


On the Robust and Stable Flowshop Scheduling Under Stochastic and Dynamic Disruptions

Feng Liu, Shengbin Wang, Yuan Hong, *Member, IEEE*, and Xiaohang Yue 

Abstract— In this paper, we consider a permutation flowshop scheduling problem with the total flow time as the schedule performance measure. A proactive–reactive approach is designed to simultaneously deal with stochastic disruptions (e.g., machine breakdowns) and dynamic events (e.g., newly arriving jobs and delay in job availability). In the proactive stage, the stochastic machine breakdown is hedged against the construction of a robust and stable baseline schedule. This schedule is either optimized by incorporating uncertainty into two surrogate measures or obtained by simulation. Robustness is measured by the expected schedule performance, while stability is measured by the aggregation of dissatisfactions of manager, shopfloor operator, and customers using the prospect theory. In the reactive stage, we assume that the stochastic and dynamic disruptions concurrently occur. Unlike the simple right-shifting method, a more effective rescheduling approach is proposed to balance the realized schedule performance with stability. A common issue in these two stages is the conflict between objectives. Thus, we propose a hybridization strategy that successfully enhances the classic Non-dominated Sorting Genetic Algorithm (NSGA-II) and the hybridized algorithm outperforms NSGA-II, multiobjective evolutionary algorithm based on decomposition, and multiobjective memetic algorithms designed for deterministic scheduling problems. Finally, extensive computational studies on the Taillard flowshop benchmark instances are conducted to illustrate the effectiveness of the proposed proactive–reactive approach and the algorithm hybridization strategy.

Index Terms—Evolutionary multiobjective optimization (EMO), flowshop, robustness, stability, stochastic and dynamic disruptions.

I. INTRODUCTION

AS ONE of the most imperative processes in the manufacturing and service systems, scheduling centers on allocating limited resources (e.g., machines or tools) to a set of tasks (e.g., jobs or operations) during a visible planning horizon, such

that one or more objectives with respect to tasks' completion times are optimized [1]. The output of scheduling is a baseline schedule that specifies the time/machine/operation assignments. Among the existing scheduling problems in various machine environments, the one in the flowshop environment has attracted an enormous attention from both academia and practice due to its challenging theoretical complexities and wide applications in assembling/manufacturing industries as well as medical operations [2], [6], [35]–[37].

The classic flowshop scheduling problems in existing literature mainly focus on generating a baseline schedule in a static and deterministic environment [4]. However, in practice, there also exist flowshop scheduling problems that involve stochastic and dynamic disruptions [7]–[9]. The scheduling process of the surgical operating theaters is a good example of such flowshop scheduling problem with stochastic and dynamic disruptions [2], [6], [7], [35], [36]. When hospitals have to undergo the economic pressure or resource shortages, the efficiency of the surgical suite utilization is of great importance due to a high turnover rate of the medical equipment and materials and the patients' unwillingness of waiting a long time for the surgical operations. In such scheduling process, the medical devices are prone to break down stochastically, while the arrivals of emergency patients can be considered as dynamic events. Hospital managers must consider these stochastic and dynamic scenarios when they generate baseline schedules for the operating theaters and/or reschedule the subsequent operations according to the new or emergent patients' arrivals [35]. The overall goals of proactive–reactive scheduling and rescheduling include, but are not limited to, ensuring the patient safety and the optimal patient outcome, increasing the utilization of staff and equipment, reducing delays, and enhancing the overall staff, patients, and surgeon/physicians' satisfaction [35], [36]. Another example of flowshop scheduling with stochastic and dynamic disruptions is the scheduling of *seru*—a cellular assembly system in the electronics industry [37]. Dynamic and high-costs market conditions forced many electronic manufacturers to adopt the *seru* production system, which prioritizes responsiveness to the dynamic disruptions over cost reduction in setting a firm's operations strategy [37]. Since the baseline schedules are often infeasible due to these disruptions, the methods used in the classical flowshop scheduling problems must be adjusted in order to solve the above-mentioned practical problems. Other similar examples can be found in steelmaking-continuous casting [39], TV fabrication, air conditioner manufacturers [41], petrochemical industry [43], etc. However, the results in literature considering

Manuscript received August 22, 2016; revised February 21, 2017; accepted June 1, 2017. Date of publication June 20, 2017; date of current version October 18, 2017. This work was supported by the National Natural Science Foundation of China under Grant 71502026 and Grant 71501032, and by the China Postdoctoral Science Foundation under Grant 2016M590228. Review of this manuscript was arranged by Department Editor B. Jiang. (*Corresponding author: Xiaohang Yue.*)

F. Liu is with the Dongbei University of Finance and Economics, Dalian 116023, China (e-mail: liufengapallo@163.com).

S. Wang is with North Carolina Agricultural and Technical State University, Greensboro, NC 27411 USA (e-mail: swang@ncat.edu).

Y. Hong is with the Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: yuan.hong@iit.edu).

X. Yue is with the University of Wisconsin–Milwaukee, Milwaukee, WI 53201 USA (e-mail: xyue@uwm.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TEM.2017.2712611

stochastic and dynamic disruptions in the flowshop scheduling are very limited [2], [5], [6], [35]–[42].

To address this deficiency, we consider a flowshop scheduling problem with the stochastic and dynamic disruptions in both the planning and the execution stages. We adopt a proactive–reactive approach, that is, we first generate a baseline schedule under uncertainties, and then update it accordingly after a disruption occurs. In the problem setting, robustness and stability are considered simultaneously. Robustness refers to the insensitiveness of the schedule performance to the variation of the system parameters, while stability is used to capture the dissatisfactions of the system participants including the production manager, the shopfloor operator, and the customers. We consider human behaviors because they constitute a core part of the functioning and performance of a manufacturing or service system [9]–[11]. The contribution of this paper can be summarized as follows. We propose a proactive–reactive approach to optimize the robustness and stability in a Pareto optimization manner, against stochastic machine breakdown and unexpected job-related events (dynamic new job arrival and job availability delay). We fill the gap in the flowshop scheduling field by incorporating the system participants’ dissatisfactions. To the best of our knowledge, the research work that directly models human dissatisfactions and behaviors in the scheduling field is very limited ([11] is the only example we have found).

The remainder of this paper is organized as follows. Section II reviews the production scheduling literature that explicitly considers robustness and/or stability. In Section III, the research problem is formally defined and formulated. In Section IV, we design the evolutionary multiobjective optimization (EMO)-based algorithms to tackle the conflicts between the objectives in the proactive and reactive stages. Extensive computational experiments are conducted to demonstrate the effectiveness of our approach in Section V. Finally, we conclude the paper with future research directions in Section VI.

II. LITERATURE REVIEW

The scheduling models that explicitly optimize the robustness and/or stability for the baseline schedule against stochastic and dynamic disruptions are reviewed in this section. The relevant works are organized using the framework proposed in [8], based on the manufacturing environment, the schedule generation, and the schedule evaluation, which optimizes the robustness and stability measures for the single-machine scheduling problems. Moreover, “*Multiobj*” and “*BOM*” (behavioral operations management) terms are added to fit our problem as in Table I. “*Multiobj*” indicates that the robustness and stability are optimized in a Pareto-optimization manner instead of using a linearly weighted combination of the duo. “*BOM*” indicates that the behaviors of the system participants are considered.

In the jobshop environment, Leon *et al.* [12] were among the first to consider jobshop scheduling with stochastic machine breakdowns. A convex combination of the expected C_{\max} of the realized schedule and the expected performance degradation was used to measure robustness, which was approximated by a slack-based surrogate measure. Jensen [15] proposed another representative surrogate measure of robustness based on the cur-

rent schedule’s neighborhood, which was defined as the set of schedules generated from the incumbent by exchanging adjacent operations. Then, the robustness for the schedule was measured by the average performance over its neighborhood. For the uncertain processing time, Zuo *et al.* [18] established a set of workflow models and accordingly proposed a multiobjective immune algorithm to obtain the Pareto-optimal robust schedules. Goren *et al.* [21] designed an exact branch-and-bound algorithm for the problem with a surrogate stability measure and proposed a Tabu Search method to deal with the stochastic machine breakdown. The above papers all assume a fixed job-machine assignment. Refer to [22] for more complicated flexible jobshop scheduling problems that optimize both robustness and stability.

In the single-machine environment, the model constraints often involve the job release time and the objective is the due-date-related performance measure such as $\sum w_j U_j$ or $\sum w_j T_j$. O’Donovan *et al.* [14] proposed a proactive–reactive approach to deal with stochastic machine breakdowns and dynamic job processing time. Seavax and Rensen [16] proposed a hybrid genetic algorithm embedded with a simulation model to optimize the weighted sum of robustness and stability. Similarly, Liu *et al.* [17] calculated the weighted sum of robustness and stability using a surrogate measure of aggregated machine breakdowns. Goren and Sabuncuoglu [8] applied the surrogate measure proposed in [17], and compared different robustness and stability measures as well as rescheduling methods. Assuming job’s processing time to be stochastic, Goren and Sabuncuoglu [19] analyzed the optimal properties of the problem’s special cases, and thereafter designed an exact branch-and-bound algorithm for the small-sized problems and an effective Beam Search heuristic for the large-sized problems.

In the flowshop environment, Chaari *et al.* [20] used scenario modeling to consider jobs’ stochastic processing time. Robustness was measured by the schedule’s makespan deviation between all disrupted scenarios and the initial scenario without disruption. Rahmani and Heydari [23] built up a mixed-integer-programming (MIP) model and proposed a predictive–reactive approach for dynamic arrivals of new jobs and uncertain processing times. In the predictive stage, a robust optimization method was used to obtain the baseline schedule, which was updated by rescheduling after unexpected jobs arrived in the reactive stage.

To summarize, through the literature review, we can identify a few unsolved research questions.

- 1) Most works only consider one type of disruptions, that is, machine availability. However, job-related events (unexpected arrival or delay) are also common in the flowshop scheduling problems. Models that consider all of these disruptions are needed.
- 2) Only a few papers [8], [16], [17], [21] have simultaneously considered robustness and stability in the proactive stage. However, as advocated by Goren *et al.* [21], more effective approaches are needed to see a tradeoff between the two measures.
- 3) None of the reviewed articles consider the dissatisfactions of the system participants in reaction to disruptions, which however needs to be investigated and measured. Our research attempts to provide answers to these questions.

