

Hindawi Publishing Corporation
Scientific Programming
Volume 2016, Article ID 8264879, 9 pages
<http://dx.doi.org/10.1155/2016/8264879>



Research Article

A Recovery Model for Production Scheduling: Combination of Disruption Management and Internet of Things

Yang Jiang,¹ Qiulei Ding,² and Xuhui Wang²

¹School of Light Industry & Chemical Engineering, Dalian Polytechnic University, Dalian 116034, China

²School of Business Administration, Dongbei University of Finance and Economics, Dalian 116025, China

Correspondence should be addressed to Qiulei Ding; dingqiulei2016@163.com

Received 13 June 2016; Accepted 25 August 2016

Academic Editor: Chengyan Yue

Copyright © 2016 Yang Jiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is difficult to generate the new schedule effectively for minimizing the negative impact when an unanticipated disruption occurs after a subset of tasks has been finished in production scheduling. In such cases, continuing with the original schedule may not be optimal or feasible. Based on disruption management and Internet of things (IoT), this study designs a real-time status analyzer to identify the disruption and propose a recovery model to deal with the disruption. The computational result proves that our algorithm is competitive with the existing heuristics. Furthermore, due to the tradeoff between all participators (mainly including customers, managers of production enterprise, and workers) involved in production scheduling, our model is more effective than the total rescheduling and right-shift rescheduling.

1. Introduction

Providing customers with satisfactory service is the central concern for production enterprises. However, given the complexity of processing environments, random or unanticipated events, including anomaly status of processing environment and machine breakdown, constantly occur in production scheduling. As a result, the original schedule is likely to be suboptimal and may not even be feasible. In this study, we use the term “disruption” to describe the reasons that trigger a rescheduling process.

Research on rescheduling is extensive [1, 2]. We differentiate our model from the existing findings in which the new schedule will be evaluated. The objective of most rescheduling research is to find an optimal schedule in the new environment with respect to the original objective function. In our model, we consider both the original objective function and the deviation from the original schedule. The reason for considering the deviation is that once the original schedule is generated, many preparations have already been made, such as allocating resources, distributing raw materials, and fixing customer delivery dates. Hence, if any change to the original schedule has a negative impact on these preparations, then the new schedule has to minimize the deviation [3].

In this paper, we combine disruption management and IoT to deal with the disruption. IoT is to realize the information sharing of global things from any place, which is able to monitor the process of production scheduling and identify the disruption quickly. Furthermore, disruption management aims to dynamically revise the original schedule and obtain a new schedule that reflects the constraints and objectives of the evolved environment [4].

The rest of the paper is organized as follows. Section 2 provides a brief review of the related work. Section 3 designs a real-time status analyzer based on IoT to identify the disruption. Section 4 constructs a recovery model combining quantitative model and prospect theory to measure the deviation, and Section 5 proposes an improved ant colony optimization (IACO) to obtain the new schedule quickly. Section 6 shows the validity of the demonstrated model and IACO. Finally, Section 7 provides conclusions and directions for future research.

2. Related Work

Determining how to handle disruption is a significant issue in the decision-making process. Disruption management is an example of rescheduling in real time, which is necessary in

some cases because disruption is unanticipated and cannot be embedded in making the original schedule in advance. Through disruption management, a new schedule that minimizes the negative impact can be obtained.

Disruption management was first applied to the airline industry, in which flight disruptions often entail huge cost loss [5, 6]. The successful implementation of the idea led to a growing interest in applying disruption management to other fields. In the study of production scheduling, Lee and Yu [7] studied the parallel-machine scheduling problem under the disruptive environment by minimizing the sum of weighted completion times and provided a pseudopolynomial time algorithm to solve the problem. Tang and Zhang [8] investigated the disruption caused by machine breakdown and proposed the Lagrangian relaxation approach to deal with the disruption. Wang et al. [9] discussed parallel-machine scheduling problems with a deteriorating maintenance activity and provided a polynomial time algorithm to solve the total completion time minimization problem. Liu et al. [10] studied the disrupted single-machine scheduling problem. The deviation was measured by the completion times of jobs and quantum-inspired hybrid heuristics were presented. Khedlekar et al. [11] studied the flexible managerial decision policy for disruption in production system. They solved the disruption problem analytically to determine the production period before and after disruptions. Paul et al. [12] developed the recovery model and dynamic solution approach to deal with disruptions for production-inventory system. Sarker et al. [13] focused on production scheduling under uncertain disruption and adopted an improved memetic algorithm to solve the problem.

Another stream of literature relevant to our research is the recently emerging subject of behavioral operation management [14]. The operation contexts primarily investigated were supply chain management, newsvendor problem, forecasting, and decision-making, which is concerned with the theories of individual decision-making biases, social preferences, and cultural elements [15]. Su [16] proposed a decision framework of bounded rationality applied to the classic newsvendor model. Gino and Pisano [17] explored the theoretical and practical implications of incorporating behavioral and cognitive factors into models of operations. Chen et al. [18] presented a model of bounded rationality based on the quantal response equilibrium, in which players were not perfect optimizers and faced uncertainty in the actions of their opponents. Katsikopoulos and Gigerenzer [19] briefly discussed some research on the heuristics in behavioral operation management and showed how the study of quantitative models of heuristics can complement it. Chen and Zhao [20] developed a behavioral model with different mental weights on the underage and overage costs to characterize the perception bias of a retailer on a critical fractile.

