Acta Technica *62* (2017), No. 7A, 1471–1482

Improvement of workshop production flow in the assembly line project based on the lean production pattern

Huikai Zhao[1]

Abstract. An optimization method for the assembly line project scheduling based on random graph Markovian decision process (MDP) of the multi-objective fruit fly algorithm is proposed to improve the rationality of optimization process in the assembly line project under the lean production pattern. First, a new method is introduced to cope with conflict, once there is conflict in the scheduling scheme, instead of redefining the start time of the assembly line project, it is to transfer the state of the time schedule of conflict to obtain another feasible time schedule and improve the computational efficiency of algorithm. Second, the two-order multi-objective fruit fly algorithm is designed; at the first stage, two knowledge bases are adopted to carry out the initialization of fruit fly group location, flying direction guidance and reference point storage; in addition, the fruit fly de-structure shall be reconstructed according to sense of smell. At the second stage, the weighted equilibrium method is adopted to transform the optimization of intensive search process to maintain the diversity of fruit fly group and reference point, improve the searching capacity. Finally, the effectiveness of the mentioned method in convergence accuracy and computational efficiency is verified through the simulation experiment and the rationality of scheduling scheme is demonstrated through the Gantt chart method.

Key words. Lean production, Assembly line, Production flow, Multi-objective optimization, Fruit fly algorithm.

1. Introduction

In previous research, the fixed resource capacity and determination of project duration are the major issues to research the assembly line project scheduling. However, in the actual environment, it is unpractical to obtain the deterministic information only. It's difficult to relay on these information in case of the internal and/or external emergency even if we have managed to obtain the deterministic information in some way. Therefore, uncertainty has become an inevitable issue in the assembly line

[1]Xi'an Aeronautical University, Shanxi Xi'an, 710003, China

project scheduling over the recent decades. In the assembly line project scheduling, it is more important to handle the uncertainty than other fields for the assembly line project is a huge and expensive one-time action and many assembly line projects will be postponed every year. The treatment of issues on uncertain assembly line projects have explored and discussed by many authors over the past decades.

Questions are on two aspects of the above documents in research: (1) These authors have ignored the influences of selecting the reactive scheduling strategies on the optimality of baseline scheduling, vice versa. This process has not only provided the heuristic solutions based on simulation, but also selected the passive strategies from the early start-up strategies, which has greatly restricted the flexibility of process. (2) These authors only assume that the executive response of these assembly line projects exerts no influence on the formulation of benchmark scheduling and scheduling strategies. In fact, this assumption is incorrect and has sparked the necessity of baseline scheduling. In addition, we must consider that the conflict response of the assembly line project execution will not only need some cost but also damage the commercial reputation of the contractors.

An optimization method for the assembly line project scheduling based on random graph Markovian decision process (MDP) of the multi-objective fruit fly algorithm is proposed in this paper; the scheduling conflict process of project is improved, two stages of multi-objective fruit fly algorithm are designed, the high-accuracy optimization of project scheduling is realized and the effectiveness of algorithm is verified by the experiment result.

2. Problem model description

2.1. Problem definition

First of all, the assembly line project scheduling problems restricted by active and passive resources are introduced in this chapter. The parameter definitions required by the example expression of each problem are provided as below:

The virtual assembly line project set $N = \{0, 1, \cdots, n+1\}$ is given, in which 0 and $n+1$ respectively refer to the virtual start and virtual end. Each assembly line project $i \in N \backslash \{0, n+1\}$ is of non-zero independent random duration of integers and $p_i^{\min} \leq \tilde{p}_i \leq p_i^{\max}$, it is subject to the discrete distribution. It is noticed that the assume the implementation time of assembly line project is known only after the assembly line project is executed. Vector $\Phi_{\Pi,l}$ can be expressed by the finite support set $p = \left(p^1, p^2, \cdots, p^{|\beta|}\right)$, in which element p^l can be expressed as $p^l = \left(p_0^l, p_1^l, \cdots, p_{n+1}^l\right) \in \beta$ within the time of duration. The occurrence probability of assembly line project executing p^l is $\pi\left(\tilde{p} = p^l\right)$. β can be a huge set and our solution is to directly adopt the information provided by the given discrete distribution.