TABLE I
BIBLIOGRAPHY OF SCHEDULING LITERATURES EXPLICITLY ADDRESSING ROBUSTNESS AND/OR STABILITY FOR UNCERTAIN DISRUPTIONS

| | Environment | | | Method | Schedule Generation and Evaluation | | | | | |
|------|-----------------------------------|--------------------|------------------------------|-------------------------|---|--|-----|--------------|-------------------------|--------------------------------|
| | Initial | Static/ Dynamic | Deterministic/ Stochastic | | Robustness | Stability | BOM | Multiobj | When to | How to |
| [12] | $J C_{max}$ | Static | stoch brkdwn | GA | $\mathbb{E}(\text{deviation})$ | N/A | N/A | N/A | Periodic | RS |
| [13] | $J L_{max}$ | Static | stoch brkdwn | OSMH/LPH | N/A | $ C_j^R - C_j^P $ based | N/A | N/A | Periodic | RS |
| [14] | $1 r_j \sum T_j$ | Static | stoch brkdwn/ p_i | OSMH ATC derivatives | N/A | $ C_j^R - C_j^P $ based | N/A | N/A | Continuous | RS/ATC derivatives |
| [15] | $1 r_j C_{max}$ | Static | brkdwn | GA | neighborhood | N/A | N/A | N/A | Continuous | RS/hillclimbing/ reschedule |
| [16] | $1 r_j \sum w_j U_j$ | Static | stoch r_i | GA | sample | distance | N/A | Weighted sum | Periodic | RS |
| [17] | $1 r_j \sum w_j T_j$ | Static | stoch brkdwn | GA | $\mathbb{E}(\sum w_j T_j)$ | $ C_j^R - C_j^P $ based | N/A | Weighted sum | Periodic | RS |
| [8] | $1 r_j C_{max}/\sum C_j/\sum T_j$ | Static | stoch brkdwn | TS | $\mathbb{E}(C_{max}/\sum C_j/\sum T_j)$ | $ C_j^R - C_j^P $ based | N/A | Weighted sum | Periodic/ Continuous | RS/reschedule |
| [18] | $J C_{max}$ | Static | stoch p_i | VNIA | standard deviation | N/A | N/A | N/A | Periodic | RS |
| [19] | $1 r_j \sum C_j/\sum T_j$ | Static | stoch brkdwn/ p_i | B&B/BS | $\mathbb{E}(\sum C_j/\sum T_j)$ | $ C_j^R - C_j^P $ based | N/A | N/A | Periodic | RS |
| [20] | hybrid $F C_{max}$ | Static | scenarios of p_i | GA | sample | N/A | N/A | N/A | Periodic | RS |
| [21] | $J C_{max}$ | Static | stoch brkdwn/ p_i | B&B/TS | $\mathbb{E}(C_{max})$ | $ C_j^R - C_j^P $ based | N/A | Constraint | Periodic | RS |
| [22] | flexible $J C_{max}$ | Static | stoch brkdwn | GA | simulation/slack | N/A | N/A | N/A | Periodic | RS |
| [23] | $F_2 C_{max}$ | Dynamic | stoch p_i | MIP | regret | N/A In planning; $ C_j^R - C_j^P $ based in reaction | N/A | Weighted sum | Continuous | reschedule |

stoch: stochastic. brkdwn: breakdown. B&B: branch-and-bound. BS: beam search. TS: tabu search. $\mathbb{E}(\cdot)$: the expected value of a stochastic variable. p_j, w_j, r_j : job j 's processing time, weight, and release time. C_{max} : the makespan. L_{max} : the maximum lateness. $\sum w_j U_j$: the total weighted number of tardy jobs. $\sum T_j$: the total tardiness. C_j^R and C_j^P : job j 's completion time in the realized and planned baseline schedules. The absolute difference $|C_j^R - C_j^P|$ is commonly used to measure stability.

RS: Right-Shifting. "periodic" means to make decisions periodically. "continuous" means to reschedule all the unfinished jobs whenever a disruption occurs, also known as the event-driven rescheduling policy [28].

III. PROBLEM FORMULATION

The notations for parameters/variables used in defining the problem are summarized as follows. Notations in bold font denote vectors.

Notations

- i : The machine index.
 j : The job/customer index.
 k : The system participator index. $k = 1$ for the manager, $k = 2$ for the shopfloor operator, and $k = 3$ for the customers.

Parameters

- m : The number of machines.
 n_0 : The number of jobs/customers for the initial problem before disruptions occur.
 p_{ij} : The processing time of job j 's operation on machine i , $1 \leq i \leq m, 1 \leq j \leq n_0$.
 x : The disruption for the system participator, that is, the degraded schedule performance for the manager, the variation of job sequence for the shopfloor operator, and the variation of jobs' completion time for the customers.
 $V(x)$: The value variation of the system participator in a gaining or losing situation in the prospect theory.
 α, β : The degrees of risk-aversion and risk-preference in $V(x)$, when the behavioral subject faces gaining and losing situations, and $0 < \alpha, \beta < 1$.

- λ : The risk-aversion coefficient in $V(x)$, and $\lambda > 1$, meaning the subject is more sensitive to loss (see [12] for more details about the range of these parameters).
 O_k : The reference point for the system participators and $O_k \geq 0, k = 1, 2, 3$.
 $\mu_k(x)$: The dissatisfaction function of measuring stability for the system participators, $k = 1, 2, 3$.

Variables

- $\pi = \{\pi_1, \pi_2, \dots, \pi_{n_0}\}$: A feasible schedule, where $\pi_j = [j]$, $[j]$ is the index of job in the j th position, $1 \leq j \leq n_0$.
 $C_{i[j]}$: The completion time of job $[j]$ on machine i in π , $1 \leq i \leq m, 1 \leq j \leq n_0$.
 C_{mj}^P : The completion time of job j in the planned baseline schedule, $1 \leq j \leq n_0$.
 C_{mj}^R : The completion time of job j in the realized schedule, $1 \leq j \leq n_0$.

A. Deterministic Flowshop Problem

We begin our discussion with the deterministic flowshop scheduling problem to minimize the total flow time. n_0 jobs, each of which corresponds to a customer, are to be processed on m machines. Each job must visit all machines exactly once in the same sequence. The buffer/storage capacities in between any two successive machines are supposed to be infinite, that is, if a job finishes processing on a certain machine, it may leave immediately to start or wait for processing on the next machine.

For each machine, only one operation of a job can be processed at one time. All machines and jobs are available at the beginning of the planning horizon.

For a given baseline schedule π , we have $C_{i[j]} = \max(C_{i,[j-1]}, C_{i-1,[j]}) + p_{i[j]}$, $2 \leq i \leq m, 2 \leq j \leq n_0$; $C_{1[j]} = C_{1[j-1]} + p_{1[j]}$, $2 \leq j \leq n_0$; $C_{i[1]} = C_{i-1,[1]} + p_{i[1]}$, $2 \leq i \leq m$; and $C_{1[1]} = p_{1[1]}$. The completion time of each job in π is $C_{m,j}$, $1 \leq j \leq n_0$, and the objective is to minimize the total flowtime $\sum_{j=1}^{n_0} C_{m,j}$. This objective guarantees that jobs are rapidly turned around, and that the work-in-process inventory is minimized. Using the classic three-field representation scheme, we denote the deterministic problem as $F_m || \sum_{j=1}^{n_0} C_{m,j}$, which has been proved to be NP-hard in [25]. Many metaheuristics have been designed to solve this problem [4], including particle swarm optimization (PSO), iterated greedy algorithm, discrete differential evolutionary algorithm, and genetic local search algorithm. The more recent quantum-inspired iterated greedy (QIG) algorithm [26], which uses a hybrid representation of a job and a Q-bit for permutation and a new rotation gate to adaptively update the population, is very effective as it outperforms all the above-mentioned algorithms in solving the Taillard benchmark instances. Therefore, we adopt QIG to solve our initial problem, and the obtained solutions will serve as benchmarks.

B. Proactive-Reactive Approach

According to their effects on the baseline schedule, we classify the disruptions into two categories: stochastic machine breakdown requiring repair and unexpected job-related disruptions (a number of jobs dynamically arrive or a certain job in the baseline schedule is delayed with respect to its original starting time).

We adopt a proactive-reactive approach for this scenario. For a stochastic machine breakdown, we consider the general breakdown/repair planning horizon to be composed of several uptime and downtime (repair) periods. One uptime period is immediately followed by a downtime period, and vice versa. Assume that the uptime and downtime periods are both subject to identical and independent distributions with probability density functions $g(t)$ and $h(t)$ obtained from historical data, respectively [8]. In the proactive stage, the objective is to minimize the robustness and stability of the baseline schedule. Preemption is not allowed unless the processing is interrupted by a certain machine breakdown. The ‘‘Preempt-resume’’ policy is adopted, which means partial processing is allowed when a job is interrupted, and only the remaining part can be processed after the machine is repaired. In contrast to a stochastic machine breakdown, we assume that the newly arriving jobs and the delay in job availability cannot be anticipated and can only be handled after they occur. Thus, we need to take appropriate rescheduling actions in the reactive stage to minimize the realized schedule performance and the deviation from the baseline schedule.

The formal robustness and stability measures of a given baseline schedule are defined. First, the stochastic machine breakdown and repair make the robustness objective measure also stochastic. In practice, the decision maker is often more con-

cerned about $C_{m,j}^R$ than $C_{m,j}^P$. Let $f_1 = \sum_{j=1}^{n_0} C_{m,j}^R$ be the realized schedule performance. Following [8] and [19], we use the expected realized performance $\mathbb{E}(f_1) = \mathbb{E}(\sum_{j=1}^{n_0} C_{m,j}^R)$ as the robustness objective.

Stability measures the deviation between the realized and the baseline schedules when disruptions occur. We apply the prospect theory to quantify the dissatisfactions of the system participants to disruptions into the stability measure [10]. As is shown in Fig. 1(a), we define the value function $V(x)$ for the system participants

$$V(x) = \begin{cases} x^\alpha & x \geq 0 \\ -\lambda(-x)^\beta & x < 0 \end{cases} \quad (1)$$

where $0 < \alpha, \beta < 1, \lambda > 1$. 0 is the reference point for judging whether a subject is losing or gaining. From (1), the behavioral subjects’ reactions to gaining or losing are asymmetric, which makes our stability measure more practical and meaningful. The subject at the losing region is dissatisfied when the deviation is larger than what is expected. Thus, given the behavioral subject’ deviation x from the baseline schedule and the reference point O_k , we modify (1) by assuming that the subject is at the losing region for $x \geq O_k$.

The decision maker aims to mitigate the risks in the planning stage and reduce the deviations caused by disruptions in the execution stage. Since the deviation measures and reference points for the three behavioral subjects are different, (1) cannot be directly applied to measure the dissatisfactions. Therefore, we convert the value loss part of (1) ($x < 0$) to a function whose output is a real value number in $[0, 1]$, to represent the magnitude of dissatisfaction for a subject.

$$\mu_k(x) = \begin{cases} 1 & x \geq \overline{O}_k \\ \lambda_k(x/O_k - 1)^{\beta_k} & O_k \leq x < \overline{O}_k \\ 0 & 0 \leq x < O_k \end{cases} \quad (2)$$

where $\overline{O}_k = O_k(1 + \lambda_k^{-\beta_k^{-1}})$ can be obtained by solving x from $\lambda_k(x/O_k - 1)^{\beta_k} = 1$ and the result defines the deviation value for a subject when he/she is totally dissatisfied, $k = 1, 2, 3$. 0 means full satisfaction and 1 means full dissatisfaction. The formula $\mu_k(x)$ is also illustrated in Fig. 1(b). Refer to [10] for the details. With the increase of the deviation x , the subject’s dissatisfaction $\mu_k(x)$ becomes stronger. When $x \geq \overline{O}_k$, the subject becomes totally dissatisfied. If the dissatisfaction accumulates over a long period without unleashing, we will lose the customers as well as the talented workers.

1) *Manager’s Dissatisfaction*: The primary concern of the manager is the manufacturing cost, which is measured by $\sum_{j=1}^{n_0} C_{m,j}^R$. The dissatisfaction of the manager is measured by

$$\mu_1(f_1^R) = \begin{cases} 1 & f_1^R \geq \overline{O}_1 \\ \lambda_1(f_1^R/f_1^* - 1)^{\beta_1} & f_1^* \leq f_1^R < \overline{O}_1 \\ 0 & 0 \leq f_1^R < f_1^* \end{cases} \quad (3)$$

where $\overline{O}_1 = f_1^*(1 + \lambda_1^{-\beta_1^{-1}})$, and $O_1 = f_1^*$ denotes the optimal manufacturing cost for the baseline schedule, and f_1^R denotes the manufacturing cost for the original n_0 jobs in the realized

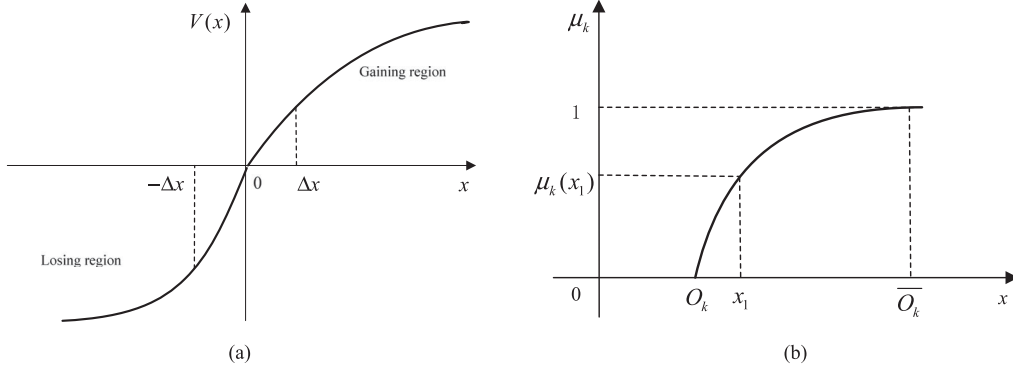


Fig. 1. Stability measure based on the prospect theory from BOM perspective. (a) Value function based on the prospect theory. (b) Dissatisfaction for losing region.

schedule obtained through QIG. For simplicity, (3) is abbreviated as μ_1 .

