Bad-scenario-set Robust Optimization Framework With Two Objectives for Uncertain Scheduling Systems

Bing Wang, Xuedong Xia, Hexia Meng, and Tao Li, *Senior Member, IEEE*

***Abstract*—This paper proposes a robust optimization framework generally for scheduling systems subject to uncertain input data, which is described by discrete scenarios. The goal of robust optimization is to hedge against the risk of system performance degradation on a set of bad scenarios while maintaining an excellent expected system performance. The robustness is evaluated by a penalty function on the bad-scenario set. The bad-scenario set is identified for current solution by a threshold, which is restricted on a reasonable-value interval. The robust optimization framework is formulated by an optimization problem with two conflicting objectives. One objective is to minimize the reasonable value of threshold, and another is to minimize the measured penalty on the bad-scenario set. An approximate solution framework with two dependent stages is developed to surrogate the bi-objective robust optimization problem. The approximation degree of the surrogate framework is analyzed. Finally, the proposed bad-scenario-set robust optimization framework is applied to a scenario job-shop scheduling system. An extensive computational experiment was conducted to demonstrate the effectiveness and the approximation degree of the framework. The computational results testified that the robust optimization framework can provide multiple selections of robust solutions for the decision maker. The robust scheduling framework studied in this paper can provide a unique paradigm for formulating and solving robust discrete optimization problems.**

***Index Terms*—Approximate solution, bad-scenario set, bi-objective problem, job shop, robust optimization framework.**

I. Introduction

IN scheduling environments where uncertainty is a major issue, robustness is a new scheduling performance measure, which is particularly concerned for robust scheduling [1], [2]. Input data uncertainty, such as variation in processing times, is frequently considered in the research of robust scheduling. Difficulties with input data uncertainty are typically dealt with by proactive scheduling policy, which tries to ensure that the preventive schedules maintain a high level of system performance [2]. Scenario approach is a significant modeling tool of input data uncertainty to develop proactive scheduling. Mulvey *et al.* [3] proposed a scenario-based representation and analysis of uncertainty rather than using stochastic models in their pioneering work. They developed the scenario-based robust optimization to handle the tradeoff associated with solution robustness and model robustness for several real-world applications. Up to now, robust optimization methodology has been applied to develop proactive scheduling in wide discrete optimization fields [4]−[7].

Li *et al.* [5] classified the scenario-based robust scheduling formulations into two groups: the scenario-based stochastic programming formulation [3], [8], [9] and the robust counterpart optimization formulation [10]. The scenario-based stochastic programming aims at optimizing expected system performance and tries to achieve optimality of scheduling system in the average sense. As an alternative to the scenario-based stochastic programming formulation, the robust counterpart optimization, which comes from mathematical programming formulations developed by Soyster [11], avoids the shortcomings of the stochastic programming formulation. The pioneering works by Ben-Tal and Nemirovski [12] and Bertsimas and Sim [13] extended the framework of robust counterpart optimization, and developed sophisticated solution techniques.

In the face of uncertainty, the attitude toward the risk of performance degradation is a particularly concerned issue when assessing performance robustness. Following the scenario-based formulation, most studies have evaluated performance robustness in different ways. A typical pessimistic attitude to risk usually focuses on a sole worst-case scenario [14]−[16]. Daniels and Kouvelis [14], [15] evaluated schedule robustness by focusing on the worst-case performance among all possible scenarios to hedge against processing time variability. Artigues *et al.* [16] used the same way to hedge against the performance degradation on the worst-case contingency by evaluating a given ordered group assignment. However, the worst-case scenario formulation would result in an extremely conservative decision due to without considering any optimality on other scenarios.

System performance degradation may occur not only in the worst-case scenario but among a set of bad scenarios. In the face of the risk of performance degradation, it is reasonable to concern more bad scenarios. Thus it is necessary to give a standard for explicitly differentiating standard performances from substandard (bad) ones across scenarios. Given a target

Manuscript received January 23, 2015; accepted October 23, 2015. This work was supported by National Natural Science Foundation of China (60874076, 61273182, 61273040), the Shanghai Rising-Star Program (15QA1402000). Recommended by Associate Editor Qianchuan Zhao.