The resource-constrained set R, during the processing time, the resource type of r_{ik} unit required by each assembly line project i is $k \in R$. The resource availability of resource type k can be expressed as R_k.

The precedence constraint set of assembly line project $E = \{(i, j) \,|i, j \in N\}$, the

data sequence pair $(i, j) \in E$ instructing the assembly line project j cannot start up before the assembly line project i is finished.

2.2. Execution chain expression of assembly line project of the solution set

As mentioned earlier, the active and passive scheduling are generally taken as two independent issues in the previous studies; therefore, solutions to the two issues are different. Accordingly, a solution strategy (PR-strategy) integrating the active and passive resource restraints is proposed here.

In the scheduling of PR-strategy, for each assembly line project p^l, if the assembly line project is executed, there will generate a response sequence of $v_{\Pi,i}$, this response sequence can be expressed as $\Phi_{\Pi,l}$:

$$\Phi_{\Pi,l} : s^{[0]_{\Pi,l}} \overset{t=t_1}{\rightarrow} s^{[1]_{\Pi,l}} \overset{t=t_2}{\rightarrow} \cdots \overset{t=t_{v_{\Pi,i}}}{\rightarrow} s^{[v_{\Pi,i}]_{\Pi,l}} \tag{1}$$

It is only after the assembly line project is executed that the scheduling of PR-strategy is aware of the occurrence and its relevant chain reaction of the assembly line project, which requires that the benchmark scheduling of all chains must be the same.

If there is a dead-end node during the execution chain of the assembly line project of the solution set, the execution chain of the assembly line project is called the dead chain; parameter $\gamma_{\Pi,l}$ is introduced here to distinguish the dead chain situations. If $\Phi_{\Pi,l}$ is the dead chain executed by the assembly line project, $\gamma_{\Pi,l} = 1$, otherwise $\gamma_{\Pi,l} = 0$. The comprehensive cost $f(\Pi, l)$ of scheduling execution sequence chain $\Phi_{\Pi,l}$ of each assembly line project in the PR-strategy II is calculated in the form of:

$$\begin{aligned} f(\Pi, l) &= \omega_b \times s_{n+1}^{[0]_{\Pi,l}} + \gamma_{\Pi,l} M + \\ &\sum_{k=1}^{v_{\Pi,i}} \left(\sum_{i \in U\left(s^{[k-1]_{\Pi,l}}, t_k\right)} \omega_{i,k} \left| s^{[k]_{\Pi,l}} - s^{[k-1]_{\Pi,l}} \right| + \omega_r \right). \end{aligned} \tag{2}$$

In which, s means the time cost of finish the benchmark scheduling. t refers to the fixed cost caused by the execution response of each assembly line project. (s, t, O, v) means that in the No. $J(s, t) \cup O$ response, the unit time cost caused by the start time deviation of the assembly line project (s, t, O, v) is increased. s means that there are dead-end penalty factors in the execution chain of the assembly line project.

The cost function is composed of three parts: (1) From the perspective of management, cost of the benchmark scheduling process is the cost for the violation of delivery time of the assembly line project that has been agreed upon initially. (2) Cost of sequence assembly line project reaction, in which t can be taken as the cost to re-plan the assembly line project. O can be taken as the administrative (fixed) cost to process the assembly line project reaction. (3) The penalty cost caused by

the dead chain.