2) *Shopfloor Operator's Dissatisfaction*: The shopfloor operator mainly focuses on the resource reallocations and the shift rearrangements, which are reflected in the sequence variance with respect to the baseline plan [27]. Denote job j 's positions in the planned and the realized schedules as s_j^P and s_j^R , respectively, $1 \leq j \leq n_0$. Taking job j 's original position as the reference point, we can calculate the relative position variance for job j as $g_j = |s_j^P - s_j^R|/n_0$. It can be easily obtained that $0 \leq g_j < 1$. Since the reference point for job j is $O_2 = 0$, the shopfloor operator's dissatisfaction for job j is measured by

$$\mu_2(g_j) = \begin{cases} 1 & g_j \geq \overline{O_2} \\ \lambda_2 g_j^{\beta_2} & 0 \leq g_j < \overline{O_2} \end{cases} \quad (4)$$

where $\overline{O_2} = \lambda_2^{-1/\beta_2}$, and $1 \leq j \leq n_0$. For n_0 jobs in the proactive stage, the dissatisfaction of the shopfloor operator is measured by

$$\mu_2 = \frac{1}{n_0} \sum_{j=1}^{n_0} \mu_2(g_j). \quad (5)$$

3) *Customer's Dissatisfaction*: The customer's dissatisfaction is mainly caused by the deviation from the original delivery date [28], which is C_{mj}^P in the baseline schedule, $1 \leq j \leq n_0$. The reference point for customer j is set as $O_3 = C_{mj}^P$. Following [10], we assume that customers are homogeneous, that is, they use the same coefficients for risk-preference and risk-aversion. The dissatisfaction of customer j is measured by

$$\mu_3(C_{mj}^R) = \begin{cases} 1 & C_{mj}^R \geq \overline{O_3} \\ \lambda_3 (C_{mj}^R/C_{mj}^P - 1)^{\beta_3} & C_{mj}^P \leq C_{mj}^R < \overline{O_3} \\ 0 & 0 \leq C_{mj}^R < C_{mj}^P \end{cases} \quad (6)$$

where $\overline{O_3} = C_{mj}^P(1 + \lambda_3^{-1/\beta_3})$, $1 \leq j \leq n_0$. The average dissatisfaction for all the customers is

$$\mu_3 = \frac{1}{n_0} \sum_{j=1}^{n_0} \mu_3(C_{mj}^R). \quad (7)$$

Since the stability is measured by the dissatisfaction degrees of the three participators, we use the average of the three functions (5)–(7), $f_2 = (\mu_1 + \mu_2 + \mu_3)/3$, to represent the average

dissatisfaction. Different weights can also be used to reflect different relative importance, but the results are largely the same. Incorporating the stochastic machine breakdown, we define the stability measure function based on (3)–(7) as the expectation of f_2

$$\mathbb{E}(f_2) = \mathbb{E}((\mu_1 + \mu_2 + \mu_3)/3). \quad (8)$$

To summarize, our proactive–reactive approach works in the following way. First, a robust and stable schedule against the stochastic machine breakdown is obtained by minimizing $\mathbb{E}(f_1)$ and $\mathbb{E}(f_2)$ in the proactive stage. Then, the schedule is implemented. After new jobs arrive or the original jobs are delayed, we react to the dynamic events by reoptimizing the processing sequence of the original unfinished and new jobs, with two objectives of the schedule performance $\sum C_{mj}^R$ and the system participators' dissatisfaction $(\mu_1 + \mu_2 + \mu_3)/3$, using a hybridization-strategy-based EMO algorithm.

IV. HYBRIDIZED EMO METHODS

Our proposed problem is NP-hard because even a reduced form, that is, the deterministic $F_m || \sum_{j=1}^{n_0} C_{mj}$ is known to be NP-hard [25]. Thus, metaheuristics have attracted immense attention from the academia and industry [4], [29]. Refer to [44] for an effective and efficient cuckoo search-based memetic algorithm, embedded with a largest ranked-value-rule-based random key and an Nawaz-Enscore-Ham (NEH)-based initialization. And also refer to [45] for a differential evolution-based memetic algorithm, which solves the benchmark problems well and finds 161 new upper bounds for the well-known difficult Demirkol-Mehta-Uzsoy (DMU) problems. In order to examine the tradeoff between robustness and stability as suggested by Goren and Sabuncuoglu [8], [19] and Goren *et al.* [21], we design a hybridized EMO method to search for the Pareto front of robustness and stability. The most important advantage of this approach is that the decision maker can choose arbitrary solutions from a set of alternatives according to his/her preferences.

A. Evaluation of Baseline Schedule in the Proactive Stage

In the EMO approach, after a solution (a phenotype) is properly encoded into a genotype, the key issue is the evaluation

of the genotype that can build the searching process toward the desired regions with high-quality solutions. Genotype evaluations in the proactive and reactive stages are different. In the proactive stage, one cannot know exactly which machine would break down at what time and last for how long, whereas in the reactive stage all uncertain information has been revealed. The problem in the reactive stage is easier to be solved because the realized schedules can be used to calculate the performance and the stability measure. Thus, the difficulty lies in the evaluation of baseline schedule in the proactive stage. To address this issue, we propose surrogate measures to estimate the impact of disruptions in the proactive stage. To this end, we make extensions to the existing measures and propose three pairs of surrogate measures, (R_i, S_i) , $i = 1, 2, 3$, to approximate $(\mathbb{E}(f_1), \mathbb{E}(f_2))$.

1) *Slack-Based and Neighborhood-Based Surrogates* (R_1, S_1) : We extend the robustness measure based on the slack proposed by Leon *et al.* [12], and modify the neighborhood-based approach to measure the stability proposed by Jensen [15]. Given a baseline schedule π , let $s_{i[j]}^*$ and $C_{i[j]}^*$, $*$ $\in \{e, l\}$ denote the starting time and completion time of the i th operation of job $[j]$, where “e” in the superscript means earliest and “l” means latest. Since the original objective $\sum_{j=1}^{n_0} C_{mj}$ is regular, the current starting and completion times for all operations are $s_{i[j]}^e$ and $C_{i[j]}^e$. “Latest” indicates that any delay in an operation will not affect the completion times of the operations on the last machine, that is, $\sum_{j=1}^{n_0} C_{mj}$ will not be affected. For each operation, we have $s_{i[j]}^l = C_{i[j]}^l - p_{i[j]}$, and the latest completion time of the operation is determined equation (9) as shown bottom of this page.

In [22], the slack of job $[j]$ ’s i th operation $C_{i[j]}^l - C_{i[j]}^e$ is used to define the surrogate robustness measure $\sum_{i=1}^m \sum_{j=1}^{n_0} (C_{i[j]}^l - C_{i[j]}^e)$. We modify this measure by considering the probability of machine breakdown during $[C_{i[j]}^e, C_{i[j]}^l]$ to define R_1

$$R_1 = \sum_{i=1}^m \sum_{j=1}^{n_0} \omega_{i[j]} \cdot (C_{i[j]}^l - C_{i[j]}^e) \quad (10)$$

where $\omega_{i[j]} = \int_{C_{i[j]}^e}^{C_{i[j]}^l} h(t)dt$, and $h(t)$ is the probability density function for the downtime.

Next, for a baseline schedule π to be evaluated, we define its neighborhood $N_1(\pi)$ as the set of schedules generated by exchanging any two adjacent jobs (this can be seen as a noise) in π , and $|N_1(\pi)| = n_0 - 1$. By modifying the mean-variance model for $N_1(\pi)$, we define the surrogate stability S_1 of π as

$$S_1 = \frac{1}{3} (\min_{\pi' \in N_1(\pi)} (f_2(\pi')) + \max_{\pi' \in N_1(\pi)} (f_2(\pi')) + \text{mean}_{\pi' \in N_1(\pi)} (f_2(\pi')) \cdot \text{var}_{\pi' \in N_1(\pi)} (f_2(\pi')) \quad (11)$$

where $\text{mean}(\cdot)$ and $\text{var}(\cdot)$ are the average value and the variance of the input variable.

This stability measure is used on $N_1(\pi)$ because there is no positive or apparent correlation between the mean and the variance. The first component of S_1 is used to depict the spread of performance, and to avoid overestimation or underestimation of the stability. The second component of S_1 penalizes the phenotype with a larger variance, so that the stable solutions are preferred. From (10) and (11), both R_1 and S_1 can be regarded as the surrogate measures for evaluating a baseline schedule before uncertainty unfolds in the proactive stage.

2) *Breakdown and Repair Estimation-Based Surrogates* (R_2, S_2) : We apply the single-machine estimation method used in [8] to the flowshop case. This method enables us to quickly generate a realized schedule and calculate its performance for the baseline schedule. We construct an approximate schedule by inserting an estimated repair time into the baseline schedule after the machine has been busy for a given period. The underlying assumption is that the machine may possibly break down after a busy period and require repairing. The length of the period is given by $\omega L + (1 - \omega)U$, and the expected repair time is given by $\mathbb{E}(h(t))$. ω is a given weight in $[0, 1]$ (e.g., $\omega = 0.6$ in [8]). L and U are the $(1000 \cdot \alpha/2)$ th and the $[1000 \cdot (1 - \alpha/2)]$ th tiles of the 1000-tile breakdown time distribution. The probability of machine breakdown during $[L, U]$ is $1 - \alpha$. In this paper, we choose $\alpha = 0.05$. Given a baseline schedule π , we obtain a realized schedule π' after the repair insertion. Then, the robustness and stability measures of π are estimated as $R_2 = f_1(\pi')$, $S_2 = f_2(\pi')$.

The repair time insertion method is different from the idle time insertion method used in [13], where the authors did not use the machine breakdown information. We estimate the breakdown time using distribution tiles information.

3) *Simulation-Based Surrogates* (R_3, S_3) : We build a simple event simulation model to tackle uncertainty in the phenotype evaluation, as shown in Fig. 2. The *EMO metaheuristic* passes each phenotype of the population to the *Simulation model* for evaluation. Uncertainty is simulated by variables that follow the machine breakdown and repair distributions $g(t)$ and $h(t)$. At the end of each run, a realized schedule can be obtained and the corresponding robustness and stability can be calculated. This process is repeated for N times in order to get a sample of robustness and stability, and the *Results analysis* returns the fitness of the phenotype.

For a given phenotype π , a sample of size N can be obtained through simulations, and R_3 and S_3 of π are then calculated through the mean-variance model as $R_3 = \frac{1}{3} (\min(f_1) + \max(f_1) + \text{mean}(f_1)) \cdot \text{var}(f_1)$, $S_3 = \frac{1}{3} (\min(f_2) + \max(f_2) + \text{mean}(f_2)) \cdot \text{var}(f_1)$.