Citation: B. Wang, X. D. Xia, H. X. Meng, and T. Li,"Bad-scenario-set robust optimization framework with two objectives for uncertain scheduling systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 143−153, Jan. 2017.

B. Wang, X. D. Xia, H. X. Meng and T. Li. are with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200072, China (e-mail: susanbwang@shu.edu.cn; hytcxiaxdong@163.com; menghx2014@163.com; sixumuzi@shu.edu.cn).

Digital Object Identifier 10.1109/JAS.2017.7510352

performance, Daniels *et al.* [17] developed the beta-robust scheduling model for single-machine scheduling with uncertain processing times. The beta-robustness objective is to maximize the likelihood of that a schedule yields actual performance no worse than the given target performance. The beta-robust scheduling formulation restricts its attention on good performances (better than the given standard performance) and hedge against the risk of performance degradation through enlarging the likelihood of achieving excellent performances as possible as they can. Other than that, we consider another paradigm to realize the robust scheduling goal. We restrict our attention on poor performances on bad scenarios, which are identified still by a given standard performance. Wang *et al.* [18] proposed the bad-scenario-set robust scheduling model based on a standard performance, whose objective is to minimize the total penalties on bad scenarios due to their worse performances than the given standard performance. The beta-robust scheduling model and the bad-scenario-set scheduling model can provide less conservative robust solution than the worst-case scenario model.

In either the beta-robust scheduling model or the bad-scenario-set scheduling model, a given threshold named standard performance by Wang *et al.* [18] and target performance by Daniels *et al.* [17] plays an important role. In production reality, a reasonable value of threshold can be given based on empirical data or provided in other ways. That will result in a one-stage robust scheduling, exactly like what Daniels and Carrillo [17] had done. The beta-robust scheduling [17] did not provide a method to determine the required target performance. Wu *et al.* [19] proposed a two-stage decision-making structure to handle the single-machine beta-scheduling problem using constraint propagation. They presented a procedure of determining a target performance, which was generated by combining average performance and variance. Such a two-stage approach is consistent with the robust decision-making formulations presented by Kouvelis *et al.* [20] and Assavapokee *et al.* [21]. Wang *et al.* [18] also did not provide a procedure to determine reasonable values of threshold although they noticed it is an important issue for the bad-scenario-set scheduling model.

We consider a more complicated situation than that in [18] in this paper. Assume that the reasonable value of threshold is not given in advance. We have to determine a reasonable value of threshold in advance and accordingly generate a corresponding robust schedule. This is a two-stage decision-making structure. Regarding both the value of threshold and the robust solution as decision variables, a robust optimization framework is formulated by a problem with two objectives generally for scheduling systems with input data uncertainty. The aim of the problem is at hedging against the risk of performance degradation while keeping excellent traditional expected performance. The robustness is evaluated based on the concept of bad-scenario set [18], which is identified by a reasonable value of threshold. Since reasonable values of threshold are restricted in an interval, two extra optimization problems need to be solved while solving the bi-objective problem. In order to avoid handling three intractable problems, an approximate solution framework with two dependent stages is developed to surrogate the bi-objective problem. A feedback adjusting mechanism is embedded in the approximate framework from the second stage to the first stage. The significance of the approximate solution framework is that it can transfer the procedure of solving three optimization problems into an iteration procedure of approximately solving two approximate problems. The third problem can be handled by the feedback adjusting mechanism in the search of two-stage algorithms.

We discuss the properties of the robust scheduling framework and conclude that the approximation degree of robust solution obtained by the framework is determined by the approximation degree of two solved problems. We apply the robust optimization framework to a scenario job-shop scheduling problem in this paper. The analysis conclusions are testified by the computational results.