Parameter v, (s,t,O,v) and (s,t',O',v) can be used to determine the weight share of each part in the combined cost. The PR-strategy II is described by a group of decision rules and it also can be described by a group of relevant execution sequence chains:

$$\Pi : \left\{\Phi_{\Pi,1}, \Phi_{\Pi,2}, \cdots, \Phi_{\Pi,|\beta|}\right\} . \tag{3}$$

The description enables us to calculate the expected combined cost of PR-strategy II:

$$O' \subseteq O \cup J(s,t) . \tag{4}$$

Similar to the combined cost of execution chain, the expected cost of PR-strategy II is also composed of three parts: the cost of benchmark scheduling process, the cost of sequence assembly line project reaction and the penalty cost caused by dead chain.

2.3. Optimization of objective model

According to the execution chain of the assembly line project of solution set, the problems can be expressed in a relatively compact form, so that it has advantages in execution efficiency. Specially it can be expressed by the following theorems:

Theorem 1: PR-strategy scope expressed on the basis of execution chain of assembly line project is bounded, its upper bound is $|\beta| \sum_{i\in N} m_i$, in which m_i refers to the discrete duration of the different assembly line projects.

Demonstration: assume that each PR-strategy contains $|\beta|$ execution sequence chains, the amount of which might be pretty huge. First, it is demonstrated that each execution sequence chain of the PR-strategy II has $\sum_{i\in N} m_i$ responses at most. In the meantime, a random schedule $(s, t \to t', O \to O', v)$ selected from (s,t',O',v) can be considered. If schedule (s,t,O,v) is feasible to p^l, PR-strategy II will produce no response and the size of execution sequence chain i is 1; otherwise, PR-strategy II shall execute the response related to scheduling strategy $i \in O \cap O'$ on the moment t'. Assume there is an assembly line project i that will make the schedule t infeasible to t', then:

$$\alpha\left(s^{[0]_\Pi}, p^l, i_1\right) < \alpha\left(s^{[1]_{\Pi,l}}, p^l, i_1\right) . \tag{5}$$

In which, i is the maximum value of $i \in J(s,t) \cap O'$ to make the schedule t feasible to t'. Also, it is noticed that the assembly line project i might make the other scheduling strategy t' of the execution sequence chain t infeasible; similarly, it can be defined as:

$$\alpha\left(s^{[k]_{\Pi,k}}, p^l, i_1\right) < \alpha\left(s^{[1]_{\Pi,k+1}}, p^l, i_1\right) . \tag{6}$$

Different $\alpha(s,\rho^i,i)$ can be obtained through the above-mentioned defini-tion, in which whether s is the random scheduling scheme, the maximum scope is m_i, the maximum number of the infeasible assembly line project exe-cution chain caused by the assembly line project i[8] and the response number to be solved shall be m_i.

The similar argument is tenable to any other assembly line projects $i \in N\backslash\{i_1\}$; therefore, the direct conclusion can be obtained, namely each chain contains the most response of $\sum_{i\in N} m_i$. Accordingly, the PR-strategy scope expressed on the basis of execution chain of assembly line project is bounded, its upper bound is $|\beta| \sum_{i\in N} m_i$.

All possible PR-strategy sets II are defined, which means the PR-strategy sets can be constructed with (s, t, O, v), what we should do is to simplify and express $(s', t, O, v + 1)$ as below:

$$(s, t, O, v) \ . \tag{7}$$

In this formula, $(s', t, O, v+1)$ is a problem very difficult to solve, even to a small example. The solution problem of F is simplified into the solution problem of O, in which the small strategy sets are included:

$$P : \min_{\Pi\in\Pi_1} \sum_{l=1}^{|\beta|} \pi \left(\tilde{p} = p^l\right) f \left(\Pi, l\right) \ . \tag{8}$$

In the above optimization models, $U\left(s, t\right)$ is the schedule only included in Samong all PR-strategies.