Clearly different baseline schedule can be generated under a different measure (R_i, S_i) , $i = 1, 2, 3$, and they need further

$$C_{i[j]}^l = \begin{cases} \min \{s_{i+1,[j]}^l, s_{i,[j+1]}^l\} & i = 1, \dots, m-1, j = 2, \dots, n_0-1 \\ s_{i+1,[j]}^l & i = 1, \dots, m-1, j = n_0 \\ C_{i[j]}^e & i = 1, \dots, m-1, j = 1; i = m, j = 1, \dots, n_0-1 \end{cases} \quad (9)$$

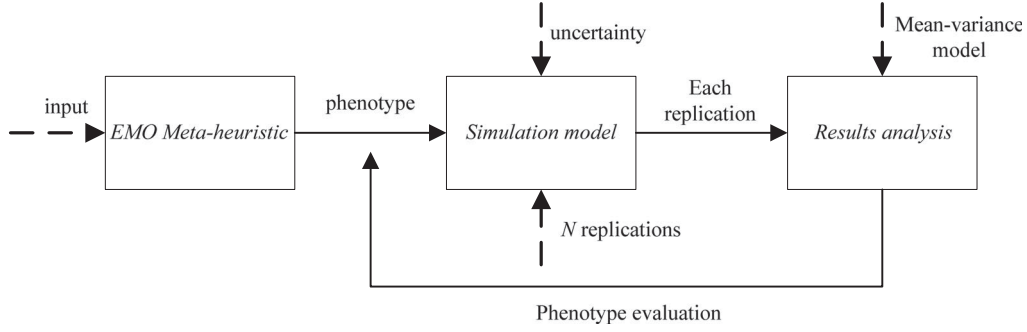


Fig. 2. Simulation-based stability and robustness approximation.

performance evaluation, which will be elaborated in Section V. From the computational perspective, (R_3, S_3) would incur the highest complexity because each replication in the simulation can be seen as an implementation of (R_2, S_2) . The complexity of (R_1, S_1) is also higher than (R_2, S_2) since the construction of the neighborhood involves $n_0 - 1$ times evaluations of the robustness.

B. New Hybridization Strategy

According to Yenisey and Yagmahan [4], the metaheuristics for solving permutation flowshop problems can be categorized into two classes: the population-based methods [genetic algorithm, particle swarming algorithm, and quantum-inspired genetic algorithm (QGA)] and the trajectory methods (iterated greedy, simulated annealing, and Tabu Search). In the searching process, the population-based methods maintain and update a population with high parallelism, self-construction, self-learning, and self-adaptation. However, the local search ability needs to be strengthened. The trajectory methods start from a randomly generated solution, and then improve the initial solution iteratively according to various mechanisms until a near-optimal and satisfactory solution is found. Considering the strengths and weaknesses of the two types of methods, we propose a new hybridization strategy through initial solution improvement [17], [30] and exploitation intensification [3]. The quality of the initial solutions positively affects that of the final solutions. And exploitation intensification enhances the local search ability and helps escaping from the local optimum.

The hybridization strategy combines an inner loop for exploitation intensification with an outer loop for exploration. First, the initial solutions with high quality are generated and passed to the inner loop from the outer loop. Second, after receiving the initial solutions, the inner loop would exploit the solution space to balance exploration with exploitation using different searching mechanisms. Finally, the qualified solutions obtained from the two loops are selected and combined as the starting solutions for evolution in the next iteration. The hybrid strategy can lower the impact of the quality of the initial solutions on the algorithm performance and can also maintain those elite individuals by combining the results of the two loops. As is shown in Fig. 3, NSGA-II [31] is selected as the inner loop. The algorithm provides a unique Pareto front classification and crowding distance calculation which can maintain the

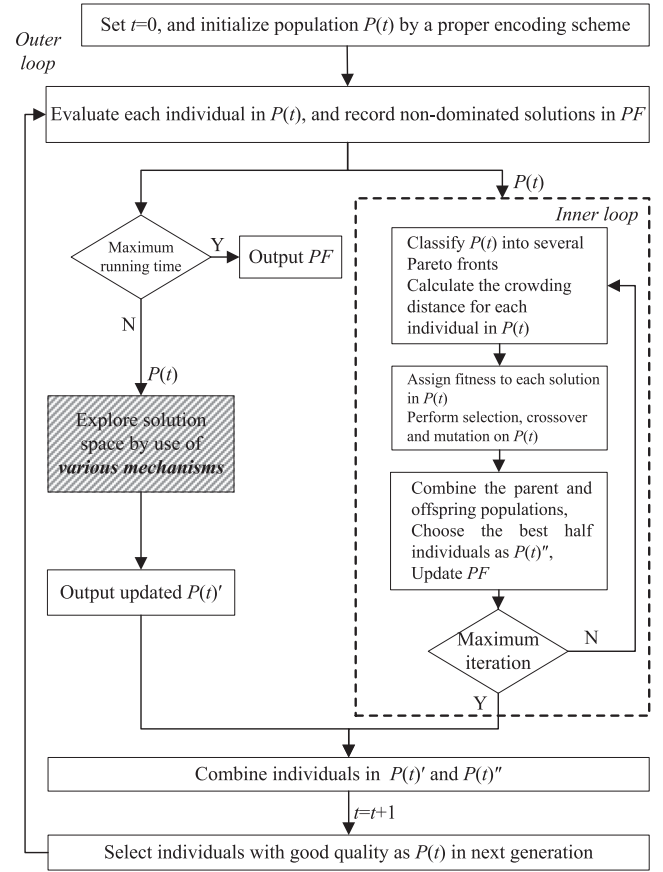


Fig. 3. Flowchart of the hybridization strategy.

proximity and diversity of the solutions during evolutions. When the outer-loop method is population-based, the initial population in NSGA-II in each iteration is selected from the high-quality output in the outer loop. When the outer-loop method is trajectory, the initial population is constituted by one of popsize - 1 randomly generated solutions, where popsize is NSGA-II's population size. In Fig. 3, we finally obtain four hybridized algorithms by implementing different searching mechanisms in the shaded rectangle.

C. Four Hybridized EMO Algorithms

For the population-based outer loop, the QGA and the PSO algorithm are selected. The unique Qubit encoding scheme in

QGA is superior in maintaining the population diversity [29], [32]. Meanwhile, PSO algorithm searches the entire solution space randomly and iteratively, which is easy-to-apply with a low computational complexity. For the trajectory outer loop, the simulate annealing (SA) algorithm and the quantum iterated greedy (QIG) algorithm are selected. SA is generally applicable for combinatorial optimization problems, and is capable of escaping the local optimum through probabilistic jumping. Using a quantum-based encoding scheme, QIG, originally designed for solving $F_m || \sum_{j=1}^{n_0} C_{mj}$, searches the solution space iteratively in a greedy manner. It is anticipated that QIG can return promising results for our problem. For QGA, PSO, and SA in the outer loop that are not specially designed, the weighted linear combination is used to deal with the biobjectives. The weights for the first and second objectives are $\text{rand}_1/(\text{rand}_1 + \text{rand}_2)$ and $\text{rand}_2/(\text{rand}_1 + \text{rand}_2)$, where rand_1 and rand_2 are uniformly generated from $[0, 1]$. Therefore, the objective to be optimized in the outer loop is the weighted sum of the two given objectives.

1) *Hybrid Quantum Genetic Algorithm (HQGA)*: In HQGA, the outer loop is based on the QGA. First, the encoding scheme $\begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_q \\ \beta_1 & \beta_2 & \dots & \beta_q \end{bmatrix}$, which can be considered as a string of Qubit $(\alpha_l \beta_l)^T$ as the smallest information unit in the Quantum theory, is used to represent a job. α_l^2 and β_l^2 specify the probabilities that a Qubit is found at $|0\rangle$ state and $|1\rangle$ state, respectively, with $|\alpha_l|^2 + |\beta_l|^2 = 1, l = 1, 2, \dots, q$. Therefore, n_0 jobs will require a Qubit string of length $n_0 q$, and $q = \log_2 n_0$, where the operator $\lceil \cdot \rceil$ returns the smallest integer that is no less than the input value.

As shown in Fig. 3, the input for the shaded rectangular in the outer loop of the HQGA is either an initialized population $P(t)$ through the Qubit encoding when $t = 0$, or an updated population $P(t)$ when $t \geq 0$. The population is obtained by classic genetic operations. First, all individuals in the population are ranked according to their fitness values. Then, the individual with the highest fitness is selected as one parent, while the other parent is determined by another randomly selected individual. A two-point crossover operator is implemented to produce an offspring, and a swap or insert operation is used for mutation [29].

After genetic operations are completed, each Qubit is converted to 0 or 1 through the quantum collapse. For each job, a 0–1 binary string is obtained and then further converted to a decimal number. Following this process, a permutation of n_0 jobs can be achieved by applying a random-key representation to the n_0 -length real value vector. During the evolution, each genotype is compared with the up-to-now best genotype. Then, a rotation angle θ_l is obtained from the rotation angle table for performing the update operation $\begin{bmatrix} \alpha_l^{t+1} \\ \beta_l^{t+1} \end{bmatrix} = \begin{bmatrix} \cos(\theta_l) & -\sin(\theta_l) \\ \sin(\theta_l) & \cos(\theta_l) \end{bmatrix} \cdot \begin{bmatrix} \alpha_l^t \\ \beta_l^t \end{bmatrix}$, $l = 1, 2, \dots, q$, and t is the generation number (see [40] for more details).

2) *Hybrid Particle Swarm Optimization (HPSO)*: In HPSO, each permutation solution is encoded as a particle at a certain position which flies at a certain speed in the search

space. For generation t , particle l 's position vector is defined as $\mathbf{X}_l^t = \{x_{l1}^t, x_{l2}^t, \dots, x_{ln_0}^t\}$, and its speed vector is given as $\mathbf{V}_l^t = \{v_{l1}^t, v_{l2}^t, \dots, v_{ln_0}^t\}$, $v_{lj} \leq V_{\max}, 1 \leq j \leq n_0$. \mathbf{X}_l^t is real-value coded, and can be converted to a jobs' sequence permutation through the random-key representation method. V_{\max} is used to limit the region of particles. The j th elements in \mathbf{X}_l^t and \mathbf{V}_l^t are defined for job $[j]$, $1 \leq j \leq n_0$. Up to the t th generation, the best position vector for particle l is recorded as $\mathbf{P}_l^t = \{p_{l1}^t, p_{l2}^t, \dots, p_{ln_0}^t\}$. Let \mathbf{g}_{best} be the best position vector of the entire particle population up to the t th generation.

The speed and position of each particle must be updated to achieve information sharing, so that the entire population becomes regulated. The update operation is $\mathbf{V}_l^{t+1} = \omega \cdot \mathbf{V}_l^t + c_1 \cdot \gamma_1 \cdot (\mathbf{P}_l^t - \mathbf{X}_l^t) + c_2 \cdot \gamma_2 \cdot (\mathbf{g}_{\text{best}} - \mathbf{X}_l^t)$ and $\mathbf{X}_l^{t+1} = \mathbf{X}_l^t + \mathbf{V}_l^{t+1}$. ω is the coefficient for a particle to keep its current speed inertial, c_1 and c_2 are positive learning factors often assuming value 2, and γ_1 and γ_2 are random numbers sampled from a uniform distribution $[0, 1]$.

3) *Hybrid Quantum-Inspired Iteration Greedy (HQIG)*: The encoding scheme of HQIG combines permutation and qubit

as $\begin{bmatrix} 1 & 2 & \dots & n_0 \\ \alpha_1 & \alpha_2 & \dots & \alpha_{n_0} \\ \beta_1 & \beta_2 & \dots & \beta_{n_0} \end{bmatrix}$ with $|\alpha_l|^2 + |\beta_l|^2 = 1, l = 1, 2, \dots, n_0$.

Unlike the encoding scheme in the HQGA, each Qubit in the HQIG stands for one job. All jobs are divided into two partial schedules through quantum collapse on the initial schedule φ_c . Those jobs with state value 1 constitute a partial schedule φ_p , whereas those jobs with 0 constitute another partial schedule Ω . Jobs in φ_p and Ω maintain their original sequences in φ_c . Then, the NEH procedure is applied: jobs in Ω are inserted sequentially into φ_p to form a complete schedule, which is then locally improved through the variable neighborhood searching (VNS). When the NEH and VNS procedures are completed, a new schedule φ_n is formed.

Let φ^* be the best solution searched so far. φ_n , φ_c , and φ^* are used to compute the rotation angle for the l th Qubit: $\theta_l = \text{sign}(\alpha_l \beta_l) \cdot \{c_1 \gamma_1 [1 - f_1(\varphi_n)/f_1(\varphi_c)] + c_2 \gamma_2 [1 - f_1(\varphi_n)/f_1(\varphi^*)]\} \pi$ with $c_1 = c_2 = 1$, $\gamma_1, \gamma_2 \in [0, 1]$ and $\text{sign}(\cdot)$ returns the input's sign. We can see that the rotation angle is dynamically determined based on the relationship between solutions in HQIG. After the rotation angle is determined, the l th Qubit is updated using the same rotation gate formula in HQGA.