II. Bad-scenario-set Robust Scheduling Framework

We consider a general scheduling system to minimize certain performance measure of interest under input data uncertainty. Scenario approach is used to structure uncertain input data. Some notations are given as follows: Given the set of scenarios of uncertain input data denoted by Λ, and the number of scenarios in Λ is denoted by $|\Lambda|$. A scenario $\lambda \in \Lambda$ represents a possible realization of the uncertain input data. Let s represent a feasible solution satisfying all required constraints, and let $C(s, \lambda)$ represent the system performance corresponding to any feasible solution s on any scenario λ. The set of all possible feasible solutions is denoted by S. In the following, we refer to a scheduling problem in such an uncertain scenario environment as the scenario scheduling problem (SSP).

A. Formulating a Robust Scheduling Problem Under a Given Threshold

For the discussed SSP, system performance varies as scenario varies and may be very poor on some scenarios. Our goal to a schedule is to hedge against the risk of system performance degradation, which occurs on a set of bad scenarios. The bad-scenario set is defined based on a given threshold denoted by T. Referred to Wang *et al.* [18], for any feasible solution s, a scenario λ on which system performance is worse than T is regarded as a bad scenario. Generally, given T a value of interest performance for the scenario set Λ, the bad-scenario set based on T is a subset of Λ for a feasible solution $s \in S$ as follows:

$$\Lambda_T(s) = \{\lambda | C(s, \lambda) \geq T, \lambda \in \Lambda\}. \tag{1}$$

The performance value T is actually a threshold of identifying bad scenarios from all possible scenarios. We refer to $\Lambda_T(s)$ as a threshold bad-scenario set (TBS) for the feasible solution s. Thus both the value of T and the solution s affect the TBS $\Lambda_T(s)$. The number of bad scenarios (NBS) will vary as the solution s or the value of T varies.

The value of T can also be regarded as a standard performance. TBS includes all bad scenarios on which system performances of the solution s are substandard to the standard performance T. We intend to generate robust schedules, where substandard performances on TBS will be hedged against as possible as we can. Actually, a solution may incur a penalty if system performances are substandard on some bad scenarios. We consider adding a penalty on the bad scenarios of TBS. The amount of the penalty may be associated with not only the likelihood of generating substandard performances but also the degree of deviation of poor performances from the standard performance. The bad-scenario-set robust scheduling model was proposed to minimize the total penalties incurred by substandard performances across the identified bad scenarios by Wang *et al.* [18]. The effectiveness as well as the robust solution obtained by the bad-scenario-set robust scheduling model is proved to be dependent on the given value of threshold. Therefore, the value of threshold is a significant parameter for the bad-scenario-set robust scheduling model.

The amount of penalty on TBS (PT) is measured based on the value of given threshold as follows:

$$PT(s) = \sum_{\lambda \in \Lambda_T(s)} [C(s,\lambda) - T]^2. \tag{2}$$

The square item in the formula (2) is applied in the penalties in order to intensively hedge against worse scenarios. This will benefit the decision maker's preference of risk averse much more than a linear item. A PT-robust scheduling problem (PSP) is formulated by minimizing the penalty $PT(s)$.

$$\text{(PSP)} \qquad \min_{s \in S} PT(s). \tag{3}$$

Obviously, the PSP is formulated based on the TBS, and the TBS is defined based on a given value of T, i.e., the PSP corresponds to the value of T. In order to differ PSPs for different values of T, we might as well denote the PSP for the value of T by PSP|T . For the PSP, if the value of T is too big, $\Lambda_T(s)$ could be empty for some feasible solutions. In this situation, the penalty $PT(s)$ is zero and the corresponding PSP|T is invalid because the model may lose the ability of differing some elite solutions [18]. In principle, we have Definition 1 as follows:

Definition 1: The value of T is referred to as a reasonable value for the PSP|T if $\Lambda_T(s)$ is always nonempty for any feasible solution $s \in S$. We call the PSP|T as effective model if the value of T is reasonable.

