experiment result is shown in Fig.7. As for corresponding parallel factor with different DAG task graph (0.5, 1, 2, 5), average rates for solution in MOEA

optimal solution set dominated by solution in MOMA are of 71%, 75%, 74%, and 74%, while average rates for solution in MOMA optimal solution set dominated by solution in MOEA are of 11%, 12%, 10%, and 12%. Average index S of MOMA is less than that of MOEA.It can be found that influence of different parallel factors of DAG task graph on confidence performance is less and MOMA algorithm has superior convergence performance compared with MOEA algorithm.

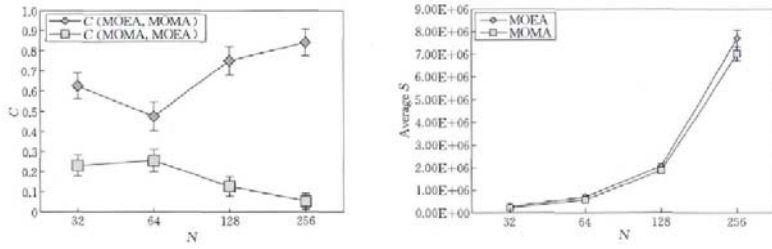

Fig. 5. C index and S index under different task quantity

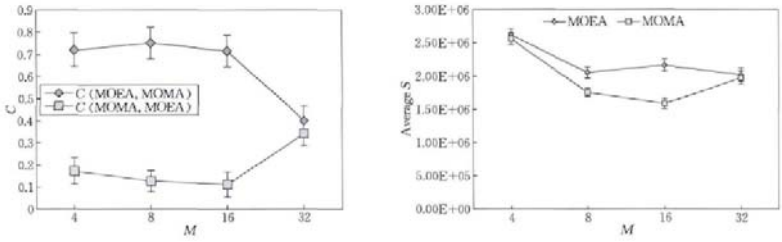

Fig. 6. C index and S index under different quantity of processor

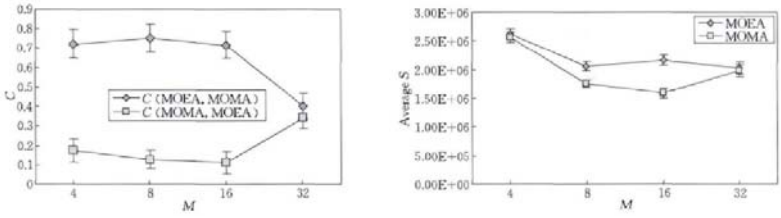

Fig. 7. C index and S index under different parallel coefficient

5. Conclusion

The Paper is to enhance the improvement for performance of process optimization for quality management of lean production mode oriented at customized products and design schedule optimization model under Cloud environment for process optimization for quality management of lean production mode oriented at customized products. Optimization constraint was simplified with Petri network model and

dimensionality reduction for binary input of system state space was realized with Finite Automaton (DFA) to reduce complexity of the model so as to verify the validity of the algorithm.

Scheduling optimization algorithm with multiple tasks of lean production system oriented at complex heterogeneous production enterprises was researched as key point in the Paper. Available computing method was proposed to construct flexible optimization scheduler for more practical optimization schedule with multiple targets. Targeted at dynamic characteristics and management requirement of specific distributed production system of production enterprises, using suitable real-time preference decision is of the same importance. This will be important work of application and research for the method.

References

[1] Y. J. Zhao, L. Wang, H. J. Wang, and C. J. Liu: *Minimum Rate Sampling and Spectrum Blind Reconstruction in Random Equivalent Sampling.* Circuits Systems and Signal Processing *34* (2015), No. 8, 2667–2680.

[2] S. L. Fernandes, V. P. Gurupur, N. R. Sunder, N. Arunkumar, S. Kadry: *A novel nonintrusive decision support approach for heart rate measurement.* Pattern Recognition Letters. https: //doi.org/10.1016/j.patrec.2017.07.002 (2017).

[3] N. Arunkumar, K. Ramkumar, V. Venkatraman, E. Abdulhay, S. L. Fernandes, S. Kadry, S. Segal: *Classification of focal and nonfocal EEG using entropies.* Pattern Recognition Letters *94* (2017), 112–117.

[4] J. W. Chan, Y. Y. Zhang, and K. E. Uhrich: *Amphiphilic Macromolecule Self-Assembled Monolayers Suppress Smooth Muscle Cell Proliferation*, Bioconjugate Chemistry *26* (2015) No. 7, 1359–1369.

[5] M. P. Malarkodi, N. Arunkumar, N. V. Venkataraman: *Gabor wavelet based approach for face recognition.* International Journal of Applied Engineering Research *8* (2013), No. 15, 1831–1840.

[6] L. R. Stephygraph, N. Arunkumar: *Brain-actuated wireless mobile robot control through an adaptive human-machine interface.* Advances in Intelligent Systems and Computing *397* (2016), 537–549.

[7] W. Pan, S. Z. Chen, Z. Y. Feng: *Automatic Clustering of Social Tag using Community Detection.* Applied Mathematics & Information Sciences *7* (2013), No. 2, 675–681.

[8] Y. Y. Zhang, E. Mintzer, and K. E. Uhrich: *Synthesis and Characterization of PEGylated Bolaamphiphiles with Enhanced Retention in Liposomes*, Journal of Colloid and Interface Science *482* (2016), 19–26.

[9] X. Du, S. Zhen, Z. Peng, C. Zhao ,Y. Zhang, W. Zhe, X. Li, G. Liu, X. Li: *Acetoacetate induces hepatocytes apoptosis by the ROS-mediated MAPKs pathway in ketotic cows.* Journal of Cellular Physiology (2017) No. 232, 3296–3308.

[10] D. S. Abdelhamid, Y. Y. Zhang, D. R. Lewis, P. V. Moghe, W. J. Welsh, and K. E. Uhrich: *Tartaric Acid-based Amphiphilic Macromolecules with Ether Linkages Exhibit Enhanced Repression of Oxidized Low Density Lipoprotein Uptake*, Biomaterials *53* (2015), 32–39.

Received May 7, 2017