4) *Hybrid Simulated Annealing (HSA)*: In HSA, the encoding scheme of the outer loop is permutation-based. The energy of the current permutation is E , which is defined as the objective value obtained by the weighted combination. A new permutation is obtained by perturbation, which is achieved through randomly designating two positions and swapping the jobs arranged on these two positions. The energy of the new permutation is denoted as E' . If $E - E' \geq 0$, the current permutation is replaced by the new one. Otherwise whether or not to accept the new permutation depends on a probability $e^{(E-E')/T_t}$, and $T_t \geq 0$ is the annealing temperature for the iteration controller t .

A random number r is generated from $[0, 1]$. If $e^{(E-E')/T_t} > r$, the new permutation is accepted, and T_{t+1} is updated as $T_{t+1} = \varepsilon \cdot T_t$ for next iteration ($\varepsilon \in [0, 1]$ is the cooling down

factor); otherwise the new permutation is discarded. Based on the acceptance probability formula $e^{(E-E')/T_t}$, we can see that the higher the temperature is, the higher the probability of accepting an inferior solution is, and the more probable the SA jumps out of the local optimum. This iterative procedure terminates until the iteration controller t is larger than the given length of the Mapkob chain.

V. COMPUTATIONAL STUDY

The purpose of the computational study is to verify the hybridization strategy, to select the most effective one as the EMO method in the proactive-reactive approach, to evaluate the value of incorporating uncertainty into the baseline schedule, and to examine the impact of three pairs of the surrogate measures in the reactive stage when the stochastic breakdown and the unexpected events concurrently occur.

A. Testing Instances

Since there exists no similar research that can be found in the literature, we are not able to directly compare our approach with any benchmark cases. Therefore, we modify 90 flowshop instances in the TA class from the current Operations Research library for testing [33]. We consider two types of disruption seriousness in two scenarios, only stochastic machine breakdown in scenario 1, and unexpected new job arrival, delay in job availability and machine breakdown in scenario 2.

In scenario 1, we use the Gamma distribution to describe the machine breakdown and the repair duration [21]. Gamma distribution is determined by two parameters, shape and scale. In this case, the shape parameter was set to be 0.7, and the scale was determined as $\delta \cdot (\frac{1}{mn_0} \sum_{i=1}^m \sum_{j=1}^{n_0} p_{ij}) / 0.7$, where $\delta > 0$. A large value of δ indicates a long busy time and a low breakdown frequency. Similarly for the machine repair time, the shape parameter was set as 1.4, and the scale parameter was determined in a similar manner: $\varepsilon \cdot (\frac{1}{mn_0} \sum_{i=1}^m \sum_{j=1}^{n_0} p_{ij}) / 1.4$, $\varepsilon > 0$, and a large value of ε indicates a long repair time. In scenario 2, for new job arrivals, let n_N be the number of new jobs. The problem becomes more difficult to solve with a larger n_N . The processing time of new jobs on each machine is randomly generated from the uniform distribution [1, 99]. For delays in job availability, we randomly select one job in the baseline schedule and delay its planned starting time by a certain period. The length of that delay period is related to the average processing time that is adjusted by a positive coefficient $\tau > 0$: $\tau \cdot (\frac{1}{mn_0} \sum_{i=1}^m \sum_{j=1}^{n_0} p_{ij})$.

The parameters in both scenarios are summarized as follows: In scenario 1, $\delta = 10$, $\varepsilon = 0.5$ for Type 1, and $\delta = 3$, $\varepsilon = 1.5$ for Type 2. In scenario 2, $\delta = 10$, $\varepsilon = 0.5$, $n_N = 20$, $\tau = 0.5$ for Type 1, and $\delta = 3$, $\varepsilon = 1.5$, $n_N = 30$, $\tau = 1.5$ for Type 2. We can see that for each scenario, Type 2 implies a more difficult problem than Type 1.

B. Parameters Setting

The parameters of NSGA-II are used both separately and in the inner loop shared by the four hybridized methods. In order to obtain high-quality solutions during evolution, the probabilities

of crossover mutation in the inner loop are both set to be 1. After genetic operations, the new and original populations are combined to one set, from which the good individuals are selected to form the population of the next generation. This process keeps good quality candidates as well as reduces the negative impact from the high crossover and mutation probabilities. The termination of the inner loop is controlled by a maximum iteration number set to be 10.

For HQGA, a string of Qubits with length q is used to represent a job. Since the maximum number of jobs in our paper is 130 (100 jobs in TA81-90 instances and maximum 30 new jobs), we have $2^7 < 130 < 2^8$. Thus, we set $q = 8$. The value of the rotation angle θ is determined by the rotation table in [32].

For HPSO, the speed inertial coefficient ω is 0.5. γ_1 and γ_2 are randomly sampled from $[0, 1]$. The learning coefficients are set to be $c_1 = c_2 = 5$. The value of each particle position is in $[-100, 100]$, and its maximum speed is $V_{\max} = 20$.

For HSA, the termination temperature is set to be 0.01, the cooling down factor is set to be 0.9, and the length of the Mapkob chain is set to be 5. The tuning of the initial temperature T_0 is very important. Higher T_0 value incurs higher jumping probability, so high-quality solutions might be missed. In the same time, lower T_0 values are prone to get the searching process trapped in the local optimum. After a pilot testing, we find that when $T_0 = 10$, the outer loop runs seven iterations, the inner loop runs ten iterations, and the temperature falls from 10 to 4 in the given running time. Thus, 10 is a reasonable value for T_0 .

For HQIG, the initial solution is generated by applying the $LR(N/M)$ heuristic, and the parameters are set following Zhang and Li [26]: the number of local searches in the VNS is 1, and the probability of accepting an inferior solution is 0.01. The initialization of the Qubit encoding is $\alpha_l = \sqrt{1 - 9/n_0}$ and $\beta_l = \sqrt{9/n_0}$, $l = 1, \dots, n_0$.

For all EMO methods, the running time is used to control the termination of the entire procedure. We follow the procedure in [34] and set the population size of the NSGA-II, HQGA, and HPSO as $popsiz = n_0$, and the maximum running time of each algorithm is $3n_0$ CPU seconds. All the algorithms are coded in C++ language, and are tested on a PC with a CPU of Intel (R) Core (TM) 2 Duo 3.00 GHz and a 4G RAM.

Finally, we specify the parameters in the stability measure. For all three participants, the dissatisfaction parameters are set to be $\beta = 0.88$ and $\lambda = 2.25$ [10].

C. Performance Metrics for the Pareto Front

Since the EMO method is implemented in both proactive and reactive stages, the Pareto front of all nondominated solutions are calculated in two stages under various conditions. We outline several metrics in the literature for measuring the Pareto front's performance [29]. These metrics gauge the proximity to the optimal Pareto front and the diversity of the nondominated solutions.

First, the metrics for measuring the Pareto front's proximity contain CM, D_{ave} , D_{max} , and Rate. CM (C metric) is defined to quantify the dominance relationship between two Pareto fronts PF_1 and PF_2 . $CM(PF_1, PF_2)$ maps an ordered pair

(PF_1, PF_2) into a value in $[0,1]$, which indicates the proportion of the solutions in PF_2 that are dominated by the solutions in PF_1 . $CM(PF_1, PF_2) > CM(PF_2, PF_1)$ indicates PF_1 is better than PF_2 . The distance metrics D_{ave} and D_{max} are defined as the average and the maximum distances between a given Pareto front PF and the optimal PF . The smaller the distance metrics are, the closer PF is to the optimal one. As the optimal PF is often unknown beforehand. The common practice is to combine all the PF s for comparison into one set, and to construct the reference set \mathcal{R} of the optimal PF from all the non-dominated solutions in that set. D_{ave} and D_{max} are then calculated with respect to \mathcal{R} . Rate is defined as the proportion of the number of solutions in PF that appear in \mathcal{R} . The larger the Rate value, the better the proximity to the optimal PF .

Second, the metrics for measuring a PF 's diversity contain the overall nondominated vector generation (ONVG) and the Tan's spacing (TS). The ONVG is defined as $|PF|$ for a given PF , meaning the number of nondominated solutions. The larger the $|PF|$ value is, the better diversity it has. The TS metric is calculated as the solution's Euclid distance to its nearest neighbor in the solution space for a given PF . The smaller the TS value is, the more uniformly distributed the solutions in the PF are, and the better diversity the PF has.

Third, the metrics for measuring the PF 's average quality (AQ) contain the AQ metric, considering proximity and diversity simultaneously. It is defined as the average scalarized function value over a representative weight sample. The smaller the AQ value is, the better average quality the given PF has.

D. Computational Results

1) *Analysis of the Hybridization Strategy:* The benchmark problem instances TA081-090 are modified and tested with 30 new jobs arriving at the beginning of the baseline schedule. For each problem instance of TA081-090, independent cases are generated from the uniform processing time distribution [1, 99] of new jobs and solved by the EMO methods. In this paper, we have developed four hybridized EMOs based on NSGA-II. Besides NSGA-II, we also make comparison with two state-of-the-art EMO algorithms, the multiobjective evolutionary algorithm based on decomposition (MOEA/D, [46]) and the multiobjective memetic algorithms (MOMA, [48]). MOEA/D has successfully solved many difficult numerical testing problems, and outperforms NSGA-II and SPEA2 (the improved strength Pareto evolutionary algorithm) for the two objectives and three objectives benchmark flowshop scheduling instances [47]. Note that the metrics in Section V-C are often used to compare two alternative Pareto fronts. Thus, we modify these metrics to ensure that seven Pareto fronts can be compared simultaneously. For ONVG, the metric for each method is $\frac{(ONVG_{max} - ONVG_*)}{ONVG_{max}} \cdot 100\%$, and for AQ, the metric is $\frac{(AQ_* - AQ_{min})}{AQ_{min}} \cdot 100\%$, where $* \in \{HQIG, HSA, HPSO, HQGA, NSGA-II, MOEA/D, MOMA\}$. TS is kept as originally designed because when the PF has only two elements, its value is zero. The rest metrics, CM, D_{ave} , D_{max} , and Rate, are calculated with respect to the reference set \mathcal{R} , which is formed by combining seven Pareto

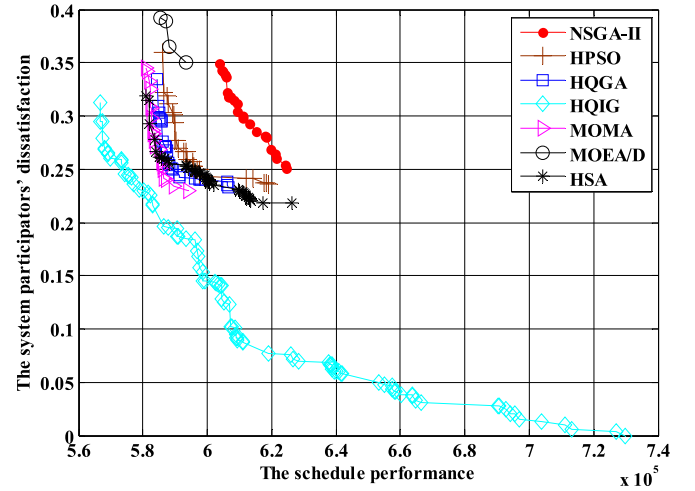


Fig. 4. Comparison of Pareto fronts returned by seven algorithms.

fronts and selecting the nondominated solutions in the set. \mathcal{R} is used as the first input in CM's formula $CM(\mathcal{R}, \cdot)$. To summarize, only the Rate metric, which is the proportion of solutions of each Pareto front in \mathcal{R} , is maximal and the other metrics are converted to be minimal. The comparison of PF s returned by seven algorithms for TA085 is shown in Fig. 4, and the rest results are similar. The detailed average values of all metrics are shown in Tables II and III. The best value of seven algorithms for each metric is highlighted in bold font.