In any case, the effectiveness of PSP|T depends on the value of T. An effective PSP|T model concerns a set of bad scenarios more than the worst-case scenario. It will generate robust solutions with less conservatism than the worst-case scenario problem. Thus the degree of solution conservatism of the PSP|T also depends on the value of T. A reasonable value of T is vital for an effective PSP. To establish the effective PSP|T , a reasonable value of T needs to be determined in advance.

B. Determining Reasonable Values of Threshold

The value of T is used not only as a threshold to identify bad-scenario set but also as a standard performance to measure the PT objective for the discussed SSP. For the discussed SSP herein, what value of T is reasonable depends on system input data. Different values of T represent different decision preferences, and further establish different PSP|T models that generate different robust solutions. In the following, we have to handle the issue regarding how to determine a reasonable value of T.

On one side, a reasonable value of T must guarantee the effectiveness of PSP|T . A reasonable value of T should be located on the common interval of all possible performances that the system is able to achieve for all possible feasible solutions among all possible scenarios. We denote the optimal worst-case performance among all possible scenarios by WC^*, then WC^* is the upper bound of the common interval. If $T > WC^*$, TBS is possibly null for some elite solutions. Therefore, a reasonable value of T should be subject to $T \leq WC^*$, which is a compulsory condition to formulate an effective PSP|T .

On the other side, our scheduling goal is to hedge against the risk of performance degradation among bad scenarios while keeping an excellent expected performance. The reasonable value of T should also help the PSP|T model achieve our robust scheduling goal, i.e., a robust solution of PSP|T could represent a risk-averse preference, just like what Daniels *et al.* [14] declared, the value of threshold should be required to be bigger than the optimal expected performance. We denote the optimal expected-case performance among all possible scenarios by EC^*, and determine that EC^* should be the lower bound of the common interval, i.e., $T \geq EC^*$ is required. In summary, to achieve our robust scheduling goal, the reasonable values of T should be located on the interval $[EC^*, WC^*]$. We name the interval $[EC^*, WC^*]$ as the reasonable-value interval of T. Referring to [18], we present Proposition 1 without proof as follows:

Proposition 1: Any value of T on the reasonable-value interval $[EC^*, WC^*]$ can guarantee that the corresponding PSP|T is effective and the PSP|T model is capable of generating a PT robust solution.

Obviously, the interval $[EC^*, WC^*]$ can provide infinite number of reasonable values of T, which correspond to infinite number of PSP|T models. Each PSP|T model can generate a PT robust solution, which can be represented by $PT(s)$. The PT robust solutions generated by PSP|T models with different reasonable values of T could be regarded as different decision choices provided to the decision maker, which can realize different degrees of tradeoff between the robustness and the optimality. We represent a decision choice by a pair of performances $(T, PT(s,T))$. In the following, we intend to propose a bi-objective optimization problem formulation, which is able to generate all decision choices $(T, PT(s,T))$. The bi-objective optimization problem provides a robust scheduling framework, which can accommodate all PSP|T models for all possible reasonable values.

C. Formulating a Robust Scheduling Framework

Regarding the reasonable value of T and the PT robust solution as decision variables, we can formulate a bi-objective optimization problem. In fact, the PSP|T models involve two objectives. One objective is to minimize the penalty $PT(s)$.

A smaller value of $PT(s)$ should benefit the robustness of scheduling system. Another objective is to minimize the value of T. The reason is that a smaller value of T results that more bad scenarios are concerned in the PSP|T model, and should benefit a better expected performance, i.e., a smaller value of T should benefit the statistical optimality of scheduling system. However, two objectives are conflicting because a smaller value of T leads to the bigger value of $PT(s)$. Therefore, we accommodate all the PSP|T models in a PSP framework (PSPF), which is a bi-objective optimization problem.