In summary, identifying the disruption was likely to be neglected in previous literature because the researchers focused on obtaining the revising schedule quickly. In addition, the existing researches assumed that humans in production scheduling were perfectly rational and emphasized the optimization of material and financial resources. They

disregarded these facts that people facing a new situation often have different perceptions and the obtained solution may not be optimal or feasible. Accordingly, how to identify the disruption and measure the deviation still needs further study.

3. A Real-Time Status Analyzer Based on IoT

In production scheduling, the ambient condition subtly changes all the time. However, considering and adjusting it as a whole are worthy only when the current schedule does not work properly. Thus, a real-time status analyzer based on IoT is designed and developed to identify the disruption before adopting the recovery model.

3.1. System Architecture. The status data are sent to computer terminals (CTs) of the Internet via wireless sensor networks (WSNs) and satellite. The data processing center analyzes the data, and it is alerted when a disruption occurs. The system can achieve the following three functions.

(1) *Data Acquisition.* Jobs in the shop are frequently sensitive to temperature, humidity, and pressure. Thus, the status of processing environment needs to be monitored in real time. All data of the basic properties and environmental status of jobs are also sent to the radio frequency identification (RFID) reader via the WSNs. Finally, the aforementioned information is summarized and sent to the corresponding CTs.

(2) *Exception Analysis.* Based on the initial schedule and according to different characteristics of jobs, the corresponding thresholds for equipment status, shop temperature, humidity, and pressure are preset. Thus, the massive time-varying data can be monitored and analyzed in real time to determine whether any exception occurs.

(3) *Real-Time Alert.* When an exception occurs as revealed by the monitored data, the disruption occurs and the system gives out alerts immediately, thereby revising the original schedule and obtaining a new schedule that minimizes the negative impact.

3.2. Redundant Data Processing. RFID is prone to be absorbed and disrupted by the ambient condition, thereby resulting in any omission or deficiency in the reading process of the reader. This study aims to avoid such problems by proposing increasing the number of readers or the read-write frequency, thereby ensuring data integrity. However, this proposal can inevitably produce a significant amount of redundant data, increasing the burden to the RFID data transmission and storage. Thus, the system efficiency is greatly reduced. Therefore, an effective way to solve problems of redundant data processing must be determined to screen all data and to reduce the burden on the system.

Basic sorted-neighborhood method (SNM) is an efficient algorithm to cope with redundant data. However, it is difficult to control the size of smoothing window. This study presents a modified SNM algorithm with specific procedures as follows.

Step 1. The data are sorted according to the timestamp.

Step 2. In the data comparison process, a smoothing window featured with adaptive adjustment is used, and the window is resized as desired.

Step 3. The first data entry in the current window is compared with the next, and whether it is repeated is determined. If it is repeated, Step 4 follows; otherwise, Step 5 is next.

Step 4. When the repeated entry is found, the entry with the oldest timestamp is deleted and that with the latest timestamp is retained.

Step 5. Whether the first entry in the window is the last is determined. If it is the last, Step 6 follows; otherwise, Step 3 is next.

Step 6. Whether any unprocessed data exists is determined. If so, Step 1 follows; otherwise, the entire workflow ends.

4. Recovery Model for Production Scheduling

Considering its wide application in production and manufacturing industries, the Job-Shop Scheduling Problem (JSP) is chosen as an example to construct the recovery model.

4.1. Construction of the Model for Original Schedule

4.1.1. Problem Definition. The problem is confined to the following conditions. Given n jobs, each job has m operations and must be processed on m machines. The objective is to find a schedule of minimal time to complete all jobs, where

- (1) the machine can only process one job at a time,
- (2) every job is available for processing at time 0,
- (3) once processing is initiated, the operation must be completed on the machine without interruption.

4.1.2. Notations

n : the number of jobs.

m : the number of machines.

V : the set of jobs, $V = \{v_1, v_2, \dots, v_n\}$.

U : the set of machines, $U = \{u_1, u_2, \dots, u_m\}$.

c_{ik} : the completion time of job i in machine k .

p_{ik} : the processing time of job i in machine k .

d_i : the due date of v_i .

