Int j simul model 13 (2014) 4, 472-484
ISSN 1726-4529 Original scientific paper

MULTI-OBJECTIVE RESCHEDULING MODEL FOR PRODUCT COLLABORATIVE DESIGN CONSIDERING DISTURBANCE

Yu, G. D.[*]; Yang, Y.[*]; Zhao, X.[**] & Li, G.[*]
[*] State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing, China
[**] Qingdao Cigarette Factor, China Tobacco Shandong Industrial Corporation, Qingdao, China
E-Mail: yuguodong@cqu.edu.cn

Abstract

Disturbance is inevitable in product collaborative design (PCDSP), which has always posed a great challenge for enterprises making quick response. Thus, in the paper, a multi-objective rescheduling model and its solution algorithm are presented. Disturbance cases are analysed and dynamic scheduling procedure based on event-driven and lifecycle-driven is developed firstly. Then a multi-objective rescheduling model aiming to minimum the makespan and tardiness penalty is developed. As a solution, multi-objective dynamic adaptive scheduling algorithm (MODASA), based on bi-layer coding strategy, self-adaptive double point crossover and self-adaptive mutation, is proposed following closely. Finally, analytic results from a case of a wind turbine are used to illustrate the model and method proposed in this paper. Simulation results shows that the model and algorithm have full advantages in computing speed and precision. With the analysis, it can provide insight into ways of improving the strategic and operational decision making for enterprises.

(Received, processed and accepted by the Chinese Representative Office.)

Key Words: Product Collaborative Design, Rescheduling Problem, Disturbance, Multi-Objective Optimization, Algorithm

1. INTRODUCTION

Product Collaborative Design (PCD) has received an increasing attention with the prosperity of market and rapid development of technology [1]. As a distributed cooperative system, one of the Gordian knots is guaranteeing all tasks succeed in while ensuring design cost and customer satisfaction over the duration of development under uncertain environment [2]. Actually, due to the multifarious indeterminate factors of PCD, inevitable perturbations such as rush orders, customer requirements changes, customer churn and employee turnover are always hindering production progress and even lead to mission abort if handled improperly [3]. In this situation, it is fundamentally crucial to devise a PCD scheduling (PCDS) strategy for reducing the impact and keep high efficiency.

Attempts to solve this dilemma have resulted in the development of some effective ways, which mainly focuses on two aspects, namely the *scheduling model* and *algorithm*. For *scheduling model*, Belhe and Kusiak used directed graphs represent multiple design projects competing for the limited available resources and the scheduling problem is decomposed into a series of multidimensional (multisource) knapsack problems [4]. Luh et al. developed a scheduling model based on LR and SDP to resolve the design projects with uncertain number of iterations [5]. Artigues et al. proposed a flow network model taking advantage on the flow structure to resource-constrained project static and dynamic scheduling problem [6]. Danilovic and Browning formalized an approach using a domain mapping matrix (DMM) to compare two DSMs of different project domains, then activities schedule is conducted based the priority of the projects [7]. Pan et al. developed a model based on synchronous mapping from weighted directed graph to DSM and polychromatic sets to transform related information needed for design task scheduling [8]. Browning and Yassine addressed the static

DOI:10.2507/IJSIMM13(4)CO17

resource-constrained multi-project scheduling problem (RCMPSP) with two lateness objectives, project lateness and portfolio lateness [9]. Li et al. proposed a multi-mode optimal scheduling of product design project based on person-task-resource matching degree [10]. Lapegue et al. presented a CP model which deals simultaneously with the assignment of tasks and the design of personnel schedules in order to solve a real-world problem [11]. For *algorithm*, there are many kinds of improved solutions. Lam et al. developed a scheduling Genetic Algorithm (GA) to minimize product design time [12]. Gomez-Gasquet et al. developed a modified GA for collaborative scheduling of products-packages service [13]. Krishnamoorthy et al. proposed a Lagrangean approach for solving the Personnel Task Scheduling Problem (PTSP) [14]. Besides, competent genetic algorithms [15], ant colony algorithms [16], particle swarm optimization (PSO) [17], simulation [18], simulated annealing (SA) [19] and many other algorithms and methods [20-21] were used in PCD scheduling.

In summary, most existing approaches have made some achievements. However, extra-absolute of these researches, such as fuzziness considered inadequately, inaccuracy of multipurpose optimization and neglect of accidents also created some problems in the practical application. In this paper, a multi-objective rescheduling model (MRM) and its algorithm considering the emergencies in PCD are presented to solve above dilemma. Based on the general information of PCD, dynamic scheduling procedure combining the event-driven and the lifecycle-driven is proposed to manage possible perturbation, Hereafter, regard makespan and tardiness penalty caused by emergency as the objectives of multipurpose optimal model, meanwhile implement a multi-objective heuristic genetic algorithm, multi-objective dynamic adaptive scheduling algorithm (MODASA), as a model solution.

2. PROBLEM FORMULATION

2.1 Basic scheduling model

Suppose that a PCD organization contains *m* designers and a product to be designed is decomposed into *n* tasks after careful analysis. Then, the basic objective is that all tasks are assigned to designers excellently based on the sequence of activities. Simultaneously, time parameters of each task are clear and definite. Moreover, the makespan is minimum in that state, as shown in eq. (1).

$$f_1 = \min\{\max(CT_i \mid i = 1,2,\cdots,n)\} \tag{1}$$

where CT_i denotes the completion time of the i^{th} task.

To accomplish this, we begin by defining the following basic definitions and descriptions.

Definition 1. *Tasks set*. Let $Ta = \{Ta_i \mid 1 \le i \le n\}$ denote all design tasks of a product. Ta_i is composed by n_i sorted processes $P_i = \{P_{ij} \mid 1 \le j \le n_i\}$.