3. Two-stage multi-objective fruit fly optimization algorithm

In this chapter, a two-stage multi-objective fruit fly optimization algorithm is proposed to carry out the multi-objective problem optimization. At the first stage, the wo knowledge bases of the original FOA variant initialization are adopted to determine the original fruit fly population location, afterwards the fruit flies are led by other food sources to fly to the non-inferior solutions. The first knowledge base is used to store the non-dominant solutions of the current iterations and copy them into the external files. The second knowledge base is used to store the reference points to represent the minimum parameter of the individual objective function of the current iteration and renew to make it converge to the real non-dominant solution. What's more, the solution structure of fruit fly is adjusted to improve the diversity of the non-dominant solutions. During this process, a new solution is explored by randomizing around the nearest members stored in the non-dominant solution base or the surrounding reference points (the single objective optimum value of each objective function). At the second stage, the weight method is adopted to transform the single objective optimized intensive search to obtain the locations of fruit fly population and the reference point. The specific steps of the two-stage multi-objective fruit fly optimization algorithm are as below:

Step 1: (algorithm initialization) the fruit fly is randomly initialized within the search space to make the following constraint satisfied:

$$\boldsymbol{x}\text{-}axis_i = x_{ij}^L + rand() \times (x_{ij}^U - x_{ij}^L) \, . \tag{9}$$

In this formula, $i = 1, 2, \cdots, PS$.

Step 2: (replacement process) in the traditional FOA algorithm, the PS food source of population is produced in a random way at the current location of the fruit fly, see formula (4), in which:

$$\boldsymbol{x}_i\text{-}axis{=}\boldsymbol{x}_{\text{best}}, if\ f(\boldsymbol{x}_{\text{best}}) < f(\boldsymbol{x}_i\text{-}axis) \tag{10}$$

However, to carry out the multi-objective improvement on the single-objective fruit fly algorithm, the initial location of the fruit fly population shall be redefined; to guarantee the favorable start-up performance of the algorithm, the individual fruit fly is dominated to replace from the nearest non-dominant individuals, the specific process is as below:

Step 2.1: the minimum value of individual objective is used to determine the optimum food source, $\boldsymbol{x}_{best,k}$:

$$RP = \{\boldsymbol{x}_{best,k}\}_{k=1}^{K} = \{\arg(\min_{i=1,2,\cdots,PS} f(\boldsymbol{x}_i))\}_{k=1}^{K}. \tag{11}$$

In addition, knowledge base $E^{(t+1)}$ and $A^{(t)}$ are respectively sued to store the non-dominant solutions in the current and previous iterations:

$$E^{(t+1)} = ND(\{f(\boldsymbol{x}\text{-axis}_i)\}_{i=1}^{PS}), A^{(t)} = \{E^{(t)}\}. \tag{12}$$

In which, $ND(\cdot)$ means all non-dominant solution functions that can be tested.

Step 2.2: different from the standard fruit fly algorithm that uses the origin distance and odor concentration judgment value, to determine the neighborhood location, a method using different concentration distances and concentration judgment values of sense of smell is proposed, in which Euclidean distance characterizes the distance to the fruit fly location to the objective space in the current non-dominant knowledge base. The knowledge base RP and the No. i dominant fruit fly, namely the No. i dominant fruit fly is the nearest alternative non-dominant individual in $E^{(t+1)}$ or RP. The distance and odor concentration judgment value (S) can be stated as below:

$$\begin{aligned} d(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) &= ||f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)|| \\ &= \sqrt{\sum_{k=1}^{K} \left(f_k(\boldsymbol{x}_i) - f_k(\boldsymbol{x}_j)\right)^2} \end{aligned} \tag{13}$$

$$S_j = 1/d(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)). \tag{14}$$

In this formula, $i \in RP \vee E^{(t+1)}$ and $j \notin E^{(t+1)}$. Assume there are six fruit flies randomly starting up in the search space as shown in Figure 1.