We explain the results of metrics in Tables II and III. First, with respect to almost all metrics, HQIG shows the performance dominance, implying the Pareto front of HQIG has the best proximity and diversity. The evolution of Pareto fronts for HQIG in TA085 is shown in Fig. 5. This could be explained by that the original QIG is designed for $F_m || \sum_{j=1}^{n_0} C_{mj}$, and the rotation angle in HQIG is determined dynamically so that the perturbation strength during the evolution is very adaptive. This leads to better performances compared with the constant rotation angle method. The best TS metrics of ten instances are evenly distributed among all methods. Second, some metric values of HSA, HQGA, and HPSO are comparable to those of NSGA-II, MOEA/D, and MOMA. See ONVG, TS, and AQ. It is implied that their average qualities are close to each other. However, the D_{ave} and D_{max} metrics of HSA, HQGA, and HPSO are better than NSGA-II, MOEA/D, and MOMA almost all instances, indicating closer distances to the reference set \mathcal{R} . To summarize, the effectiveness of the hybridization strategy is well demonstrated.

Since we used the algorithm running time as the termination controller, and the population-based metaheuristics often take a long time to converge. It is highly likely that when a certain algorithm is forced to terminate at some point, the searching process is not completed. This case is shown in an instance of TA090 with 30 new jobs. If we allow the algorithm to run longer than the given running time, HQGA's and HPSO's performances would have been significantly improved. Thus, precisely speaking, only under the given running time, can HQIG and HSA provide better solutions than the other algorithms. Finally,

TABLE II
EFFECTIVENESS OF THE HYBRIDIZATION STRATEGY IN ONVG, TS, AND AQ

| Instances | ONVG | | | | | | | TS | | | | | | |
|-----------|------|-------------|------|------|------|------|------|--------------|--------|--------------|--------|--------------|-------------|--------------|
| | N | I | S | P | G | D | E | N | I | S | P | G | D | E |
| TA081 | 0.14 | 0.00 | 0.30 | 0.56 | 0.48 | 0.95 | 0.59 | 20.79 | 36.06 | 21.52 | 32.80 | 27.43 | 84.15 | 19.66 |
| TA082 | 0.57 | 0.00 | 0.54 | 0.44 | 0.50 | 0.96 | 0.92 | 17.63 | 42.17 | 23.97 | 26.35 | 20.45 | 2.44 | 108.22 |
| TA083 | 0.35 | 0.00 | 0.35 | 0.38 | 0.52 | 0.94 | 0.37 | 20.76 | 64.74 | 28.72 | 19.56 | 25.66 | 28.44 | 12.21 |
| TA084 | 0.50 | 0.00 | 0.27 | 0.77 | 0.50 | 0.94 | 0.64 | 25.81 | 62.69 | 19.07 | 11.52 | 28.21 | 8.49 | 16.26 |
| TA085 | 0.71 | 0.00 | 0.35 | 0.69 | 0.70 | 0.95 | 0.70 | 25.50 | 39.46 | 36.02 | 24.59 | 17.97 | 39.17 | 47.30 |
| TA086 | 0.65 | 0.00 | 0.61 | 0.68 | 0.66 | 0.95 | 0.86 | 27.55 | 42.63 | 23.30 | 115.92 | 33.21 | 38.88 | 84.57 |
| TA087 | 0.56 | 0.00 | 0.56 | 0.59 | 0.46 | 0.93 | 0.77 | 20.22 | 104.50 | 17.79 | 34.78 | 17.94 | 48.42 | 62.72 |
| TA088 | 0.69 | 0.00 | 0.55 | 0.49 | 0.43 | 0.95 | 0.91 | 51.97 | 26.43 | 20.46 | 41.51 | 18.71 | 39.00 | 101.50 |
| TA089 | 0.29 | 0.00 | 0.27 | 0.58 | 0.45 | 0.95 | 0.95 | 12.17 | 54.60 | 52.10 | 31.86 | 25.68 | 40.85 | 171.24 |
| TA090 | 0.67 | 0.00 | 0.44 | 0.43 | 0.51 | 0.97 | 0.71 | 33.80 | 24.72 | 29.05 | 10.36 | 25.61 | 0.00 | 106.12 |
| Instances | AQ | | | | | | | | | | | | | |
| | N | I | S | P | G | D | E | | | | | | | |
| TA081 | 0.03 | 0.00 | | | 0.03 | | 0.04 | | 0.04 | | 0.06 | | | 0.01 |
| TA082 | 0.04 | 0.00 | | | 0.03 | | 0.05 | | 0.03 | | 0.04 | | | 0.01 |
| TA083 | 0.03 | 0.00 | | | 0.02 | | 0.02 | | 0.02 | | 0.05 | | | 0.00 |
| TA084 | 0.02 | 0.00 | | | 0.01 | | 0.03 | | 0.02 | | 0.05 | | | 0.01 |
| TA085 | 0.06 | 0.00 | | | 0.02 | | 0.03 | | 0.03 | | 0.03 | | | 0.02 |
| TA086 | 0.06 | 0.00 | | | 0.01 | | 0.01 | | 0.02 | | 0.07 | | | 0.01 |
| TA087 | 0.04 | 0.00 | | | 0.00 | | 0.02 | | 0.01 | | 0.04 | | | 0.02 |
| TA088 | 0.06 | 0.00 | | | 0.01 | | 0.01 | | 0.02 | | 0.04 | | | 0.02 |
| TA089 | 0.03 | | 0.01 | | 0.03 | | 0.04 | | 0.03 | | 0.06 | | | 0.00 |
| TA090 | 0.04 | 0.00 | | | 0.02 | | 0.02 | | 0.01 | | 0.03 | | | 0.01 |

“N, I, S, P, G, D, E” stand for NSGA-II, HQIG, HSA, HPSO, HQGA, MOEA/D, and MOMA respectively.

TABLE III
EFFECTIVENESS OF THE HYBRIDIZATION STRATEGY IN CM, D_{ave} , D_{max} , AND RATE

| Instances | CM | | | | | | | D_{ave} | | | | | | |
|-----------|-----------|-------------|-------------|------|------|------|------|-----------|-------------|------|------|------|------|------|
| | N | I | S | P | G | D | E | N | I | S | P | G | D | E |
| TA081 | 0.58 | 0.00 | 0.72 | 0.53 | 0.50 | 0.70 | 0.13 | 0.38 | 0.00 | 0.34 | 0.39 | 0.37 | 0.62 | 0.38 |
| TA082 | 0.64 | 0.00 | 0.45 | 0.51 | 0.48 | 0.26 | 0.29 | 0.35 | 0.00 | 0.34 | 0.30 | 0.31 | 0.52 | 0.29 |
| TA083 | 0.53 | 0.00 | 0.56 | 0.64 | 0.44 | 0.53 | 0.55 | 0.27 | 0.00 | 0.27 | 0.23 | 0.25 | 0.51 | 0.25 |
| TA084 | 0.58 | 0.00 | 0.36 | 0.33 | 0.39 | 0.36 | 0.19 | 0.26 | 0.00 | 0.29 | 0.37 | 0.29 | 0.57 | 0.36 |
| TA085 | 0.51 | 0.00 | 0.52 | 0.49 | 0.39 | 0.29 | 0.29 | 0.42 | 0.00 | 0.35 | 0.36 | 0.37 | 0.59 | 0.35 |
| TA086 | 0.82 | 0.16 | 0.40 | 0.36 | 0.77 | 0.73 | 0.37 | 0.36 | 0.00 | 0.28 | 0.25 | 0.29 | 0.51 | 0.36 |
| TA087 | 0.51 | 0.05 | 0.27 | 0.35 | 0.33 | 0.38 | 0.54 | 0.30 | 0.00 | 0.27 | 0.33 | 0.27 | 0.55 | 0.34 |
| TA088 | 0.79 | 0.17 | 0.37 | 0.68 | 0.59 | 0.65 | 0.49 | 0.34 | 0.00 | 0.21 | 0.21 | 0.24 | 0.43 | 0.25 |
| TA089 | 0.57 | 0.39 | 0.06 | 0.56 | 0.56 | 0.62 | 0.36 | 0.26 | 0.01 | 0.22 | 0.27 | 0.23 | 0.45 | 0.35 |
| TA090 | 0.57 | 0.00 | 0.43 | 0.41 | 0.30 | 0.38 | 0.43 | 0.33 | 0.00 | 0.25 | 0.28 | 0.28 | 0.53 | 0.33 |
| Instances | D_{max} | | | | | | | Rate | | | | | | |
| | N | I | S | P | G | D | E | N | I | S | P | G | D | E |
| TA081 | 0.71 | 0.00 | 0.65 | 0.72 | 0.66 | 0.94 | 0.75 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| TA082 | 0.72 | 0.00 | 0.71 | 0.65 | 0.67 | 0.91 | 0.63 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| TA083 | 0.66 | 0.00 | 0.67 | 0.60 | 0.64 | 0.95 | 0.67 | 0.00 | 0.95 | 0.00 | 0.00 | 0.00 | 0.00 | 0.05 |
| TA084 | 0.60 | 0.00 | 0.70 | 0.77 | 0.68 | 0.94 | 0.77 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| TA085 | 0.72 | 0.00 | 0.68 | 0.66 | 0.70 | 0.89 | 0.67 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| TA086 | 0.72 | 0.02 | 0.70 | 0.67 | 0.68 | 0.88 | 0.81 | 0.00 | 0.84 | 0.00 | 0.16 | 0.00 | 0.00 | 0.00 |
| TA087 | 0.63 | 0.00 | 0.62 | 0.68 | 0.62 | 0.90 | 0.69 | 0.00 | 0.94 | 0.06 | 0.00 | 0.00 | 0.00 | 0.00 |
| TA088 | 0.77 | 0.02 | 0.69 | 0.69 | 0.71 | 0.95 | 0.75 | 0.00 | 0.79 | 0.18 | 0.03 | 0.00 | 0.00 | 0.00 |
| TA089 | 0.66 | 0.07 | 0.68 | 0.72 | 0.67 | 0.89 | 0.74 | 0.00 | 0.60 | 0.36 | 0.00 | 0.00 | 0.00 | 0.04 |
| TA090 | 0.70 | 0.00 | 0.64 | 0.65 | 0.68 | 1.00 | 0.70 | 0.00 | 0.88 | 0.05 | 0.00 | 0.07 | 0.00 | 0.00 |

comprehensively considering diversity, proximity, and efficiency, HQIG is selected as the EMO method for our proactive-reactive approach.

2) *Analysis of Robustness and Stability Measures Against Determined Schedule:* In this section, we evaluate the value

of incorporating uncertainty into the baseline schedule. In particular, we compare the performance of the proactive schedule generated in (R_i, S_i) , $i = 1, 2, 3$, with that of the determined schedule produced through QIG. As Scenario 1 only considers the stochastic machine breakdown, for which

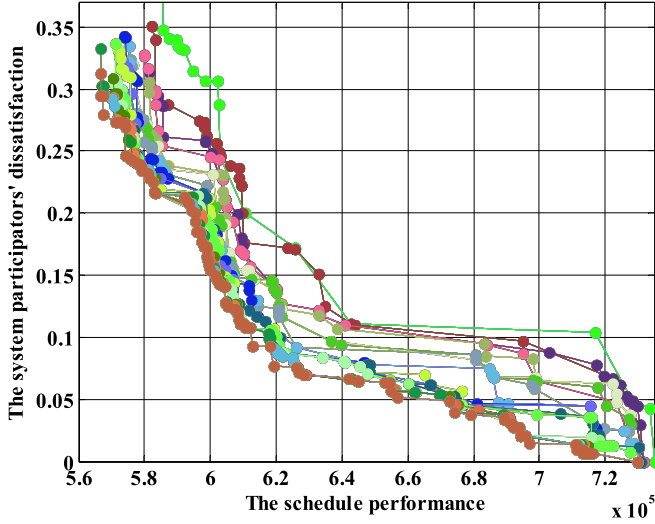


Fig. 5. Evolution of Pareto fronts for HQIG.

the surrogate measures of robustness and stability are designed, we perform experiments under Scenario 1 to evaluate the effectiveness of (R_i, S_i) , $i = 1, 2, 3$, in obtaining the baseline schedule.