M : a large positive number.

$$a_{ihk} = \begin{cases} 1, & v_i \text{ is processed in } u_h \text{ before } u_k \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

$$x_{ijk} = \begin{cases} 1, & v_i \text{ is processed before } v_j \text{ in } u_k \\ 0, & \text{otherwise.} \end{cases}$$

4.1.3. Mathematical Model. The JSP model is constructed as follows:

$$\min \max_{1 \leq k \leq m} \left\{ \max_{1 \leq i \leq n} c_{ik} \right\} \quad (2)$$

$$c_{jk} - p_{jk} + M(1 - x_{ijk}) \geq c_{ik} \quad (3)$$

$$i, j = 1, 2, \dots, n; k = 1, 2, \dots, m$$

$$c_{ik} - p_{ik} + M(1 - a_{ihk}) \geq c_{ih} \quad (4)$$

$$i = 1, 2, \dots, n; h, k = 1, 2, \dots, m$$

$$d_i \leq \max_{1 \leq k \leq m} c_{ik} \quad i = 1, 2, \dots, n. \quad (5)$$

In this model, the objective function (2) aims to minimize the makespan, which is the minimal time to complete all jobs. Formulas (3) and (4) are the technological and processing constraints. Formula (5) ensures that all jobs are completed before their due date.

4.2. Construction of the Recovery Model. When the optimal schedule is executed, the disruption occurs and is identified by the real-time status analyzer. Consequently, we need to construct the recovery model and obtain a new schedule.

Machine breakdown is the typical disruption in production scheduling. In this paper, the recovery model is illustrated by the example of machine breakdown.

4.2.1. Problem Hypothesis

- (1) The original schedule is known.
- (2) The time when disruption ends is regarded as 0.
- (3) The processing job when disruption occurs must be reprocessed after the disruption ends. In other words, jobs are uninterruptible.

4.2.2. Notations

w : the number of unprocessed jobs.

V' : a subset of unprocessed jobs, $V' = \{v_1, v_2, \dots, v_w\}$.

A_{ij}^1 : the set of operations processed before operation j in the original schedule for machine i .

A_{ij}^2 : the set of operations processed after operation j in the recovery schedule for machine i .

A_{ij} : the intersection of A_{ij}^1 and A_{ij}^2 , $A_{ij} = A_{ij}^1 \cap A_{ij}^2$.

$\overline{\overline{A_{ij}}}$: the cardinality of subset A_{ij} .

c_i^0 : completion time of v_i in the original schedule.

c_i : completion time of v_i in the recovery schedule.

f^0 : the makespan in the original schedule.

f : the makespan in the recovery schedule.

g : the sequence deviation between the original schedule and the recovery schedule.

μ_1^i : dissatisfaction degree of customer i for completion time deviation.

μ_2 : dissatisfaction degree of managers for operational cost deviation.

μ_3 : dissatisfaction degree of workers for sequence deviation.

α_1, β_1 : parameters related to gains and losses for customers.

α_2, β_2 : parameters related to gains and losses for managers.

α_3, β_3 : parameters related to gains and losses for workers.

$\lambda_1, \lambda_2, \lambda_3$: loss aversion coefficient for customers, managers, and workers, respectively.

The other notations are the same as the ones in previous sections.

4.2.3. Impact of Disruption. The general process of one schedule is as follows: (i) managers of production enterprises generate an optimal schedule according to the requirements of customers, (ii) workers execute the schedule to process jobs, and then (iii) customers receive their products within the required time. The impact of disruption on the above three participators (i.e., managers, workers, and customers) is analyzed below.

(1) *Managers.* After disruption occurs, the processing sequence of unprocessed jobs will change, leading to the increase of makespan. The makespan is relevant to the operational cost, which is the central concern of managers. Accordingly, minimizing the makespan of unprocessed jobs is the key objective of managers when generating the recovery schedule.

(2) *Workers.* The processing sequence of unprocessed jobs will be changed after disruption occurs. Considering that many preparations have been made in advance based on the original schedule, workers have to reallocate raw materials and tools, which will lead to the increase of extra workload. Therefore, the recovery schedule should minimize the sequence deviation to reduce the negative impact on workers.

(3) *Customers.* Once the environment is interrupted by disruption, a chain reaction will be generated to affect the completion time of unprocessed jobs, and a number of customers will not be served within the required time. Thus, the recovery schedule should reduce the deviation of completion time to improve the satisfaction and loyalty of customers.

4.2.4. Function of Measuring Deviation. Prospect theory [21] is one of the most important decision-making theories. It can perfectly describe the decision-making based on bounded rationality under the uncertainty condition. Hence, on the basis of prospect theory, this paper measures the deviation of participators [22].

(1) *Measuring the Deviation of Customers.* The value function of customer i can be described as follows:

$$V_1^i(x) = \begin{cases} x^{\alpha_1}, & x \geq 0 \\ -\lambda_1(-x)^{\beta_1}, & x < 0 \end{cases} \quad i = 1, 2, \dots, w, \quad (6)$$

where the reference point is c_i^0 ($c_i^0 = \max_{1 \leq k \leq m} c_{ik}$). $c_i > c_i^0$ means losses ($x < 0$) for customer i ; otherwise, it means gains ($x \geq 0$) for customer i .