$$\text{(PSP|T)} \quad \min \quad PT(s,T) = \sum_{\lambda\in\Lambda_T(s)} [C(s,\lambda)-T]^2 \tag{4}$$

$$\min \quad T \tag{5}$$

$$\text{s.t.} \quad EC^* \le T \le WC^* \tag{6}$$

$$\text{(ECP)} \quad EC^* = \min_{s\in S} \sum_{\lambda\in\Lambda} C(s,\lambda) \tag{7}$$

$$\text{(WCP)} \quad WC^* = \min_{s\in S} \max_{\lambda\in\Lambda} C(s,\lambda). \tag{8}$$

See the expressions (4)−(8), where the expected-case problem (ECP) is referred to the model with the criterion of expected performance across all possible scenarios of Λ, the worst-case problem (WCP) is referred to the model with the criterion of worst-case scenario performance among all possible scenarios of Λ, and the PSP|T is the model with the objective $PT(s)$ for the value of T. Thus the PSPF is actually a scheduling framework implicitly involving three problems: PSP|Ts, the ECP, and the WCP. Each candidate value of T is subject to the (6), where EC^* and WC^* are determined by optimally solving the ECP and the WCP. The ECP and the WCP are extra two problems need to handle. They are at least the same intractable as the original SSP because they can be reduced to the SSP when $|\Lambda| = 1$.

It is worth recalling that a solution of bi-objective problem is said to be efficient (or non-dominated) if it is impossible to improve it with respect to two objectives without violating any constraint. The solution quality and the solution diversity of efficient solutions are two aspects of evaluating the efficient solutions of bi-objective problem. For more studies of bi-objective scheduling problems, one can refer to the literature [22]−[24].

For the PSPF, equilibrium of two objectives could be achieved at each efficient solution $(T, PT(s,T))$, which consists of a reasonable value of T and the corresponding PT performance. We might as well denote the PT solution of PSP|T by $s(T)$, denote the PT objective of $s(T)$ by $PT(s(T))$ instead of $PT(s,T)$ of (4), and denote an efficient solution of PSPF by $(T, PT(s(T)))$. We have Proposition 2 as follows.

Proposition 2: The PSPF must be effective on the reasonable-value interval $[EC^*, WC^*]$ and it is capable of generating a set of efficient solutions.

Proof: Since each value of T subject to the inequality (6) is on the reasonable-value interval $[EC^*, WC^*]$, the PSPF must be effective due to that each PSP|T is effective according to Proposition 1.

Let $s^*(T)$ be the optimal robust solution of PSP|T, i.e., $\{s^*(T) = \arg\min_{s(T)\in S} PT(s(T))\}$. We denote the objective of $s^*(T)$ by $PT(s^*(T))$. Then T and $PT(s^*(T))$ are bi-objective performances of PSP|T for the solution $s^*(T)$. $PT(s^*(T))$ must be the smallest value for the value of T and is impossibly improved further without increasing the value of T. Conversely, the value of T must be the smallest value for the $PT(s^*(T))$ because a smaller value than $PT(s^*(T))$ has to correspond to a bigger value of T. Thus $s^*(T)$ is an efficient solution of PSPF, and $(T, PT(s^*(T)))$ is an efficient solution of PSPF. That is, for each value of T subject to the inequality (6), an efficient solution of PSPF can be generated. As the value of T varies from EC^* to WC^*, there exists a set of reasonable values of T, correspondingly a set of efficient solutions of PSPF can be generated. ■

The PSPF is a framework consisting of a set of PSP|Ts for different values of T. The PSPF is unconditionally effective for any values of T on the reasonable-value interval. To solve the PSPF, the prerequisite is to determine the reasonable-value interval $[EC^*, WC^*]$. It seems that the ECP and the WCP should be solved optimally in advance of solving the PSPF. However, each one of three problems is NP-hard only if the deterministic SSP is NP-hard. All of them will be intractable and exact solution algorithms will be computationally expensive for them.