Figure 1 indicates that the solution is divided into two groups, namely the non-dominant solution (i.e. $\{\text{-}axis_1, \text{-}axis_3, \text{-}axis_4\}$) and the dominant solution (i.e. $\{\text{-}axis_2, \text{-}axis_5, \text{-}axis_6\}$). As a result, this algorithm has replaced the dominant fruit fly with the nearest non-dominant individual and guaranteed the favorable performance of the start-up point. For example, $\text{-}axis_6$ is the nearest individual fruit fly to

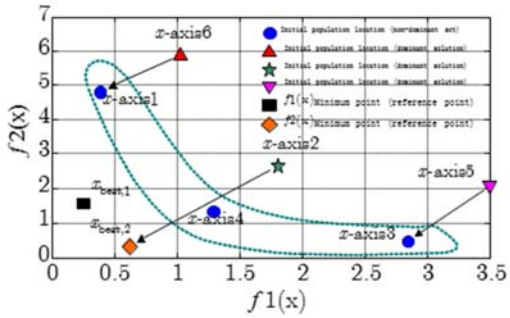

Fig. 1. Graphical representation of fruit fly location

-$axis_1$, so that it has the judgment value of relatively the maximum concentration and -$axis_6 \Rightarrow$ -$axis_1$. The start-up point of all fruit flies in the population is:

$$\begin{aligned} &\{\boldsymbol{x}\text{-}axis_1 = \boldsymbol{x}\text{-}axis_1, \boldsymbol{x}\text{-}axis_2 = \boldsymbol{x}_{best,2}, \\ &\boldsymbol{x}\text{-}axis_3 = \boldsymbol{x}\text{-}axis_3, \boldsymbol{x}\text{-}axis_5 = \boldsymbol{x}\text{-}axis_3, \\ &\boldsymbol{x}\text{-}axis_6 = \boldsymbol{x}\text{-}axis_1\} \end{aligned} \tag{15}$$

Step 3: (solution structure of sense of smell) the standard fruit fly algorithm is generates food source with radius 1 outside its population location; see Formula (4∼5). This radius is fixed and cannot be changed during the iteration optimization process. Therefore, a new control parameter is introduced to adjust the search, so that it can maintain the diversity of fruit fly population during the convergence process and realize the effective exploration around the fruit fly location (see Figure 2, in which the search radius is $R = 10$ and dynamically decreased as the number of iterations increases). As a result, the construction of fruit fly's sense of smell is as follows:

$$\boldsymbol{x}_i = x\text{-}axis_i \pm rand() \cdot \Delta_i\,(t, R)\ . \tag{16}$$

$$\Delta_i(t, R) = R\cos\{\frac{\pi}{2}\sin\frac{\pi}{2}[\sin(\frac{\pi}{2}\sin(\frac{\pi t}{T}))]\}\,. \tag{17}$$

In which, $t = 1, 2, \cdots, T$.

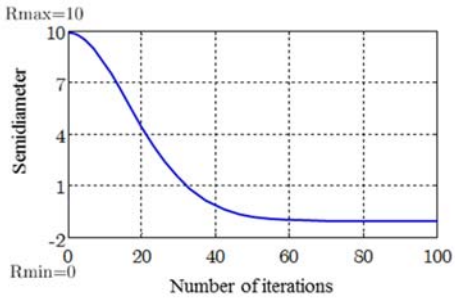

Fig. 2. Declining trend of search radius

As the algorithm operates, $\Delta_i(t, R)$ is introduced to cover the entire search space. Also, the random one-step $\Delta(t, R)$ changes dynamically along with the number of iterations and returns to a value within the interval $[0, R]$; it is gradually approaching 0 as the number of iterations is increased. The algorithm process is seen in the pseudo-code 1.