The unsatisfied membership function of customer i for completion time deviation can be given as follows:

$$\mu_1^i(c_i) = \begin{cases} 1, & c_i \geq R_{1i} \\ \lambda_1(c_i - c_i^0)^{\beta_1}, & c_i^0 \leq c_i < R_{1i} \\ 0, & 0 \leq c_i < c_i^0 \end{cases} \quad (7)$$

$i = 1, 2, \dots, w,$

where $R_{1i} = [c_i^0 + (1/\lambda_1)^{1/\beta_1}]$ ($i = 1, 2, \dots, w$).

(2) *Measuring the Deviation of Managers.* The value function of managers can be constructed as follows:

$$V_2(x) = \begin{cases} x^{\alpha_2}, & x \geq 0 \\ -\lambda_2(-x)^{\beta_2}, & x < 0, \end{cases} \quad (8)$$

where the reference point is f^0 . $f > f^0$ means losses ($x < 0$) for managers; otherwise, it means gains ($x \geq 0$) for managers.

The unsatisfied membership function of managers for operational cost deviation can be given as follows:

$$\mu_2(f) = \begin{cases} 1, & f \geq R_2 \\ \lambda_2(f - f^0)^{\beta_2}, & f^0 \leq f < R_2 \\ 0, & 0 \leq f < f^0, \end{cases} \quad (9)$$

where $R_2 = f^0 + (1/\lambda_2)^{1/\beta_2}$.

(3) *Measuring the Deviation of Workers.* The value function of workers can be constructed as follows:

$$V_3(x) = -\lambda_3(-x)^{\beta_3}, \quad x < 0, \quad (10)$$

where the reference point is 0 because the sequence deviation does not exist in the original schedule. In recovery schedule, g is equal to $\sum_{i=1}^m \sum_{j=1}^w \overline{A}_{ij}$. $g > 0$ means losses ($x < 0$) for workers.

The unsatisfied membership function of workers for sequence deviation can be given as follows:

$$\mu_3(g) = \begin{cases} 1, & g \geq R_3 \\ \lambda_3 g^{\beta_3}, & 0 \leq g < R_3, \end{cases} \quad (11)$$

where $R_3 = (1/\lambda_3)^{1/\beta_3}$.

4.2.5. *Recovery Model.* On the basis of the above functions of measuring deviation, the objective function of the recovery model is constructed through lexicographic multiple goal programming as follows:

$$\begin{aligned} \min \quad & \text{Lex} = P_1 : \sum_{i=1}^w \mu_1^i (c_i) \\ & P_2 : \mu_2 (f) \\ & P_3 : \mu_3 (g). \end{aligned} \quad (12)$$

Formula (12) aims to minimize the sum of the unsatisfaction degrees of customers, managers, and workers. P_1 , P_2 , and P_3 represent the different priorities, which can be flexibly adjusted in the practical application.

5. IACO for the Recovery Model

The proposed model is NP-hard, which has an optimal solution that is difficult to obtain. Ant colony optimization (ACO) is an evolutionary computation technique developed by Colorni et al. [23]. Compared with other heuristics, ACO has the characteristics of positive feedback, distributed computation, and constructive greedy heuristic. In particular, the ACO has been shown to be an efficient algorithm for solving NP-hard problems [24–26]. However, the algorithm still has the weaknesses of local optimization and low search speed. In this study, we illustrate IACO to solve the recovery model.

5.1. Introduction of Adjusting the Pheromone Trail

- (1) In ACO, pheromone trails left by ants do not always show the evolutionary direction, and the pheromone deviating from the optimal solution has the probability of increasing, which leads to premature convergence. This study proposes an approach to enhance the global search capability of ACO by adjusting the pheromone trail adaptively with the evolutionary process.
- (2) The updated pheromone trail may reach the maximum value τ_{\max} or minimum value τ_{\min} after every search process is completed. τ_{\max} leads to premature convergence, and τ_{\min} increases the computational time. Thus, IACO limits the pheromone trail τ_{ij} in the interval $(\tau_{\min}, \tau_{\max})$. After the pheromone trails are updated, τ_{ij} is replaced by τ_{\max} when $\tau_{ij} > \tau_{\max}$ or by $(\tau_{\min} + \tau_{\max})/2$ when $\tau_{ij} < \tau_{\min}$.
- (3) ACO has difficulty obtaining the optimal solution because the trail persistence ρ is fixed. The smaller the value of ρ is, the better the global search capability will be. However, the computational time will increase gradually. On the contrary, increasing ρ can improve the computational time, but the algorithm can easily be trapped in the local optimal solution. Therefore, IACO adopts a dynamic ρ rather than a fixed value.

5.2. *Approach of the Crossover and Mutation.* The crossover and mutation operations can increase the variance of the population and search the solution space completely in genetic algorithm (GA). Thus, these two operations are adopted in IACO to avoid premature convergence.

(1) *Crossover Operation.* In the paper, a schedule s is represented by the following encoding structure:

$$s = \left\{ \underbrace{\text{solu}_1, \dots, \text{solu}_m}_{s_1}, \underbrace{\text{solu}_{m+1}, \dots, \text{solu}_{2m}, \dots}_{s_2}, \dots, \underbrace{\text{solu}_{(w-1)m+1}, \dots, \text{solu}_{wm}}_{s_w} \right\}, \quad (13)$$

where s_i represents the sequence of operations of job $i = 1, 2, \dots, w$. Each solu in s_i is an unrepeated integer number in the interval $[1, w]$.