First, we apply HQIG to solve the 90 benchmark instances subject to the stochastic machine breakdown of types 1 and 2. (R_i, S_i) , $i = 1, 2, 3$, can produce three Pareto fronts. Then, another simulation-based approach is applied. We randomly generate a sample of machine breakdowns using the gamma breakdown/repair distribution. This sample serves as a proactive schedule in each Pareto front. After that, we right-shift those affected operations until the machine is available again. The entire process is replicated 100 times. We then obtain two average values for the realized f_1 and f_2 , which are used as approximations for $\mathbb{E}(f_1)$ and $\mathbb{E}(f_2)$ for evaluation purposes. For each approach, we generate the baseline schedules under (R_i, S_i) , $i = 1, 2, 3$, and QIG. Thus, we have run a total of $90 \times 2 \times 100 = 18\,000$ simulation replications.

Note that the output of the proactive scheduling in the (R_i, S_i) , $i = 1, 2, 3$, criteria contains a set of nondominated solutions. For the purpose of further analysis in the reactive stage, we apply weighted linear combination to determine the optimal solution from the set. An upper bound of f_1 first needs to be calculated for scalarization.

Denote the maximum processing time among all operations as $p^{UB} = \max_{1 \leq i \leq m, 1 \leq j \leq n_0} \{p_{ij}\}$. The upper bound for the completion time of the first job on machine m can be calculated as mp^{UB} . The upper bound for the completion time of the second job is $(m+1)p^{UB}$, and so on so forth. Finally, the upper bound for f_1 is $(n_0 m + 0.5 n_0 (n_0 - 1))p^{UB}$. We use the following linear combination to evaluate a baseline schedule: $\frac{1}{100} \sum_{num=1}^{100} (0.2 f_1 ((n_0 m + 0.5 n_0 (n_0 - 1))p^{UB})^{-1} + 0.8 f_2)_{num}$, where num denotes the numth replication in the simulation, and we assume the stability is more of a concern than $\sum C_{mj}$.

The means and standard deviations of the four methods under various instance sizes and disruption types are shown in Fig. 6. The numbers in the horizontal axis, “1–9,” represent TA20*5,

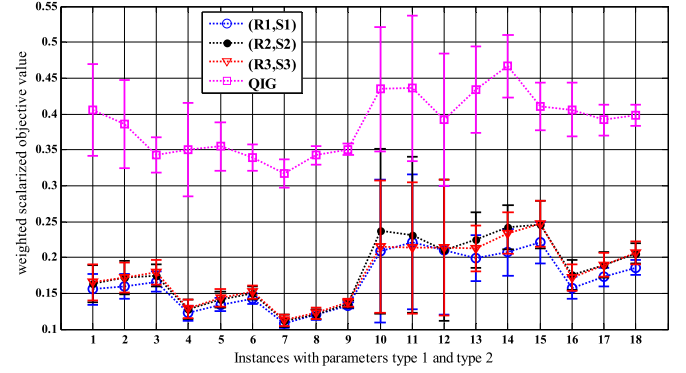


Fig. 6. Means and standard deviations of 4 methods on weighted scalarized objective under different instance size and disruption type.

TABLE IV
STATISTICAL RESULTS OF FRIEDMAN TEST FOR THREE RELATED SAMPLES ON WEIGHTED SCALARIZED OBJECTIVE

| Methods | Mean | Stdev | Median | Rank | Sample size | χ^2 | Degree of freedom | P-value |
|--------------|------|-------|--------|------|-------------|----------|-------------------|---------|
| (R_1, S_1) | 0.17 | 0.06 | 0.16 | 1.63 | 18 000 | 34526.03 | 3 | 0.000 |
| (R_2, S_2) | 0.18 | 0.06 | 0.17 | 2.15 | | | | |
| (R_3, S_3) | 0.18 | 0.06 | 0.17 | 2.22 | | | | |
| QIG | 0.39 | 0.07 | 0.38 | 4.00 | | | | |

Stdev: Standard Deviation.

TA20*10, TA20*20, TA50*5, TA50*10, TA50*20, TA100*5, TA100*10, and TA100*20 of type 1. Each group contains ten instances. And numbers “10–18” represent the same nine groups of the instances of type 2. From Fig. 6, we can see that for the problems of the same size, when disruptions from type 1 to type 2 become more severe, the average objective value increases for each baseline schedule. However, for each objective function, as the problem size increases, no obvious trend can be observed because the upper bounds for different problems are different. In addition, the average objective values based on the (R_i, S_i) , $i = 1, 2, 3$, criteria have significant improvement in comparison with that of the baseline. The objective value can be reduced by more than 50% for all testing instances with either disruption type, by incorporating the uncertainty into the generation of the baseline schedules. What is more, the standard deviations derived by the four approaches are generally smaller than those derived by QIG, except for cases 10, 11, and 12. Thus, it can be concluded that the proactive approaches are generally more stable than the deterministic scheduling.

The variation trends of the weighted scalarized objective value are very close among (R_i, S_i) , $i = 1, 2, 3$. However, these trends are apparently different from that of QIG. In order to explain this statistically, we ran the Friedman test on the 18 000-size sample. The results of the Friedman test are shown in Table IV. The p -value equals 0.000 which is less than 0.1% indicates that the test results are acceptable at a 0.1% significance level. All three important parameters (mean, median and standard deviation) of (R_i, S_i) , $i = 1, 2, 3$, are significantly better than those of QIG. Based on the Rank values, that is, 1.63 of (R_1, S_1) , 2.15 of (R_2, S_2) , 2.22 of (R_3, S_3) , and 4.00 of QIG,

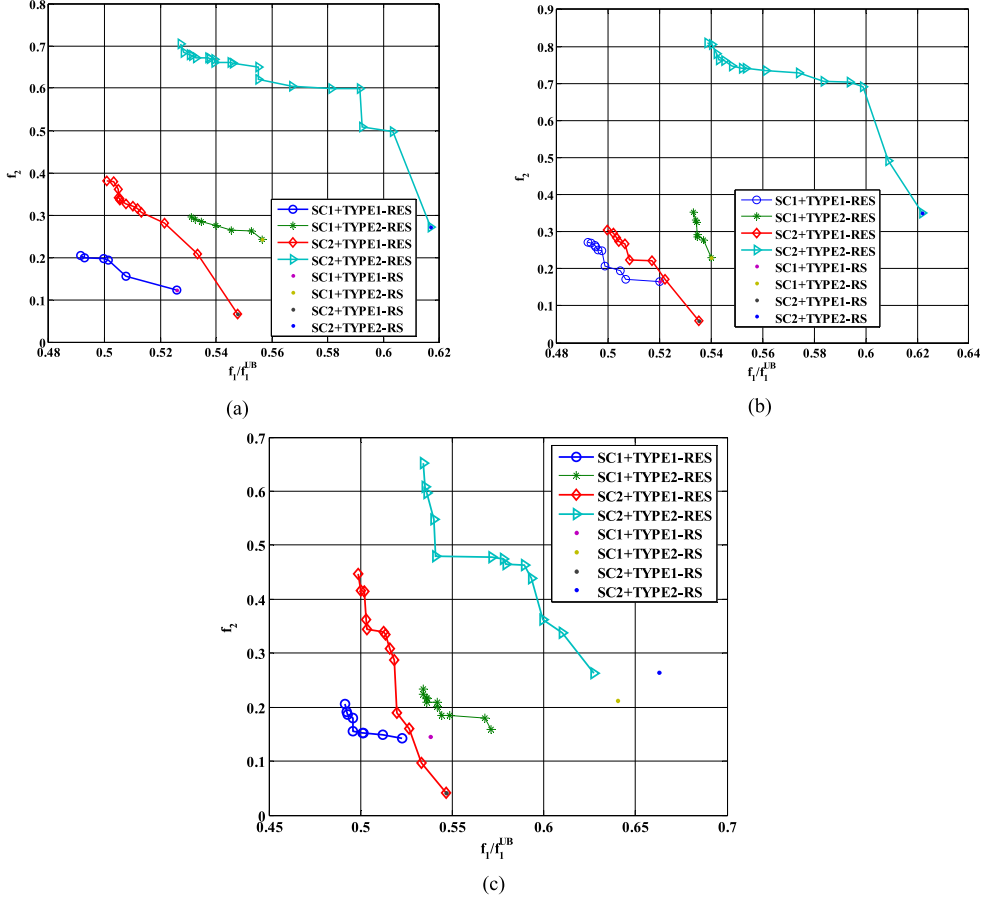


Fig. 7. Comparison of reactions to Scenarios 1 and 2 of disruptions types 1 and 2. (a) Results under (R_1, S_1) . (b) Results under (R_2, S_2) . (c) Results under (R_3, S_3) .

we can conclude that the proactive scheduling approaches perform significantly better than the determined scheduling. The difference between (R_1, S_1) and (R_i, S_i) , $i = 1, 2, 3$ is not significant, whereas no difference is observed between (R_2, S_2) and (R_3, S_3) . If the running time is a primary concern for decision makers, (R_1, S_1) will be recommended as the robustness and stability measures; otherwise, (R_2, S_2) will be recommended.

3) Analysis of the Reactive Methods: In this section, we focus on a class of more complicated problems where various disruptions are handled simultaneously. The effectiveness of different reactions will also be evaluated accordingly. Recall in Scenario 2 of the testing environment, machine breakdown, new arriving jobs, and delay in job availability for $\tau \cdot (\frac{1}{mn_0} \sum_{i=1}^m \sum_{j=1}^{n_0} p_{ij})$ are considered and simulated at the beginning of the planning horizon. For each combination of the scenario and parameter type, we apply the “RS” method by right-shifting the unfinished operations of jobs affected by the machine breakdown, adding the new arriving jobs at the end of the current queue, and inserting the delayed job into the available position.

For the purpose of comparisons, we reschedule all original and new jobs through HQIG, and repair the schedule if the delayed job is arranged before it is available using insertion in the RS. This rescheduling from the scratch approach is defined

as “RES.” The number of jobs in HQIG is updated from n_0 to $n_0 + n_N$. The population size is also updated accordingly, while the other parameters remain the same. For both RS and RES methods, we tested a total of $90 \times 2 \times 2 = 360$ problem instances. For each instance, ten replications are simulated. One representative result of TA005 is shown in Fig. 7, where “SC” means scenario, RS and RES are the two reactions. The other results are similar, and therefore are not reported here.

From Fig. 7, we can observe that the RS often returns a solution with good stability, while the RES can effectively balance schedule performance with stability. However, Fig. 7(c) shows that the stability obtained from the RS is not necessarily optimal. Higher level disruption parameters (type 2) always lead to larger dissatisfaction of the system participators than lower level (type 1) parameters do. More alternative solutions are obtained with high quality. We can conclude that the proactive-reactive approach subject to both stochastic and dynamic disruptions is more appropriate for the manufacturing/service systems than the proactive scheduling approach.

VI. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this paper, we study a permutation flowshop scheduling problem with the objective of minimizing robustness and stability. We develop a proactive-reactive approach to cope with

stochastic and dynamic disruptions. Uncertain machine breakdown is tackled in the proactive stage by generating a robust and stable baseline schedule. The robustness measure is determined by the schedule performance, and the stability measure is determined by the behaviors of various participants using the prospect theory. Stochastic and dynamic disruptions are simultaneously considered in the reactive stage. In order to deal with the conflict between robustness and stability, we develop an algorithm hybridization strategy that is able to effectively search the Pareto front in each stage. The effectiveness of the proactive-reactive approach and algorithm hybridization strategy are demonstrated by extensive computational experiments on the Taillard flowshop benchmark instances.