Pseudo-code 1: solution construction process of sense of smell

1. Input: $R = X^U - X^L$, T, $\boldsymbol{x}_i - axis$, $i = 1, 2, \cdots, PS$, $t = 1$
2. *for* $i = 1 : PS$ *do*
3. $\varepsilon = rand(0, 1)$;
4. $\boldsymbol{x}_i = \boldsymbol{x}_i\_axis \pm rand().\Delta_i(t, R)$;
5. if $\boldsymbol{x}_i\_axis > X^U$ then set $\boldsymbol{x}_i\_axis = X^U$
6. else if $\boldsymbol{x}_i\_axis < X^L$ then set $\boldsymbol{x}_i\_axis = X^L$
7. *endif*
8. *endfor*
9. *if* $\boldsymbol{x}_i \notin \Omega$ *then* $t = t + 1$ *endif*
10. output:$\boldsymbol{x}_i \in \Omega$

Step 4: (evaluation of the individual fruit fly) the dominant concept is adopted to evaluate the fruit fly individual, in which the non-dominant solution is stored in the knowledge base $E^{(t+1)} = ND(\{f(x - axis_i))_{i=1}^{PS})$; the optimum food source $\boldsymbol{x}_{best,k}$ related to the minimum value of each objective can be calculated as below:

$$RP^{(t+1)} = \{\boldsymbol{x}_{best,k}\}_{k=1}^{K} = \{\arg(\min_{i=1,2,\cdots,PS} f(\boldsymbol{x}_i))\}_{k=1}^{K}. \tag{18}$$

The above-mentioned optimum food source $\boldsymbol{x}_{best,k}$ is restored in the knowledge base RP.

Step 5: (visual foraging location renewal) there exist a group of sense of vision instead of the single sense of vision to renew the location as follows:

Step 5.1: The optimum food sources with minimum value of objective that are discovered earlier will replace the population location and renew in the next iteration if they are superior to the previous locations stored in $RP^{(t)}$, which means:

$$\forall k, \boldsymbol{x}_{best,k} = \boldsymbol{x}_{best,k}^{t+1} \; if \; f_k(\boldsymbol{x}_{best,k}^{t+1}) < f_k(\boldsymbol{x}_{best,k}^{t}). \tag{19}$$

Step 5.2: according to the method in Step 2.2, it can obtain better effect at the start-up point by replacing the dominant individual fruit fly with the nearest non-dominant individual. What's more, if $\boldsymbol{x}$-axis$_i(t)$ and $\boldsymbol{x}$-axis$_i(t+1)$ are respectively members of $A^{(t)}$ and $E^{(t+1)}$, one of them is the minimum distance to knowledge base $RP^{(t+1)}$. The selection of new start-up point is as shown in Figure 3.

Step 6: (renews the external archives) the optimum archive of external Pareto is updated as follows:

Step 6.1: Members of $E^{(t+1)}$ are copied into the external archives.

Step 6.2: search the non-dominant fruit flies in the external Pareto archive and

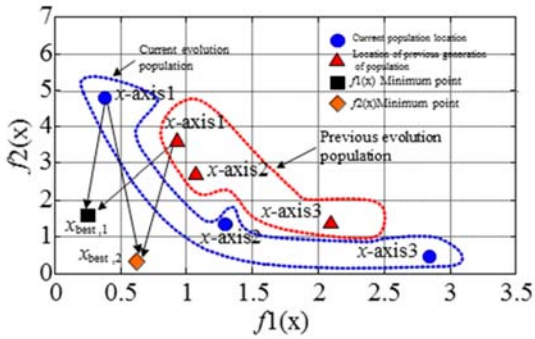

Fig. 3. Bon-dominant fruit fly renewal strategy

delete all dominant solutions in the archive.

Step 7: (the internal circulation stops) if the number of iterations are beyond the maximum limit, the internal circulation shall stop; otherwise it shall be transferred to **Step 3**.