When the search becomes trapped in the local optimal solution, crossover operation is conducted as follows:

- (i) Assume that encoding of schedule a is $\{a_1, a_2, \dots, a_w\}$ and encoding of schedule b is $\{b_1, b_2, \dots, b_w\}$.
- (ii) Randomly generate an integer number j in the interval $[1, w]$. Then, exchange a_j and b_j , and generate two new encodings, that is, $\{a_1, a_2, \dots, b_j, \dots, a_w\}$ and $\{b_1, b_2, \dots, a_j, \dots, b_w\}$.
- (iii) Update the encoding of optimal solution.

(2) *Mutation Operation.* Mutation plays a significant part in improving the diversity of the population. Similar to GA, IACO is designed to avoid premature convergence and obtain a better solution by significantly decreasing pheromone trails in any path of the local optimization routing. These paths will be selected by small random probabilities because the distribution of the pheromone trails will be destroyed by too many mutation operations.

5.3. *Combination with Other Heuristics.* ACO has the characteristic of strong coupling, which makes it easy to combine with other heuristics. Exchange mechanism is originally applied in the vehicle routing problem with time windows. It can obtain a new solution from the current solution by exchanging the nodes. We demonstrate a neighborhood exchange mechanism to significantly improve the convergence speed of ACO.

(1) *Operators of the Neighborhood Exchange Mechanism.* Two neighborhood operators (among those listed below) are included in this study.

- (i) *Random Swaps of Subsequences.* This operator randomly selects i and j with $i \neq j$ and swaps s_i and s_j in the solution sequence. Figure 1 shows an example of this operator.
- (ii) *Random Insertions of Subsequences.* This operator randomly selects i and j with $i \neq j$ and puts s_j in front of s_i . An example is shown in Figure 2.

TABLE 1: Computational results of solving different instances.

Algorithm	TA05	TA11	TA20	TA37	TA43	DMU12	DMU26	DMU45	DMU61	DMU77
<i>i</i> -TSAB	1224	1361	1351	1778	1859	3519	4679	3321	5294	6930
GES	1224	1357	1348	1779	1870	3518	4667	3273	5293	7006
AlgFix	1224	1358	1348	1784	1869	3522	4688	3273	5310	6949
IACO	1224	1357	1348	1779	1858	3520	4665	3275	5277	6908

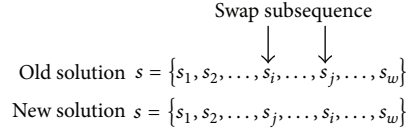


FIGURE 1: Random swaps of subsequences.

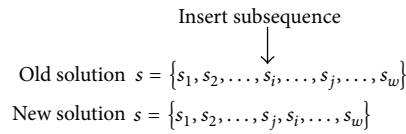


FIGURE 2: Random insertions of subsequences.

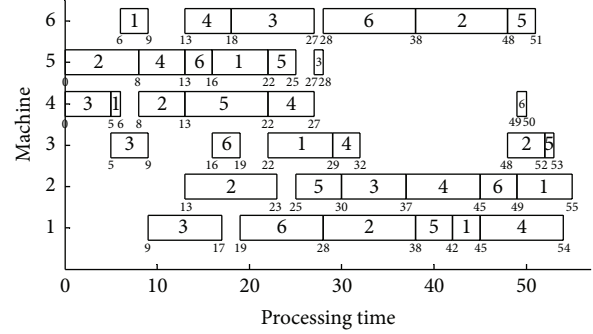


FIGURE 3: Original schedule.

(2) Rules of the Neighborhood Exchange Mechanism

- (i) To avoid generating too many unavailable solutions, the operations in the same machine can be exchanged.
- (ii) To avoid destroying the distribution of the previous pheromone trails, one neighborhood operator is chosen randomly from the above operators and applied once to the current solution.

6. Numerical Examples

In this section, we illustrate computational experiments to validate the effectiveness of IACO and recovery model presented in the previous sections.

6.1. Computational Experiment for IACO

(1) *Computational Results.* The IACO is tested using the classical sets of JSP, which are TA [27] and DMU [28]. For each instance, the proposed algorithm is independently executed 10 times to compute the average value. We then conduct a performance comparison between IACO and other heuristics, including *i*-TSAB [29], GES [30], and AlgFix [31]. The results are illustrated in Table 1.

(2) *Comparison among Different Heuristics.* The comparison obtained from the above results can be summarized as follows:

- (i) Compared with *i*-TSAB, IACO obtains better or close solutions in 8 out of 10 problems (80%).
- (ii) Compared with GES, IACO obtains better or close solutions in 8 out of 10 problems (80%).

- (iii) Compared with AlgFix, IACO obtains better or close solutions in 9 out of 10 problems (90%).