Definition 2. *Designers set*. Let $D = \{D_k \mid 1 \le k \le m\}$ denote all designers available of PCD. Besides, D_{ij} indicates the designers can undertake P_{ij}, $D_{ij} \subseteq D$. That is to say, P_{ij} can be accomplished by any one of D_{ij}, however, the time varies.

Definition 3. *State set of Task*. Each task has four states: unresolved S_1, to be resolved S_2, being resolved S_3 and resolved S_4.

Description 1. All *designers* are available and all tasks can be started at the initial time $t = 0$.

Description 2. P_{ij} is *executed* only when $P_{i(j\text{-}1)}$ is accomplished.

$$CT_{ijk} - CT_{i(j-1)g} \ge PT_{ijk} + ST_{ijk} + ET_{ijk}\ , j\neq 1 \tag{2}$$

where CT_{ijk}, PT_{ijk}, ST_{ijk} and ET_{ijk} denotes the completion time, the processing time, the set-up time and the ending time of P_{ij} designed by D_k respectively.

Description 3. The *value* of execution time of P_{ij} must not be negative.

$$CT_{ijk} - PT_{ijk} \geq ST_{ijk} + ET_{ijk} \tag{3}$$

Description 4. *Introduce* a binary variable X_{ijk} indicating the relationship of P_{ij} with D_k, such that $X_{ijk} = 1$ means that P_{ij} is undertaken by D_k and $X_{ijk} = 0$ indicates that D_k is not selected to complete P_{ij}. Similarly Y_{rsijk} is a binary variable, such that $Y_{rsijk} = 1$ indicates that P_{rs} precedes P_{ij} by D_k and otherwise $Y_{rsijk} = 0$.

Description 5. P_{ij} *should* be implemented by one of D_{ij} over the duration of the development.

$$\sum_k X_{ijk} = 1, (k \in D_{ij}) \quad \forall i, j \tag{4}$$

Description 6. At any moment, only one process P_{ij} occupies D_k.

$$CT_{ijk} - CT_{rsk} \geq PT_{ijk} + ST_{ijk} + ET_{ijk} \quad Y_{rsijk} = 1, X_{ijk} = 1, X_{rsk} = 1 \tag{5}$$

Description 7. For *each* task or process after the first one, the completion time is subject to the following rule.

$$CT_{ijk} = \max\left\{CT_{i(j-1)l}, BT_{ijk}\right\} + PT_{ijk} + ST_{ijk} + ET_{ijk} \tag{6}$$

where BT_{ijk} means the start time of P_{ij} undertaken by D_k. If $j = 1$ especially, then

$$CT_{i1k} = BT_{i1k} + PT_{i1k} + ST_{i1k} + ET_{i1k} \tag{7}$$

According to above definitions, descriptions and eqs. (1-7), a basic scheduling formulation can be established on the context of ideal condition where all time parameters are explicit and there are no any interference factors.

2.2 Disturbance

(1) Disturbance cases

PCD involves multidisciplinary synergy, cross-organizations and distributed designers. Due to rush orders, customer requirements changes, and customer churn etc., collaborative work environment is always in a state of dynamic change, which leads to a greatly increasing complexity and uncertainty of task procedures. This scenario argues that scheduling procedure should take impossible emergencies into consideration. An immediate consequence of emergency is task tardiness. Thus, tardiness penalty f_2 is considered to minimize effect of emergency. For PCD, tardiness occurs once $CT_i > DD_i$. Let λ_i indicate the penalty coefficient of T_i, the unit tardiness of tardy Ta_i.

$$f_2 = \min\left(\sum_{i=1}^{n} T_i\right) = \min\left(\sum_{i=1}^{n} \lambda_i \times \max\left(0, CT_i - DD_i\right)\right) \tag{8}$$

(2) Rescheduling Procedure

Event-driven, lifecycle-driven and hybrid-driven are three most frequent dynamic scheduling methods to cope with emergency [19]. Event-driven means that rescheduling is activated only when task set changes because of accidents. Lifecycle-driven implies that rescheduling is implemented every once in a while no matter what happens at set intervals. Obviously, lifecycle-driven scheduling is able to maintain task set at certain stability but incapable of tackling emergencies while event-driven is just the reverse. Thus, hybrid-driven scheduling, synthesizing advantages both of event-driven and lifecycle-driven, can not only deal with unexpected incidents but also keep the relative stability of the collaborative system. Therefore, this paper stresses on the dynamic scheduling based on hybrid-driven, as shown in Fig. 1.

On the premise of minimizing the makespan of PCD, dynamic scheduling should reduce the tardiness penalty where possible. Based on eqs. (1) and (8), the optimization objective function containing completion time and tardiness penalty can be formed as eq. (9).