Step 8: (diversity renewal of weighted phase) for the issue of multi-objective optimization, the scalar function of objective function is given as follows:

$$Smell(\boldsymbol{x}_i) = \sum_{k=1}^{K} w_k f_k(\boldsymbol{x}_i)\,. \tag{20}$$

$$w_k = \frac{random_k(\cdot)}{\sum_k^K random_k(\cdot)}\,. \tag{21}$$

In this formula, $k = 1, 2, \cdots, K$. The program steps of this stage are as below: first of all, formula (9) is used to initialize the population and make it satisfy the constraint conditions. Secondly, the weight is generated in a random way and substituted into formula (20) to evaluate the fruit fly population. In addition, the best location $\boldsymbol{x}_{best}^{(t)}$ of fruit fly can be determined by the following formula:

$$\boldsymbol{x}_{best}^{(t)} = \arg\left(\min_{i=1,2,\cdots,PS}(Smell_i)\right) \tag{22}$$

In the end, the non-dominant solution is tested and transferred to the external archive; all dominant solutions are deleted from the archive. In addition, the reference point is renewed as:

$$\forall k, \boldsymbol{x}_{best,k} = \boldsymbol{x}_{best,k}^{t+1}\ if\ f_k(\boldsymbol{x}_{best,k}^{t+1}) < f_k(\boldsymbol{x}_{best,k}^{t})\,. \tag{23}$$

Step 9: (stopping criterion) the external circulation shall stop when the number of iterations of the entire algorithm is beyond the maximum limit; otherwise it will be transferred to **step 2**.

4. Experiment analysis

To verify the effectiveness of the algorithm, seven multi-mode standard test examples of different scopes namely J10, J12, J14, J16, J18, J20 and J30 are selected from the assembly line project library; each group is respectively composed of 10, 12, 14, 16, 18, 20 and 30 non-virtual assembly line projects, 3 modes, 2 renewable resources and 2 non-renewable resources. Each group contains 640 examples but some of the examples can be eliminated for having no feasible solutions, so there are 3290 examples in total.

The algorithm is realized through the Matlalb language programming; the Dell desktop with Intel i5 3.2 GHz and 6GB ddr4-2400kB memory is adopted to operate in the Windows10 operational environment. Table 1 has listed the computing result after 600 iterative computations as well as the comparation between the computing results of document [14] and [15]; the comparison indicators are the average variance and the percentage to obtain the optimum solution.

Table 1. Computing result of benchmark examples

Number of assembly line projects	Feasible example	Document [14]		Document [15]		Proposed algorithm	
		Accuracy/%	Time/s	Accuracy/%	Time/s	Accuracy/%	Time/s
10	534	0.23	12.3	0.21	14.2	0.18	10.3
12	546	0.18	12.5	0.16	14.5	0.12	10.2
14	552	0.93	12.8	0.87	14.8	0.75	10.8
16	551	1.42	13.4	1.35	15.2	0.83	11.2
18	553	1.86	13.7	1.74	15.3	0.92	11.4
20	556	2.11	14.1	2.07	15.7	1.27	11.5
30	564	2.25	15.3	2.14	16.3	1.36	12.4

In the experiment data of Table 1, the convergence accuracy and computational efficiency are taken as the valuating indicators to compare the performance superiorities between the proposed algorithm and the selected two comparative algorithms. According to data in Table 1, in terms of convergence accuracy, in contrast with the percentage of the set value, the deviation is approximately between 0.12% and 1.36%. On the other hand, the accuracy indication of document [14] is distributed between 0.18% and 2.25%, the accuracy indication of document [15] is distributed between 0.16% and 2.14%; the proposed algorithm has obvious advantages in terms of convergence accuracy. In the meantime, in terms of the assembly line project scheduling time, the three algorithms have less difference, the proposed algorithm is slightly superior to the two contrastive algorithms in document [14] and [15].

Figure 4 provides the computing Gantt chart distribution of the proposed algorithm as well as document [14] and [15] when the number of selected assembly line projects is 30.