The comparison indicates that the IACO is competitive with the existing heuristics. Moreover, it can improve the best solutions known for a number of examples, especially for large size problems, such as DMU61 and DMU77. It is noted that those parameters affect the overall performance of IACO, so the solution obtained by our algorithm would be improved further if every parameter is set more reasonably.

6.2. Computational Experiment for the Recovery Model

(1) *Case Description.* In the given job-shop, $n = 6$ and $m = 6$. The parameters of each job are shown in Table 2. Figure 3 illustrates the original schedule where the makespan is 55 h.

(2) *Computational Results.* After 10 h, the disruption suddenly occurs and the duration time is 3 h. The remaining jobs, including the processing job when disruption occurs, must be reprocessed after the disruption ends.

Following Tversky and Kahneman [32], we set $\beta = 0.88$ and $\lambda = 2.25$. By adopting our model, total rescheduling, and right-shift rescheduling [33], we obtain the results shown in Figures 4, 5, and 6, respectively. Table 3 shows the results from different approaches.

(3) *Comparison among Different Approaches.* The results obtained from Table 3 can be summarized as follows:

- (i) From the aspect of the deviation of customers, the result of our model is much better than the results of the other two approaches. In other words, our model plays an obvious role in reducing the unsatisfaction degree of customers.

TABLE 2: Parameters of each job.

	Job 1	Job 2	Job 3	Job 4	Job 5	Job 6
Operation 1	M4:T1	M5:T8	M4:T5	M5:T5	M4:T9	M5:T3
Operation 2	M6:T3	M4:T5	M3:T4	M6:T5	M5:T3	M3:T3
Operation 3	M5:T6	M2:T10	M1:T8	M4:T5	M2:T5	M1:T9
Operation 4	M3:T7	M1:T10	M6:T9	M3:T3	M1:T4	M6:T10
Operation 5	M1:T3	M6:T10	M5:T1	M2:T8	M6:T3	M2:T4
Operation 6	M2:T6	M3:T4	M2:T7	M1:T9	M3:T1	M4:T1
Due date	60 h	56 h	60 h	65 h	55 h	50 h

Note: $M_i:T_j$ means that the process time of the job's operation is j hours in machine i .

TABLE 3: Results from different approaches.

	Customers' deviation	Managers' deviation	Workers' deviation
Our model	2	1	5
Total rescheduling	3	1	4
Right-shift rescheduling	6	1	0

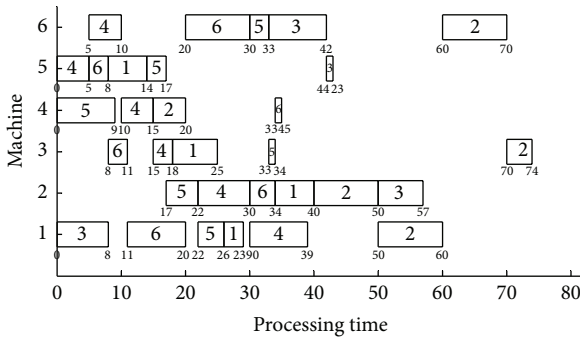


FIGURE 4: Recovery schedule of our model.

- (ii) From the aspect of the deviation of managers, our model, total rescheduling, and right-shift rescheduling obtain the same result. Therefore, our model is not worse than the other two approaches in reducing operational cost.
- (iii) From the aspect of the deviation of workers, right-shift rescheduling obtains the best result based on the original schedule. The results of our model and total rescheduling are relatively poor.

In summary, compared with total rescheduling and right-shift rescheduling, our model increases the deviation of workers. By contrast, our model decreases the deviation of customers and thus significantly improves customer satisfaction. Furthermore, the production enterprise is subject to improving the loyalty of customers to expand the influence and attract more new customers. Such improvement helps enhance the potential profit and promote the development of enterprise in the long run. Therefore, our model is more reasonable and valid than the other two approaches.

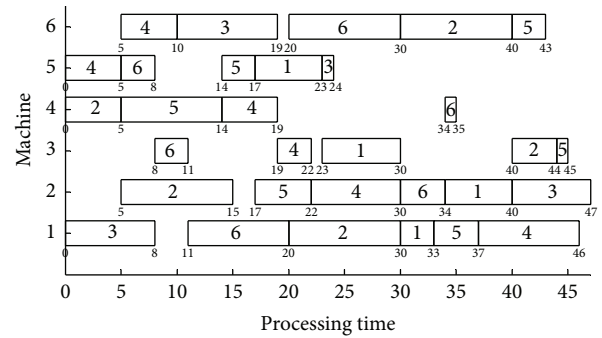


FIGURE 5: Recovery schedule of total rescheduling.