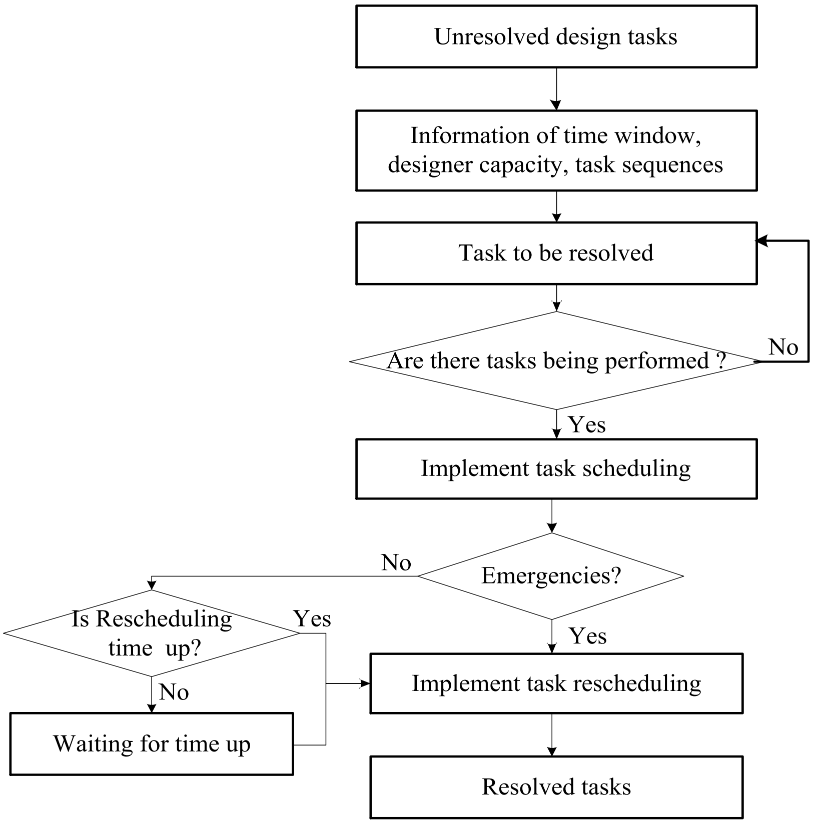
Unresolved design tasks
Information of time window, designer capacity, task sequences
Task to be resolved
Are there tasks being performed ?
No
Yes
Implement task scheduling
Emergencies?
No
Yes
Is Rescheduling time up?
Yes
No
Waiting for time up
Implement task rescheduling
Resolved tasks

Figure 1: Dynamic scheduling procedure of PCD.

$$\min G = \max_{k \in M}\left(T_k - bt_h\right) + \lambda_i \sum_{i \in Ta_{i,h,S_3}} \max\left(0, CT_i - DD_i\right) \tag{9}$$

$$T_k = RT_k + \sum_{j \in Ta_{j,S_1,k}} \left(PT_{jk} + ST_{jk} + ET_{jk}\right) + bt_h \tag{10}$$

$$bt_{h+1} = \min_{i \in Ta_{i,h,S_3}} \left(bt_h + \Delta T_h + t_e\right) \tag{11}$$

Here, T_k denotes the time that M_k accomplish the last tasks. bt_h indicates the start time of the h_{th} rescheduling. Ta_{i,h,S_3} implies the set of tasks being resolved while the h_{th} rescheduling starts. $Ta_{i,S_1,k}$ means the set of tasks in state S_1 by M_k. RT_k represents the remaining time of M_k to fulfill the tasks in S_3. ΔT_k signifies the h_{th} rescheduling period. t_e expresses the time that rescheduling occurs.

2.3 Multi-objective dynamic fuzzy scheduling model

Based on the aforementioned, PCDS is actually a multi-objective dynamic fuzzy scheduling problem. The objective functions can be described as eqs. (12) and (13). Constraints are shown in eqs. (14-20).

$$f_1 = \min\left\{\max(CT_i \mid i = 1, 2, \cdots, n)\right\} \tag{12}$$

$$f_2 = \min\left(\sum_{i=1}^{n} \lambda_i \times \max\left(0, CT_i - DD_i\right)\right) \tag{13}$$

$$CT_{ijk} - CT_{i(j-1)g} \geq PT_{ijk} + ST_{ijk} + ET_{ijk}, j \neq 1 \tag{14}$$

$$CT_{ijk} - PT_{ijk} \geq ST_{ijk} + ET_{ijk} \tag{15}$$

$$\sum_k X_{ijk} = 1, (k \in D_{ij}) \quad \forall i, j \tag{16}$$

$$CT_{ijk} - CT_{rsk} \geq PT_{ijk} + ST_{ijk} + ET_{ijk}\,,\; Y_{rsijk} = 1, X_{ijk} = 1, X_{rsk} = 1 \tag{17}$$

$$CT_{ijk} = \max\left\{CT_{i(j-1)l}, BT_{ijk}\right\} + PT_{ijk} + ST_{ijk} + ET_{ijk} \tag{18}$$

$$T_k = RT_k + \sum_{j \in Ta_{j,S_1,k}} \left(PT_{jk} + ST_{jk} + ET_{jk}\right) + bt_h \tag{19}$$

$$bt_{h+1} = \min_{i \in Ta_{i,h,S_3}} \left(bt_h + \Delta T_h + t_e\right) \tag{20}$$

3. SOLUTION PROCEDURE

3.1 Algorithm

The PCDS problem under study is NP hard even on the context of the deterministic processing time case. Accordingly, traditional optimization methods are hardly to achieve an optimal solution [22]. Many heuristic solutions, such as Genetic Algorithm (GA), Simulated Annealing (SA), Particle Swarm Optimization (PSO) and others, have been developed for optimization. Among these novel methods, GA has been well documented in the literature and obtains a predictive effect for a variety of optimization problems. Thus, in this paper, a model solution based on GA, multi-objective dynamic adaptive scheduling algorithm (MODASA) is proposed for the problem under study.

The MODASA achieves a dynamic flexible allocation of task-designer based on the bi-layer coding strategy formed by the fusion of task sequences and designers attributes. This method is able to call the designers and resources competent for tasks with the minimum tardiness penalties when the accident and even task suspension appears. Consequently, the influence of time delay can be reduced and the efficiency of PCD can be improved. Fig. 2 shows the algorithm process of the MODASA.