Fortunately, the PSPF is a special bi-objective problem, in which the objectives (4) and (5) are not aggressive for each other though they are conflicting. The reasonable value of T should be determined before a PT robust solution is solved. The PT robust solution is actually a function of the value of T. Therefore, the PSPF can actually reduce to solving PSP|Ts for different values of T. Therefore, we plan to develop an approximate solution framework of PSPF to generate a set of efficient solutions for the PSPF. The approximate framework is presented and analyzed in the following.

III. Approximate Solution Framework for the PSPF

A. Formulating a Surrogate Framework

The aim of the solution framework of PSPF is to generate a set of efficient solutions of PSPF. We plan to provide an approximate solution framework based on a surrogate model of PSPF. For an efficient solution $(T, PT(s(T)))$ of PSPF, $PT(s(T))$ is actually measured depending on the value of T according to(2). Each efficient solution can be sequentially determined in a two-stage surrogate decision-making mechanism. The first stage is to determine the first reasonable value of T, and the second stage is to solve a series of PSP|Ts formulated by (4) with respect to the first value of T. Thus a surrogate model is proposed instead of the PSPF to determine a set of reasonable values of T without optimally solving both the ECP and the WCP. We reformulate the PSPF by a surrogate framework with two-stage approximate problems (SFTAP) as follows.

The first-stage approximate problem:

$$\text{(ECP}^{\sim}\text{)} \quad EC^{\sim} \approx \min_{s\in S} EC(s) = \sum_{\lambda\in\Lambda} C(s,\lambda)). \tag{9}$$

The second-stage approximate problem:

$$(\mathrm{PSP}^{\sim}) \quad \min_{s(T)\in S} PT(s(T)) = \sum_{\lambda\in\Lambda_T(s)} [C(s,\lambda)-T]^2 \tag{10}$$

$$T = \beta \cdot EC^{\sim} \tag{11}$$

where $\beta = \beta_1, \beta_2, \ldots, \beta_x$

$$\beta_1 = 1; \beta_{k+1} = \beta_k + \Delta\beta; \Delta\beta > 0$$
$$k = 1, 2, \ldots, x-1$$
$$\Lambda_T(s) \text{ is not empty.} \tag{12}$$

In the first stage of SFTAP, the problem ECP$^{\sim}$ is to approximately solve the ECP to obtain an approximate value of EC^*, denoted by $EC^{\sim}$. In the second stage, the problem PSP$^{\sim}$ is to approximately solve a set of PSP|Ts to get a serial of $PT(s(T))$ under a serial of values of T. The serial of efficient solutions $(T, PT(s(T)))$ constitute the set of efficient solutions (SES) for the PSPF. In the SFTAP, the expression (9), (11) and the constraint (12) are used to surrogate the expressions (6)−(8) respectively. Seen from (11), the set values of T is provided by $EC^{\sim}$ and $\Delta\beta$. The value of $EC^{\sim}$ is the first value of T in the SES. The parameter $\Delta\beta$ is used to locate specific reasonable values of T within the interval $[EC^*, WC^*]$ for the SES. In order to discuss the surrogate framework SFTAP, we give another definition as follows:

Definition 2: We call the SFTAP an effective surrogate framework for the PSPF only if the SFTAP is capable of generating at least one efficient solution of PSPF for the SES.

The reason that we use the approximate problems to surrogate the PSPF is that the following analysis is given based on considering approximate (heuristic) algorithms to solve the ECP$^{\sim}$ and the PSP$^{\sim}$. Moreover, we will also discuss the situation of approximate solutions approaching exact solutions. We might as well consider iterative search structures to provide solution algorithms for two stages of SFTAP. We denote two approximate solution algorithms respectively for the ECP$^{\sim}$ and the PSP$^{\sim}$ by algorithms A1 and A2. In the situation that A1 and A2 solve the ECP$^{\sim}$ and the PSP$^{\sim}$ approximately, the effectiveness of SFTAP for the PSPF probably can not be guaranteed. In fact, the SFTAP may be ineffective if $EC^{\sim}$ is a rather poor approximation. In order to guarantee the effectiveness of SFTAP, a feedback adjusting mechanism is designed which is embedded in the surrogate framework from the first stage to the second stage. The solution framework of SFTAP is illustrated in Fig. 1.