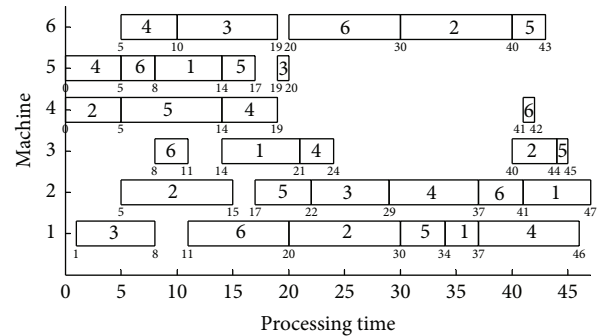


FIGURE 6: Recovery schedule of right-shift rescheduling.

7. Conclusions

This study designs a real-time status analyzer to identify the disruption and provides a recovery model to measure the deviation in production scheduling by combining disruption management with IoT. It aims to effectively identify the disruption and obtain a new schedule to minimize the negative impact on the participators (customers, managers, and workers). Meanwhile, IACO is presented to solve the above model.

To validate the effectiveness of our model in handling the disruption, we use an example to test different approaches. The computational result proves that our model is competitive with existing rescheduling and can obtain a relatively satisfactory schedule.

A limitation of this research is that we conducted the computational experiment for the recovery model by adopting the values of β and λ provided by Tversky and Kahneman [32]. Further research will concentrate on deriving the actual values of those parameters in production scheduling.

Competing Interests

The authors declare that they have no competing interests.

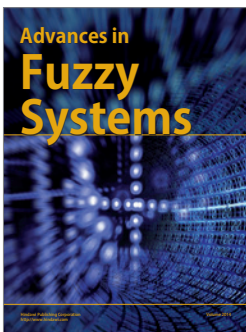
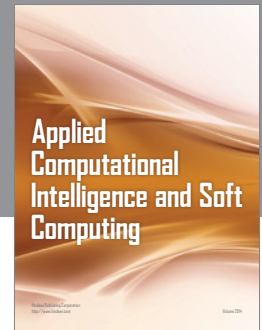
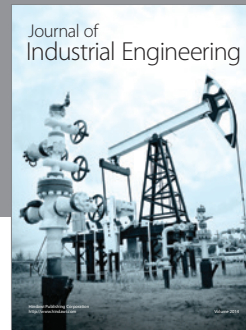
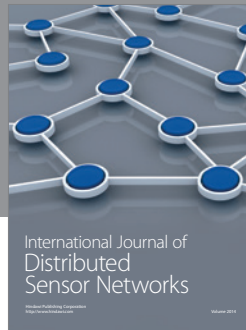
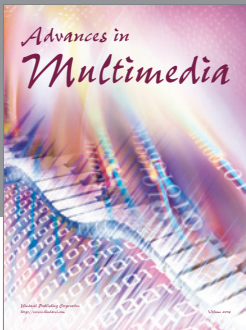
Acknowledgments

This work has been partially supported by the grants from the National Natural Science Foundation of China (nos. 71301020, 71272050), Program for Liaoning Excellent Talents in University (no. WJQ2014030), and Program for Liaoning Scientific Research in University (no. L2014216).