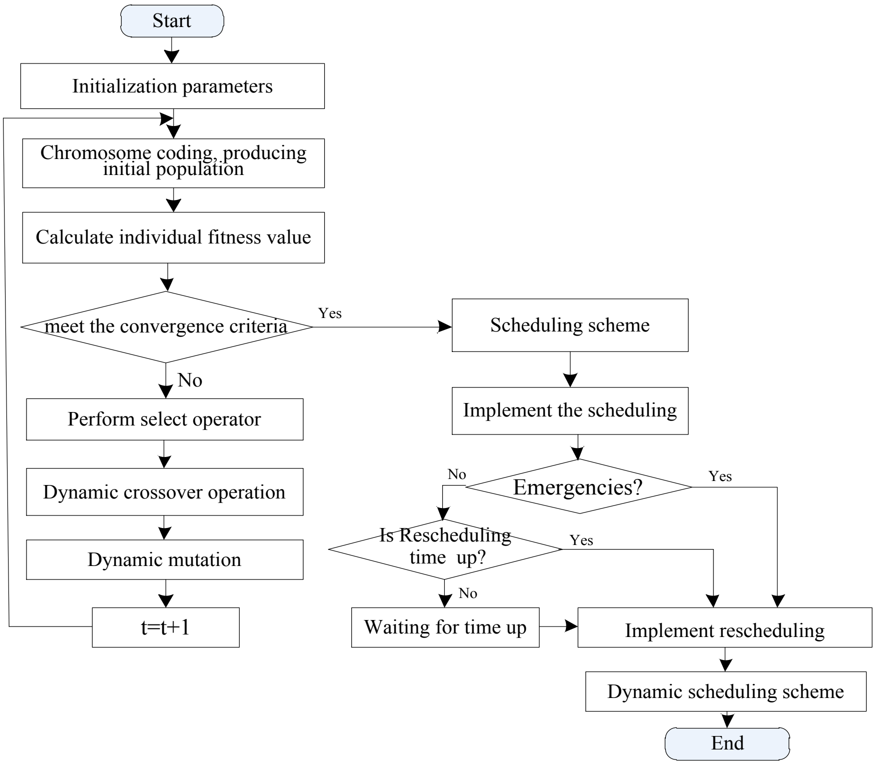
Start
Initialization parameters
Chromosome coding, producing initial population
Calculate individual fitness value
meet the convergence criteria
Yes
No
Perform select operator
Dynamic crossover operation
Dynamic mutation
t=t+1
Scheduling scheme
Implement the scheduling
Emergencies?
No
Yes
Is Rescheduling time up?
Yes
No
Waiting for time up
Implement rescheduling
Dynamic scheduling scheme
End

Figure 2: Process of the MODASA.

(1) Initialization Parameter

All related parameters should be initialized firstly. Let t indicate the evolutional generation, BT_k denote the start time of M_k performing a task and W means the maximum number of the tasks. $h=0$ and $bt_k=0$.

(2) Encoding Chromosome

According to the characteristic of this problem, a bi-layer coding strategy formed by the fusion of tasks sequences and designers attributes is proposed. The first layer is encoded based on design tasks sequences. All procedures of a task are specified a unified symbol and explained according to their appearance order in chromosome. Suppose a chromosome is [231122313], where 1, 2, 3, 4 and 5 represent Ta_1, Ta_2, Ta_3, Ta_4 and Ta_5 respectively. Due to each task includes 3 design sequences, thus each number appears five times in a chromosome. The second layer is designer code of the process, which means P_{ij} can be formed by any one of M_{ij}. As shown in Fig. 3, it is a 3×3 example. P_{21} of Ta_2 can be completed by one of M_{21}. By such analogy, the bi-layer coding strategy is straightforward able to achieve the replacement of resources when emergencies happen.

Chromosome	2	3	1	1	2	2	3	1	3
Process	P_{21}	P_{31}	P_{11}	P_{12}	P_{22}	P_{23}	P_{32}	P_{13}	P_{33}
Designer	M_{21}	M_{31}	M_{11}	M_{12}	M_{22}	M_{23}	M_{32}	M_{13}	M_{33}

Figure 3: An example of chromosome.

M_{ij} is selected to carry out P_{ij} according to the principle that the processing time is minimum and no conflict appears. If $j=1$, PT_{ijk} can be calculated by eq. (7), otherwise, eq. (6) will be used.

(3) Fitness

The objective value is calculated according to eqs. (12) and (13). Then define a joint objective function $F_i=f_1+\lambda_i f_2$ as the fitness of individual i.

(4) Selection

This paper uses the roulette wheel method to select the parent chromosomes i with probability $prob_i$ from the population with $popsize$ individuals.

$$prob_i = F_i \Big/ \sum_{i=1}^{popsize} F_i \tag{21}$$

(5) Crossover

This study adopts self-adaptive double point crossover to increase the population diversity and to prevent the operation of algorithm premature and stagnation. As shown in Fig. 4.

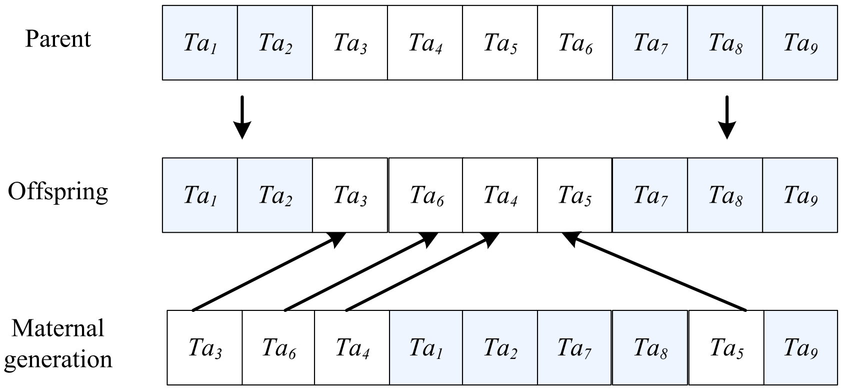
Parent
Ta_1 Ta_2 Ta_3 Ta_4 Ta_5 Ta_6 Ta_7 Ta_8 Ta_9
Offspring
Ta_1 Ta_2 Ta_3 Ta_6 Ta_4 Ta_5 Ta_7 Ta_8 Ta_9
Maternal generation
Ta_3 Ta_6 Ta_4 Ta_1 Ta_2 Ta_7 Ta_8 Ta_5 Ta_9

Figure 4: Double point crossover.