B. Handling the Approximate Problem ECP$^{\sim}$ *in the First Stage*

In the SFTAP, $EC^{\sim}$ is a baseline to provide a family of reasonable values of T and it is vital for the effectiveness of SFTAP. An eligible $EC^{\sim}$ that guarantees the effectiveness of the SFTAP should satisfy the relationship $EC^{\sim} \le WC^*$. Thus we have Proposition 3 as follows.

Proposition 3: The SFTAP is effective for the PSPF only if $EC^{\sim}$ is eligible.

Proof : According to (11), when $\beta = 1$, the first value of T provided by the SFTAP is exactly $EC^{\sim}$. If $EC^{\sim}$ is eligible, it satisfies $EC^{\sim} \le WC^*$. Since $EC^{\sim}$ is obtained through solving the ECP$^{\sim}$, $EC^{\sim}$ must be an upper bound of EC^*, i.e., $EC^{\sim} \ge EC^*$. Therefore, the value of $EC^{\sim}$ is a reasonable value of T. According to Proposition 1, the corresponding PSP|T is effective and it can generate a PT robust solution, which exactly constitutes an efficient solution of PSPF. According to Definition 2, the SFTAP is effective for the PSPF. ■

Seen from Proposition 3, the degree of approximation of $EC^{\sim}$ is vital for the effectiveness of SFTAP. How can we ensure an eligible $EC^{\sim}$? If $EC^{\sim}$ is obtained through solving the ECP$^{\sim}$ by use of an approximate algorithm A1, the degree of approximation of $EC^{\sim}$ depends on the A1. Suppose that the search structure of A1 is given, the solution quality of the ECP$^{\sim}$ depends on the specific setting of algorithm parameters including the termination condition of A1. We name Termination 1 as the termination condition of A1. We might as well focus on Termination 1 to affect the degree of approximation of $EC^{\sim}$ while supposing that other algorithm parameters of A1 are already appropriately set by tuning. Therefore, here Termination 1 is the only parameter of A1, which is used to adjust the degree of approximation of $EC^{\sim}$, see Fig. 1.

C. Handling the Approximate Problem PSP$^{\sim}$ *in the Second Stage*

Observe the whole solution process of SFTAP, we see that the WCP is actually not handled. Then how can the SFTAP ensure $EC^{\sim} \le WC^*$? In fact, a checking segment undertakes this mission in the search algorithm of the second stage. The aim of the second stage is to solve a family of PSPs by iteratively performing A2. Suppose that a search algorithm is adopted as an alternative of A2, once $\Lambda_T(s)$ is identified, the SFTAP checks $\Lambda_T(s)$ to testify if $\Lambda_T(s)$ is empty or not. We name this segment of A2 as the checking segment, which indicates different situations corresponding to specific value of β. If null TBS occurs in a checking segment at $\beta = 1$, it demonstrates that $EC^{\sim}$ is ineligible. In order to guarantee the effectiveness of SFTAP, a feedback adjusting mechanism (shaded segments illustrated in Fig. 1) is embedded in the SFTAP from A2 to A1. Once null TBS occurs in A2 at $\beta = 1$, the feedback adjusting mechanism is started up, i.e., A2 stops temporarily, and the algorithm returns back to A1 to adjust Termination 1 and solve the ECP$^{\sim}$ once again by use of the renewed Termination 1 until an eligible $EC^{\sim}$ is obtained. The feedback adjusting mechanism is actually a guarantee for an eligible value of $EC^{\sim}$. As long as the value of $EC^{\sim}$ obtained in the first stage is eligible, the feedback adjusting mechanism would not be started up.