References

- [1] H. Aytug, M. A. Lawley, K. McKay, S. Mohan, and R. Uzsoy, "Executing production schedules in the face of uncertainties: a review and some future directions," *European Journal of Operational Research*, vol. 161, no. 1, pp. 86–110, 2005.
- [2] M. Z. Liu, H. Shan, Z. Q. Jiang, M. Ge, J. Hu, and M. Zhang, "Dynamic rescheduling optimization of job-shop under uncertain conditions," *Journal of Mechanical Engineering*, vol. 45, no. 10, pp. 137–142, 2009.
- [3] X. Qi, J. F. Bard, and G. Yu, "Disruption management for machine scheduling: the case of SPT schedules," *International Journal of Production Economics*, vol. 103, no. 1, pp. 166–184, 2006.
- [4] G. Yu and X. Qi, *Disruption Management: Framework, Models and Applications*, World Scientific Publishing Company Incorporated, 2004.
- [5] M. Løve, K. R. Sørensen, J. Larsen, and J. Clausen, "Disruption management for an airline—rescheduling of aircraft," in *Applications of Evolutionary Computing*, S. Cagnoni, J. Gottlieb, E. Hart, M. Middendorf, and G. R. Raidl, Eds., vol. 2279 of *Lecture Notes in Computer Science*, pp. 315–324, Springer, Berlin, Germany, 2002.
- [6] G. Yu, M. Argüello, G. Song, S. M. McCowan, and A. White, "A new era for crew recovery at continental airlines," *Interfaces*, vol. 33, no. 1, pp. 5–22, 2003.
- [7] C.-Y. Lee and G. Yu, "Parallel-machine scheduling under potential disruption," *Optimization Letters*, vol. 2, no. 1, pp. 27–37, 2008.
- [8] L. Tang and Y. Zhang, "Parallel machine scheduling under the disruption of machine breakdown," *Industrial & Engineering Chemistry Research*, vol. 48, no. 14, pp. 6660–6667, 2009.
- [9] J.-J. Wang, J.-B. Wang, and F. Liu, "Parallel machines scheduling with a deteriorating maintenance activity," *Journal of the Operational Research Society*, vol. 62, no. 10, pp. 1898–1902, 2011.
- [10] F. Liu, J.-J. Wang, and D.-L. Yang, "Solving single machine scheduling under disruption with discounted costs by quantum-inspired hybrid heuristics," *Journal of Manufacturing Systems*, vol. 32, no. 4, pp. 715–723, 2013.
- [11] U. K. Khedlekar, D. Shukla, and R. P. S. Chandel, "Computational study for disrupted production system with time dependent demand," *Journal of Scientific & Industrial Research*, vol. 73, no. 5, pp. 294–301, 2014.
- [12] S. K. Paul, R. Sarker, and D. Essam, "Managing disruption in an imperfect production–inventory system," *Computers & Industrial Engineering*, vol. 84, pp. 101–112, 2015.
- [13] R. Sarker, D. Essam, S. M. K. Hasan, and A. N. M. Karim, "Managing risk in production scheduling under uncertain disruption," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 30, no. 3, pp. 289–299, 2016.
- [14] E. Bendoly, K. Donohue, and K. L. Schultz, "Behavior in operations management: assessing recent findings and revisiting old assumptions," *Journal of Operations Management*, vol. 24, no. 6, pp. 737–752, 2006.
- [15] C. H. Loch and Y. Wu, *Behavioral Operations Management*, Now Publishers Inc, 2007.
- [16] X. Su, "Bounded rationality in newsvendor models," *Manufacturing & Service Operations Management*, vol. 10, no. 4, pp. 566–589, 2008.
- [17] F. Gino and G. Pisano, "Toward a theory of behavioral operations," *Manufacturing & Service Operations Management*, vol. 10, no. 4, pp. 676–691, 2008.
- [18] Y. Chen, X. Su, and X. Zhao, "Modeling bounded rationality in capacity allocation games with the quantal response equilibrium," *Management Science*, vol. 58, no. 10, pp. 1952–1962, 2012.
- [19] K. V. Katsikopoulos and G. Gigerenzer, "Behavioral operations management: a blind spot and a research program," *Journal of Supply Chain Management*, vol. 49, no. 1, pp. 3–7, 2013.
- [20] Y. Chen and X. Zhao, "Decision bias in capacity allocation games with uncertain demand," *Production and Operations Management*, vol. 24, no. 4, pp. 634–646, 2015.
- [21] D. Kahneman and A. Tversky, "Prospect theory: an analysis of decision under risk," *Econometrica*, vol. 47, no. 2, pp. 263–292, 1979.
- [22] Q. L. Ding, X. P. Hu, and Y. Jiang, "A model of disruption management based on prospect theory in logistic distribution," *Journal of Management Sciences in China*, vol. 17, no. 11, pp. 1–9, 2014.
- [23] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proceedings of the European Conference on Artificial Life*, pp. 134–142, Paris, France, 1991.
- [24] S. R. Balseiro, I. Loiseau, and J. Ramonet, "An Ant Colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows," *Computers & Operations Research*, vol. 38, no. 6, pp. 954–966, 2011.
- [25] R.-H. Huang, C.-L. Yang, and W.-C. Cheng, "Flexible job shop scheduling with due window—a two-pheromone ant colony approach," *International Journal of Production Economics*, vol. 141, no. 2, pp. 685–697, 2013.
- [26] M. Faisal, H. Mathkour, and M. Alsulaiman, "AntStar: enhancing optimization problems by integrating an ant system and A* algorithm," *Scientific Programming*, vol. 2016, Article ID 5136327, 12 pages, 2016.
- [27] É. D. Taillard, "Parallel taboo search techniques for the job shop scheduling problem," *INFORMS Journal on Computing*, vol. 6, no. 2, pp. 108–117, 1994.
- [28] E. Demirkol, S. Mehta, and R. Uzsoy, "A computational study of shifting bottleneck procedures for shop scheduling problems," *Journal of Heuristics*, vol. 3, no. 2, pp. 111–137, 1997.
- [29] E. Nowicki and C. Smutnicki, "An advanced tabu search algorithm for the job shop problem," *Journal of Scheduling*, vol. 8, no. 2, pp. 145–159, 2005.
- [30] P. M. Pardalos and O. V. Shylo, "An algorithm for the job shop scheduling problem based on global equilibrium search

- techniques,” *Computational Management Science*, vol. 3, no. 4, pp. 331–348, 2006.
- [31] P. M. Pardalos, O. V. Shylo, and A. Vazacopoulos, “Solving job shop scheduling problems utilizing the properties of backbone and ‘big valley,’” *Computational Optimization and Applications*, vol. 47, no. 1, pp. 61–76, 2010.
- [32] A. Tversky and D. Kahneman, “Advances in prospect theory: cumulative representation of uncertainty,” *Journal of Risk and Uncertainty*, vol. 5, no. 4, pp. 297–323, 1992.
- [33] R. J. Abumaizar and J. A. Svestka, “Rescheduling job shops under random disruptions,” *International Journal of Production Research*, vol. 35, no. 7, pp. 2065–2082, 1997.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