The self-adaptive function of crossover rate *prob_c* is defined as eq. (22).

$$prob_C = \begin{cases} prob_{C1} - \dfrac{(prob_{C1} - prob_{C2})(F_i - F_{avg})}{F_{\max} - F_{avg}} & F_i \geq F_{avg} \\ prob_{C1} \quad F_i < F_{avg} \end{cases} \tag{22}$$

where F_{avg} and F_{max} denote the average fitness and the maximum fitness respectively.

(6) Mutation

The mutation mechanism randomly selects a gene according to the mutation rate and alters its value. Meanwhile, adaptive mutation probability is used to improve the convergence speed, and the adaptive adjustment function is formulated as eq. (23).

$$prob_m = \begin{cases} prob_{m1} - \dfrac{(prob_{m1} - prob_{m1})(F_{\max} - F_i)}{F_{\max} - f_{avg}} & F \geq F_{avg} \\ prob_{m1} \quad F < F_{avg} \end{cases} \tag{23}$$

As shown in Fig. 5, Ta_6 is randomly selected and altered to Ta_9.

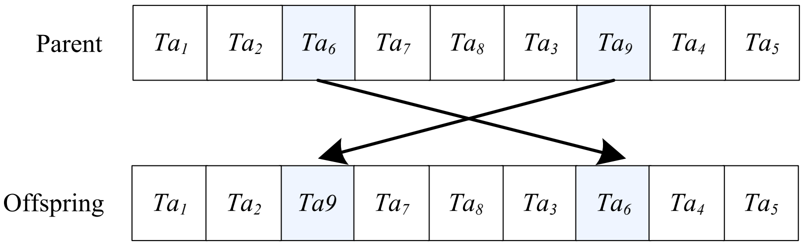

Figure 5: Mutation mechanism.

(7) Stopping rule

If the maximal generation is reached, namely $t = T$, then stop and output F_i and scheduling scheme; otherwise, perform the next iteration.

3.2 Dynamic scheduling

Based on all steps in Section 3.1, a common schedule can be completed. However, once an emergency occurs, additional measures may be adopted. In some cases, such as requirement changes, rush order and natural disaster, rescheduling must be launched immediately. At that moment, all related tasks and its processes are classified into S_2. Meanwhile, these tasks enjoy privileges of taking precedence over other tasks when the emergency settled. However, in another cases, such as operational error and technical improvement, completion status of tasks should be checked before rescheduling launched. If the task has been completed, rescheduling is needless. Otherwise, rescheduling must be activated promptly. The pseudo code of the MODASA for multiobjective scheduling is as following:

```
If T_item*T_count < T_emergency1 & T_item*T_count < T_emergency2 &
T_item*T_count < T_emergency3
    minfinishT = T_item*T_count;
    T_count = T_count + 1;
    Mark_TorE = 'T';
  else
    if  T_emergency1< $\alpha$
        designer_starttime(1) = inf;
```

```
            minfinishT = T_emergency1;
            T_emergency1 = inf;
        elseif T_emergency2< α
            designer_starttime(1) = T_emergency2;
            minfinishT = T_emergency2;
            T_emergency2 = inf;
        elseif T_emergency3< α
            due_date(i) =γ;
            minfinishT= T_emergency3;
            T_emergency3 = inf;
    end
Mark_TorE = 'E';
```

4. CASE STUDY

To evaluate the performance of the proposed model and algorithm, a case study is conducted in this section. As a complex product, a wind turbine usually consists of thousands of parts. Thus PCD is an applicable work patterns. Generally, a wind turbine contains 10 key components, which are equivalent to 10 design tasks. Suppose that each task includes 3 processes. All tasks are carried out by 3 designers from different institutions. Essential Information about tasks and designers are illustrated in Table I, where [175, 180] means that the optimum duration of due date is from 175 to 180. Without loss of generality, $ST_{ij}=ET_{ij}=1$, the penalty coefficient $\lambda_i=0.8$, the rescheduling period $\Delta T_k=8$, the population size $popsize$ = 500, the maximum generation $T=500$, the self adaptive crossover probability $prob_{c1}=0.8$, $prob_{c2}=0.6$, the mutation rate $prob_{m1}=0.1$, $prob_{m2}=0.001$.

During the PCD process, two emergencies occur. (1) M_1 interrupts the task being performed at $t=70$ because of the resource conflicts and restores the work at $t=90$. (2) Due to the customer requirements changes at $t=120$, the due date of Ta_{10} is advanced to [175, 185] from [245, 255].

4.1 Computational results

According to the solutions above, the value of the objective can be calculated, as shown in Fig. 6, which is the Gantt chart of PCD tasks in the context of without emergencies and considering emergencies. With the analysis, the maximum completion time of the wind turbine is 212 h, while it extends to 217 h when the two accidents emerge. Moreover, based on an assumption that we don't conduct the dynamic scheduling, as shown in Fig. 6, P_{72} is suspended at $t=70$ and P_{22} is tardy. In these circumstances, the PCD cycle is much greater than 317 h. Consequently, the dynamic scheduling played a positive role in improving work efficiency.

A comparison of the MODASA with the AACA [23], the SAGA [24], the MOGA [25] is conducted under the circumstances that $\Delta T=50$, $prob_{c1}=0.8$ and $prob_{m1}=0.1$. After running 50 times independently, the optimal results are shown in Fig. 7. And the best statistical results are shown in Table II. It can be seen that MODASA converges to the optimal result after 48 generations, MOGA converges to it after 76 generations, SAGA converges to it after 149 generations while SGA converges to it after 256 generations. When running on the computer with CPU core duo E4600, basic frequency 2.4 GHZ, 2 GB memory, the running time by MODASA is 16.47 s, the time by MOGA is 21.04 s, the time by SAGA is 24.15 s and it is 29.98 s by AACA. Thus, the MODASA has higher stability and shorter running time than others.