It can be seen from the computing Gantt chart distribution of the three algorithms in Figure 4, compared with the two algorithms in document [14] and [15], in the Gantt chart of the assembly line project execution obtained by the algorithm in this paper, the total execution time of the assembly line project is significantly

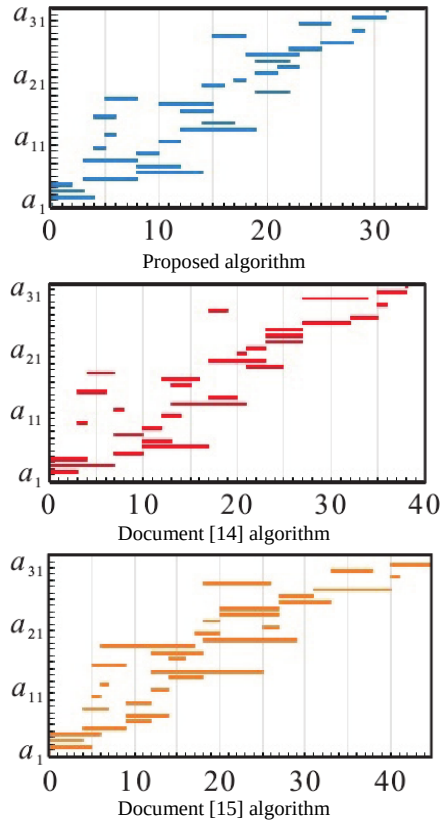

Fig. 4. Computing Gantt chart distribution of three algorithms

superior to the two algorithms in document [14] and [15]. It has indicated the algorithm performance superiorities of the proposed algorithm during the assembly line project scheduling; the scheme of assembly line project scheduling scheme is more reasonable.

5. Conclusion

An optimization method for the assembly line project scheduling based on random graph Markovian decision process (MDP) of the multi-objective fruit fly algorithm is proposed, a new method is introduced to cope with conflict, once there is conflict in the scheduling scheme, instead of redefining the start time of the assembly line project, it is to transfer the state of the time schedule of conflict to obtain another feasible time schedule and improve the computational efficiency of algorithm; the model optimization is realized through the improved multi-objective fruit fly algorithm that has been proposed.

References

[1] 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.

[2] 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.

[3] Y. Y. Zhang, Q. Li, W. J. Welsh, P. V. Moghe, and K. E. Uhrich: (2016) *Micellar and Structural Stability of Nanoscale Amphiphilic Polymers: Implications for Anti-atherosclerotic Bioactivity*, Biomaterials, 84, 230-240.

[4] L. R. Stephygraph, N. Arunkumar, V. Venkatraman: *Wireless mobile robot control through human machine interface using brain signals*, 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, ICSTM 2015 - Proceedings, art. no. 7225484 (2015), 596–603.

[5] X. Sun, Y. Xue, C. Liang, T. Wang, W. Zhe, G. Sun, X. Li, X. Li, G. Liu: *Histamine Induces Bovine Rumen Epithelial Cell Inflammatory Response via NF-κB Pathway.* Cellular Physiology & Biochemistry *42* (2017), No. 3, 1109–1119.

[6] N. Arunkumar, V. S. Balaji, S. Ramesh, S. Natarajan, V. R. Likhita, S. Sundari: *Automatic detection of epileptic seizures using independent component analysis algorithm.* IEEE-International Conference on Advances in Engineering, Science and Management, ICAESM-2012, art (2012) No. 6215903, 542–544.

[7] Y. Du, Y. Z. Chen, Y. Y. Zhuang, C. Zhu, F. J. Tang, J. Huang: *Probing Nanostrain via a Mechanically Designed Optical Fiber Interferometer.* IEEE Photonics Technology Letters *29* (2017), 1348–1351.

[8] 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).

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Z. Lv, A. Halawani, S. Feng, H. Li, S. U. Réhman: *Multimodal hand and foot gesture interaction for handheld devices.* ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) *11* (2014), No. 1s, 10.

[14] Y. Z. Chen, F. J. Tang, Y. Bao, Y. Tang, G. Chen: *A Fe-C coated long period fiber grating sensor for corrosion induced mass loss measurement.* Optics letters *41* (2016), 2306–2309.

Received May 7, 2017