Our research provides a contribution to the robustness and stability scheduling arena. Future research can be examined in the following two aspects. First, the proactive-reactive approach and the algorithm hybridization strategy can be adapted and tested in more practical flowshop problem settings, such as the distributed environment, the assemble line, and the no-idle permutation. Second, we apply the prospect theory in this paper to model the system participants' behaviors, but it might be more interesting to design behavioral lab experiments to study human reactions toward disruptions in the manufacturing/service systems. Behavioral operations management attracts significant intensions from researchers and practitioners in recent years, and we believe the parameters derived through this way would be more accurate.

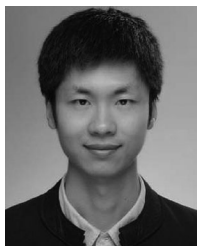
ACKNOWLEDGMENT

The authors would like to thank the Editor-in-Chief, the Associate Editor, and anonymous referees for their helpful comments and suggestions, which greatly helped improve the previous version of this paper.

REFERENCES

- [1] X. Li and M. Li, "Multiobjective local search algorithm-based decomposition for multiobjective permutation flow shop scheduling problem," *IEEE Trans. Eng. Manage.*, vol. 62, no. 4, pp. 544–557, Nov. 2015.
- [2] B. Addis, G. Carello, A. Grosso, and E. Tanfani, "Operating room scheduling and rescheduling: A rolling horizon approach," *Flexible Serv. Manuf. J.*, vol. 28, nos. 1/2, pp. 206–232, 2016.
- [3] Q. Pan and R. Ruiz, "An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem," *Omega*, vol. 44, pp. 41–50, 2014.
- [4] M. M. Yenisey and B. Yagmahan, "Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends," *Omega*, vol. 45, pp. 119–135, 2014.
- [5] Y. H. Dong and J. Jang, "Production rescheduling for machine breakdown at a job shop," *Int. J. Prod. Res.*, vol. 50, no. 10, pp. 2681–2691, 2012.
- [6] K. Stuart and E. Kozan, "Reactive scheduling model for the operating theatre," *Flexible Serv. Manuf. J.*, vol. 24, no. 4, pp. 400–421, 2012.
- [7] K. Lee, L. Lei, M. Pinedo, and S. Wang, "Operations scheduling with multiple resources and transportation considerations," *Int. J. Prod. Res.*, vol. 51, nos. 23/24, pp. 7071–7090, 2013.
- [8] S. Goren and I. Sabuncuoglu, "Robustness and stability measures for scheduling: Single-machine environment," *IIE Trans.*, vol. 40, no. 1, pp. 66–83, 2008.
- [9] F. Gino and G. Pisano, "Toward a theory of behavioral operations," *Manuf. Serv. Oper. Manage.*, vol. 10, no. 4, pp. 676–691, 2008.
- [10] D. Kahneman and A. Tversky, *Choices, Values, and Frames*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [11] C. De Snoo, W. Van Wezel, J. C. Wortmann, and G. J. Gaalman, "Coordination activities of human planners during rescheduling: Case analysis and event handling procedure," *Int. J. Prod. Res.*, vol. 49, no. 7, pp. 2101–2122, 2011.
- [12] V. J. Leon, S. D. Wu, and R. H. Storer, "Robustness measures and robust scheduling for job shops," *IIE Trans.*, vol. 26, no. 5, pp. 32–43, 1994.
- [13] S. V. Mehta and R. M. Uzsoy, "Predictable scheduling of a job shop subject to breakdowns," *IEEE Trans. Robot. Autom.*, vol. 14, no. 3, pp. 365–378, Jun. 1998.
- [14] R. O'Donovan, R. Uzsoy, and K. N. McKay, "Predictable scheduling of a single machine with breakdowns and sensitive jobs," *Int. J. Prod. Res.*, vol. 37, no. 18, pp. 4217–4233, 1999.
- [15] M. T. Jensen, "Generating robust and flexible job shop schedules using genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, pp. 275–288, Jun. 2003.
- [16] M. Sevaux and K. S. Rensen, "A genetic algorithm for robust schedules in a one-machine environment with ready times and due dates," *Quart. J. Belgian, French, Italian Oper. Res. Soc.*, vol. 2, no. 2, pp. 129–147, 2004.
- [17] L. Liu, H. Gu, and Y. Xi, "Robust and stable scheduling of a single machine with random machine breakdowns," *Int. J. Adv. Manuf. Technol.*, vol. 31, nos. 7/8, pp. 645–654, 2007.
- [18] X. Q. Zuo, H. Mo, and J. Wu, "A robust scheduling method based on a multi-objective immune algorithm," *Inf. Sci.*, vol. 179, no. 19, pp. 3359–3369, 2009.
- [19] S. Goren and I. Sabuncuoglu, "Optimization of schedule robustness and stability under random machine breakdowns and processing time variability," *IIE Trans.*, vol. 42, no. 3, pp. 203–220, 2010.
- [20] T. Chaari, S. Chaabane, T. Loukil, and D. Trentesaux, "A genetic algorithm for robust hybrid flow shop scheduling," *Int. J. Comput. Integr. Manuf.*, vol. 24, no. 9, pp. 821–833, 2011.
- [21] S. Goren, I. Sabuncuoglu, and U. Koc, "Optimization of schedule stability and efficiency under processing time variability and random machine breakdowns in a job shop environment," *Nav. Res. Logist.*, vol. 59, no. 1, pp. 26–38, 2012.
- [22] J. Xiong, L. Xing, and Y. Chen, "Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns," *Int. J. Prod. Econ.*, vol. 141, no. 1, pp. 112–126, 2013.
- [23] D. Rahmani and M. Heydari, "Robust and stable flow shop scheduling with unexpected arrivals of new jobs and uncertain processing times," *J. Manuf. Syst.*, vol. 33, no. 1, pp. 84–92, 2014.
- [24] L. K. Church and R. Uzsoy, "Analysis of periodic and event-driven rescheduling policies in dynamic shops," *Int. J. Comput. Integr. Manuf.*, vol. 5, no. 3, pp. 153–163, 1992.
- [25] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," *Math. Oper. Res.*, vol. 1, no. 2, pp. 117–129, 1976.
- [26] Y. Zhang and X. P. Li, "A quantum-inspired iterated greedy algorithm for permutation flowshops in a collaborative manufacturing environment," *Int. J. Comput. Integr. Manuf.*, vol. 25, no. 10, pp. 924–933, 2012.
- [27] N. G. Hall and C. N. Potts, "Rescheduling for new orders," *Oper. Res.*, vol. 52, no. 3, pp. 440–453, 2004.
- [28] N. G. Hall and C. N. Potts, "Rescheduling for job unavailability," *Oper. Res.*, vol. 58, no. 3, pp. 746–755, 2010.
- [29] B. Li and L. Wang, "A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling," *IEEE Trans. Syst. Man. Cybern. B*, vol. 37, no. 3, pp. 576–591, Jun. 2007.
- [30] F. Liu, J. J. Wang, and D. L. Yang, "Solving single machine scheduling under disruption with discounted costs by quantum-inspired hybrid heuristics," *J. Manuf. Syst.*, vol. 32, no. 4, pp. 715–723, 2013.
- [31] K. Deb, A. Pratap, S. Agarwal, and T. A. M. T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [32] K. Han and J. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002.
- [33] E. Taillard, "Benchmarks for basic scheduling problems," *Eur. J. Oper. Res.*, vol. 64, no. 2, pp. 278–285, 1993.
- [34] A. C. Nearchou, "Scheduling with controllable processing times and compression costs using population-based heuristics," *Int. J. Prod. Res.*, vol. 48, no. 23, pp. 7043–7062, 2010.
- [35] E. Erdem, X. Qu, and J. Shi, "Rescheduling of elective patients upon the arrival of emergency patients," *Dec. Support Syst.*, vol. 54, no. 1, pp. 551–563, 2012.

- [36] M. Heydari and A. Soudi, "Predictive / reactive planning and scheduling of a surgical suite with emergency patient arrival," *J. Med. Syst.*, vol. 40, no. 1, pp. 1–9, 2016.
- [37] Y. Yin, K. E. Steckel, M. Swink, and I. Kaku, "Lessons from seru production on manufacturing competitively in a high cost environment," *J. Oper. Manage.*, vol. 49–51, pp. 67–76, 2017. [Online]. Available: <http://dx.doi.org/10.1016/j.jom.2017.01.003>
- [38] D. Tang, M. Dai, M. Salido, and A. Giret, "Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization," *Comput. Ind.*, vol. 81, pp. 82–95, 2016.
- [39] J. Li, Q. Pan, and K. Mao, "A hybrid fruit fly optimization algorithm for the realistic hybrid flowshop rescheduling problem in steelmaking systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 932–949, Apr. 2016.
- [40] E. Ahmadi, M. Zandieh, M. Farrokh, and S. M. Emami, "A multi-objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms," *Comput. Oper. Res.*, vol. 73, pp. 56–66, 2016.
- [41] H. Seidgar, M. Zandieh, H. Fazlollahab, and I. Mahdavi, "Simulated imperialist competitive algorithm in two-stage assembly flow shop with machine breakdowns and preventive maintenance," *Proc. Inst. Mech. Eng. B., J. Eng. Manuf.*, vol. 230, no. 5, pp. 934–953, 2016.
- [42] J. Framinan and P. Perez-Gonzalez, "On heuristic solutions for the stochastic flowshop scheduling problem," *Eur. J. Oper. Res.*, vol. 246, no. 2, pp. 413–420, 2015.
- [43] D. Rahmani and R. Ramezani, "A stable reactive approach in dynamic flexible flow shop scheduling with unexpected disruptions: A case study," *Comput. Ind. Eng.*, vol. 98, pp. 360–372, 2016.
- [44] X. Li and M. Yin, "A hybrid cuckoo search via Lévy flights for the permutation flow shop scheduling problem," *Int. J. Prod. Res.*, vol. 51, no. 16, pp. 4732–4754, 2013.
- [45] X. Li and M. Yin, "An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure," *Adv. Eng. Softw.*, vol. 55, pp. 10–31, 2013.
- [46] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on de-composition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [47] P. C. Chang, S. H. Chen, Q. Zhang, and J. L. Li, "MOEA/D for flowshop scheduling problems," in *Proc. IEEE Congr. Evol. Comput.*, 2008, pp. 1433–1438.
- [48] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 204–223, Apr. 2003.



Feng Liu received the Ph.D. degree in management science and engineering from the Dalian University of Technology, Dalian, China, in 2014.

He is currently an Assistant Professor with the School of Management Science and Engineering, Dongbei University of Finance and Economics, Dalian. His research interests primarily cover seru production planning and scheduling. He has published extensively in journals such as *Omega*, *Computers & Operations Research*, the *International Journal of Logistics Management*, the *Journal of Manufacturing Systems*, the *Journal of the Operational Research Society*, and *Knowledge-Based Systems*, etc.

Dr. Liu's research is supported by the National Science Foundation of China.



Shengbin Wang received the Ph.D. degree in supply chain management from Rutgers University, New Brunswick, NJ, USA, in 2014.

He is currently an Assistant Professor of supply chain management with the School of Business and Economics, North Carolina Agricultural and Technical State University, Greensboro, NC, USA. His primary research interests include supply chain operations modeling and algorithm design, humanitarian and crisis operations management, enterprise resource planning, big data analytics in supply chain, and transportation management.



Yuan Hong (M'13) received the Ph.D. degree in information technology from Rutgers University, New Brunswick, NJ, USA, in 2013.

He is currently an Assistant Professor with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA. His research interests primarily include intersection of privacy, security, optimization, and data mining.



Xiaohang Yue received the Ph.D. degree in operations management from the University of Texas at Dallas, Richardson, TX, USA, in 2002.

He is currently a Professor with the Sheldon B. Lubar School of Business, University of Wisconsin–Milwaukee, Milwaukee, WI, USA. His research interests primarily include supply chain and logistics systems management, and industrial manufacturing systems management. He has published extensively in leading journals such as *Operations Research*, *Decision Sciences*, *IIE Transactions*, *Naval Research Logistics*, *Production and Operations Management*, the *European Journal of Operational Research*, *Omega*, etc.