Table I: Essential information of tasks and designers.

No.	Task	Due date (h)	Sequences	Available Designer	Processing time (h)
1	Blade	[175,180]	1	1,3	[14,16],[15,18]
			2	1,2,3	[18,20],[19,22]
			3	1,2	[18,25,[14,16]
2	Hub	[145,155]	1	1,3	[18,23],[12,14]
			2	1,2	[18,19],[16,18],[11,14]
			3	2,3	[14-17],[15,18]
3	Pitch system	[175,185]	1	1,2,3	[14,15],[16,18]
			2	1,2	[18,19],[12,14]
			3	2,3	[18,20],[19,22],[11,14]
4	Engine	[165,175]	1	1,3	[18,21],[14,15],[19,22]
			2	1,2,3	[14,15],[17,18]
			3	2,3	[18,21],[12,14]
5	Yaw system	[180,200]	1	1,2	[22,23],[26,18]
			2	1,2,3	[18,19],[20,22]
			3	2,3	[22,23],[15,18],[16,18]
6	Transmitted chain	[155,165]	1	2,3	[18,21][14,17]
			2	1,2	[14,15][18,19],[13,14]
			3	1,2,3	[17,18],[13,14]
7	Generator	[195,205]	1	1,2	[14,15],[15,18]
			2	1,3	[18,,19],[12,14]
			3	1,2,3	[22,25],[12,14]
8	Control System	[190,196]	1	1,2,3	[22,23],[17,18]
			2	1,3	[22,23],[17,18]
			3	2,3	[18,19],[22,25],[17,18]
9	Tower Barrel	[195,200]	1	1,2	[14,15],[17,18]
			2	1,2,3	[18,19],[12,14],[15,18]
			3	1,3	[22,24],[12,14]
10	Frame	[245,255]	1	2,3	[22,23],[17,18]
			2	1,2,3	[18,19],[20,22],[17,18]
			3	1,3	[18,19],[20,22]

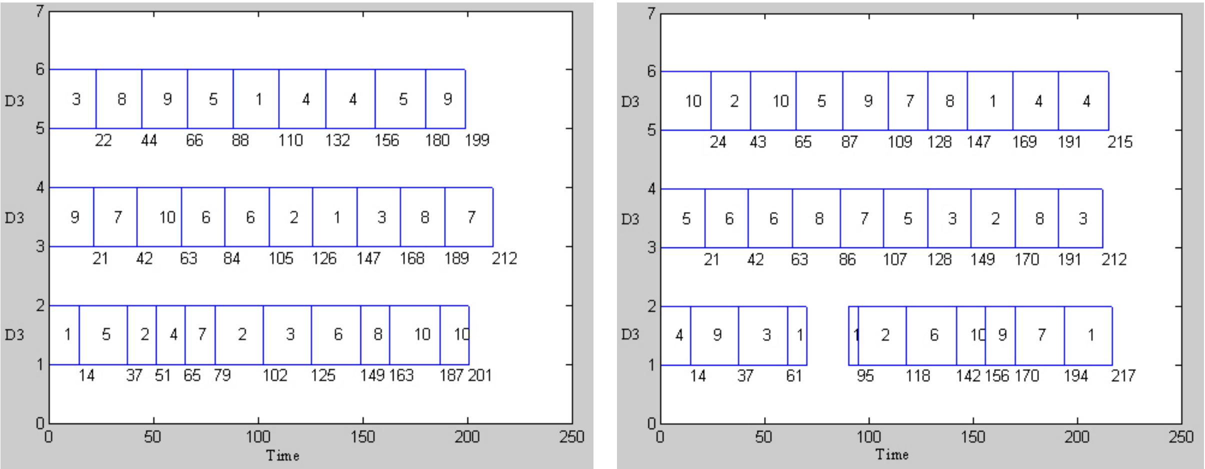

Figure 6: Gantt chart of PCD tasks in the context of without emergencies and considering emergencies.

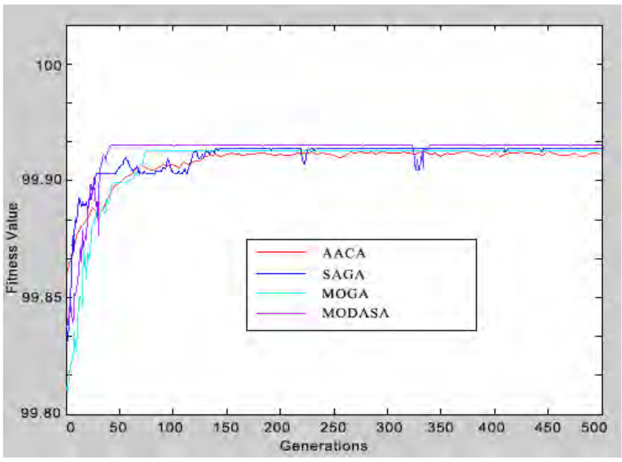

Figure 7: Results of AACA, SAGA, MOGA and MODASA.

Table II: Comparison of different algorithms optimized results.

Algorithm	Optimal result	Run time (s)	Iteration
AACA	99.9087	29.98	256
SAGA	99.9115	24.15	149
MOGA	99.9106	21.04	76
MODASA	99.9163	16.47	48

4.2 Further discuss

Furthermore, this paper analyses the impact of accidents, due date and dynamic scheduling period on the performance of PCD. Through the changes to the rescheduling period, due dates, study the variation of dynamic scheduling performance can be grasped. Analysis results are shown in Table III.

Table III: Impact of accidents, due date and dynamic scheduling period on the PCD performance.

ΔT	Task interrupt	Due date	$\bar{f}_1$	Deviation	$\bar{f}_2$	$\bar{t}$
50	(1:70-90)	(10:250-180)	217	2.30 %	137	17.99
		(9:195-160)	218	2.80 %	160	19.32
		(6:160-150)	220	3.80 %	161	24.21
50	(2:70-90)	(10:250-180)	220	3.60 %	140	19.6
		(9:195-160)	221	4.10 %	154	19.79
		(6:160-150)	232	8.60 %	157	20.72
50	(3:70-90)	(10:250-180)	221	4.10 %	159	19.9
		(9:195-160)	233	9.00 %	162	20
		(6:160-150)	235	9.80 %	164	21.33
100	(1:70-90)	(10:250-180)	220	3.60 %	132	16.5
		(9:195-160)	221	4.10 %	156	16.77
		(6:160-150)	223	4.90 %	159	17.18
100	(2:70-90)	(10:250-180)	222	4.50 %	139	15.9
		(9:195-160)	222	4.50 %	152	16.17
		(6:160-150)	224	5.40 %	160	16.31
100	(3:70-90)	(10:250-180)	220	3.60 %	154	16.7
		(9:195-160)	229	7.40 %	156	17.72
		(6:160-150)	230	7.80 %	160	19

As shown in Table III, in column task interrupt, (1:70-90) means M_1 is forced to suspend the task at $t = 70$ and recover at $t = 90$. In column due date, (10:250-180) indicates the due date of Ta_{10} is advanced to 180 from 250 (mid-value of the interval for convenience). $\overline{f_1}$ and $\overline{f_2}$ denote the average processing cycle and the average tardiness penalty when the program runs 20 times. $\bar{t}$ represents the average convergence time of running 20 times. With the analysis, it is true of an ever increasing value of the average process cycle, the deviation, the average tardiness penalty and the average convergence time over the change process from (10:250-180), (9:195-160) to (3:70-90). This kind of change explains that the due date variation for tasks with a less time have a greater impact on PCD performance. Meanwhile, on the basic of the same ΔT and due date variation, considering the task interrupt over (1:70-90), (2:70-90) to (3:70-90), there is a similar variation tendency of ever increasing for the average process cycle, the deviation, the average tardiness penalty and the average convergence time. This indicates that the designers with a well-distributed processing time have a much greater effect on PCD. Actually, these kinds of designers are usually regarded as much-needed and versatile person for an enterprise. In addition, in the process of ΔT changing from 50 to 100, the average convergence time gradually drops while the average process cycle, the deviation and the average tardiness penalty appear unstable mutations. This suggests that there is true of an optimal rescheduling period for each designer.

5. CONCLUSIONS

In summary, PCD scheduling problem has received increasing attention in the engineering field for its popularity in manufacturing and supply chain systems. The dynamics and multi-objective of PCD pose a great challenge for enterprises making quick response to unplanned perturbations. Thus, in this paper, a multiobjective fuzzy scheduling model and its algorithm was proposed. The analytical results indicated that the PCD cycle is shortened and the tardiness penalty is smaller by the method when accidents emerged. The main achievements of this paper are as following:

(1) Hybrid-driven scheduling, synthesizing advantages both of event-driven and lifecycle-driven, could not only deal with unexpected incidents but also kept the relative stability of the collaborative system.

(2) MODASA, based on bi-layer coding strategy, self-adaptive double point crossover and self-adaptive mutation, shown full advantages in computing speed and precision.

(3) On the premise of the same ΔT and task interrupt, the due date variation for tasks with a less time have a greater impact on PCD performance. While, the designers with a well-distributed processing time have a much greater effect on PCD in the context of the same ΔT and the due date. Moreover, there is true of an optimal rescheduling period for each designer because the average convergence time gradually drops when ΔT changes.

However, it is commonly believed that PCD scheduling is a complex problem. There are many factors influencing the efficiency of task scheduling, but in this article, only two objectives are studied, which are comparatively lack when applied in real situation. Thus, it is needed to be further expounded and studied for its factors and objectives. Moreover, there are many other algorithms to resolve the problem, so new attempts to resolution of PCD scheduling will be focused in the further study.

6. ACKNOWLEDGEMENTS

This research is sponsored by National Natural Science Foundation of China under Project number 71071173, 71301176. The authors also thank to Fundamental Research Funds for the

Central Universities (Project No.CDJZR12110004) and Specialized Research Fund for the Doctoral Program of Higher Education (Project 20130191120001) for partial support of this work. We are grateful for the constructive suggestions provided by the reviewers, which would improve the paper.

REFERENCES

[1] Zhang, X.-D.; Luo, L.; Yang, Y.; Li, Y.-Z.; Schlick, C. M.; Grandt, M. (2009). A simulation approach for evaluation and improvement of organisational planning in collaborative product development projects, *International Journal of Production Research*, Vol. 47, No. 13, 3471-3501, doi:10.1080/00207540802356770

[2] Salimi, R.; Motameni; H.; Omranpour, H. (2014). Task scheduling using NSGA II with fuzzy adaptive operators for computational grids, *Journal of Parallel and Distributed Computing*, Vol. 74, No. 5, 2333-2350, doi:10.1016/j.jpdc.2014.01.006

[3] Braha, D.; Bar-Yam, Y. (2007). The statistical mechanics of complex product development: Empirical and analytical results, *Management Science*, Vol. 53, No. 7, 1127-1145, doi:10.1287/mnsc.1060.0617

[4] Belhe, U.; Kusiak, A. (1997). Dynamic scheduling of design activities with resource constraints, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 27, No. 1, 105-111, doi:10.1109/3468.553229

[5] Luh, P. B.; Liu, F.; Moser, B. (1999). Scheduling of design projects with uncertain number of iterations, *European Journal of Operational Research*, Vol. 113, No. 3, 575-592, doi:10.1016/S0377-2217(98)00027-7

[6] Artigues, C.; Michelon, P.; Reusser, S. (2003). Insertion techniques for static and dynamic resource-constrained project scheduling, *European Journal of Operational Research*, Vol. 149, No. 2, 249-267, doi:10.1016/S0377-2217(02)00758-0

[7] Danilovic, M.; Browning, T. R. (2007). Managing complex product development projects with design structure matrices and domain mapping matrices, *International Journal of Project Management*, Vol. 25, No. 3, 300-314, doi:10.1016/j.ijproman.2006.11.003

[8] Pan, Z.-Y.; Huang, X.; Li, Y.-G. (2008). Product design process planning method based on synchronous mapping from directed graph to design structure matrix and polychromatic sets, *Computer Integrated Manufacturing Systems*, Vol. 2008, No. 7, 1274-1280

[9] Browning, T. R.; Yassine, A. A. (2010). Resource-constrained multi-project scheduling: Priority rule performance revisited, *International Journal of Production Economics*, Vol. 126, No. 2, 212-228, doi:10.1016/j.ijpe.2010.03.009

[10] Li, Y.-Z.; Zhang, X.-D.; Zhang, S.; Zhang, Z.-Q. (2012). Multi-mode optimal scheduling of product design project based on person-task-resource matching degree, *Industrial Engineering and Management* ,Vol. 17, No. 3,74-80

[11] Lapegue, T.; Bellenguez-Morineau, O.; Prot, D. (2013).A constraint-based approach for the shift design personnel task scheduling problem with equity, *Computers & Operations Research* ,Vol. 40, No. 10, 2450-2465, doi:10.1016/j.cor.2013.04.005

[12] Lam, F. S. C.; Lin, B. C.; Sriskandarajah, C.; Yan, H. (1999). Scheduling to minimize product design time using a genetic algorithm, *International Journal of Production Research*, Vol. 37, No. 6, 1369-1386, doi:10.1080/002075499191300

[13] Gomez-Gasquet, P.; Rodriguez-Rodriguez, R.; Franco, R. D.; Ortiz-Bas, A. (2012). A collaborative scheduling GA for products-packages service within extended selling chains environment, *Journal of Intelligent Manufacturing*, Vol. 23, No. 4, 1195-1205, doi:10.1007/s10845-010-0434-z

[14] Krishnamoorthy, M.; Ernst, A. T.; Baatar, D. (2012). Algorithms for large scale shift minimisation personnel task scheduling problems, *European Journal of Operational Research*, Vol. 219, No. 1, 34-48, doi:10.1016/j.ejor.2011.11.034

[15] Meier, C.; Yassine, A. A.; Browning, T. R. (2006). Design process sequencing with competent genetic algorithms, *Journal of Mechanical Design*, Vol. 129, No. 6, 566-585, doi:10.1115/1.2717224

[16] Kumar, G. M.; Haq, A. N. (2005). Hybrid genetic-ant colony algorithms for solving aggregate production plan, *Journal of Advanced Manufacturing Systems*, Vol. 4, No. 1, 103-111, doi:10.1142/S021968670500059X

[17] Jian, C. F; Wang, Y. (2014). Batch task scheduling-oriented optimization modelling and simulation in cloud manufacturing, *International Journal of Simulation Modelling*, Vol. 13, No. 1, 93-101, doi:10.2507/IJSIMM13(1)CO2

[18] Reddy, B. S. P; Rao, C. S. P. (2011). Flexible manufacturing systems modelling and performance evaluation using Automod, *International Journal of Simulation Modelling*, Vol. 10, No. 2, 78-90, doi:10.2507/IJSIMM10(2)3.176

[19] Balin, S. (2011). Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation, *Information Sciences*, Vol. 181, No. 17, 3551-3569, doi:10.1016/j.ins.2011.04.010

[20] Debevec, M.; Simic, M.; Herakovic, N. (2014). Virtual factory as an advanced approach for production process optimization, *International Journal of Simulation Modelling*, Vol. 13, No. 1, 66-78, doi:10.2507/IJSIMM13(1)6.260

[21] Udhayakumar, P.; Kumanan, S. (2010). Task scheduling of AGV in FMS using non-traditional optimization techniques, *International Journal of Simulation Modelling*, Vol. 9, No. 1, 28-39, doi:10.2507/IJSIMM09(1)3.139

[22] Li, W. D.; Ong, S. K.; Nee, A. Y. C. (2002). Hybrid genetic algorithm and simulated annealing approach for the optimization of process plans for prismatic parts, *International Journal of Production Research*, Vol. 40, No. 8, 1899-1922, doi:10.1080/00207540110119991

[23] Zhang, J.-B.; Chen, K. (2010). An adaptive ant colony algorithm for concurrent design task planning problem, *Journal of Computer-Aided Design & Computer Graphic*s, Vol. 22, No. 6, 1070-1074, doi:10.3724/sp.j.1089.2010.10878

[24] Yeh, W.-C.; Lai, P.-J.; Lee, W.-C.; Chuang, M.-C. (2014). Parallel-machine scheduling to minimize makespan with fuzzy processing times and learning effects, *Information Sciences*, Vol. 269, 142-158, doi:10.1016/j.ins.2013.10.023

[25] Liang, C.; Guo, J.; Yang, Y. (2011). Multi-objective hybrid genetic algorithm for quay crane dynamic assignment in berth allocation planning, *Journal of Intelligent Manufacturing*, Vol. 22, No. 3, 471-479, doi:10.1007/s10845-009-0304-8