Let the minimum value of T obtained by the SFTAP be $T_{\min}$, then $T_{\min} = EC^{\sim}$. The SFTAP can provide a family of reasonable values of T by the values of $EC^{\sim}$ and β in the second stage. The subsequent values of T will get an increment $\Delta\beta$ based on the former one. Since both the value of T and the current solution s will affect $\Lambda_T(s)$, there exist two layers of iterations in A2. The inner-layer iterations are driven by

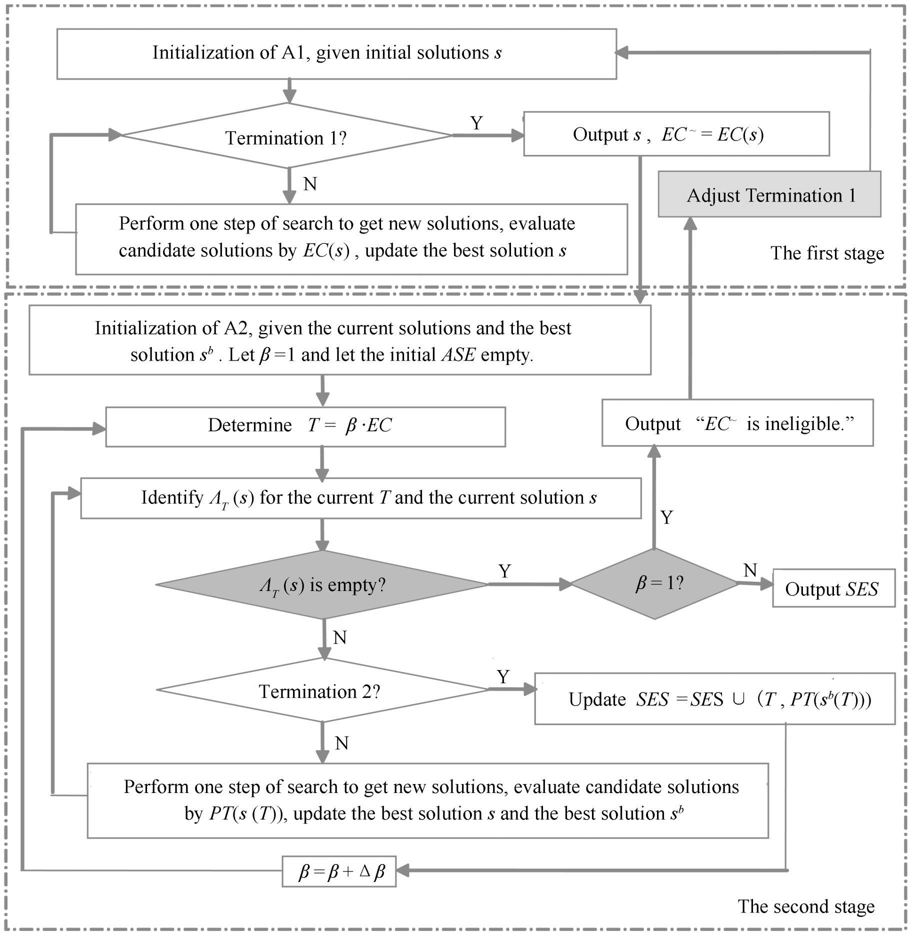

Fig. 1. The approximate solution framework of SFTAP.

updating the current solution, and the outer-layer iterations are driven by updating the value of T. Since the value of $EC^{\sim}$ is determined in the first stage, updating the value of T is exactly updating the value of β. The inner-layer iterations run at $\beta = 1$ until the termination condition of A2 (referred to as Termination 2) is satisfied and an efficient solution is included into the SES. Accordingly, the outer-layer iterations run at an updated value of β by $\beta = 1 + \Delta\beta$ until one more efficient solution is included into the SES or the SFTAP is terminated. For each value of β, A2 runs for the corresponding PSP|T for only one time. As the value of β is updated, A2 is performed for a family of PSP|T . The SFTAP is terminated only if the result of checking segment is null TBS while satisfying $\beta > 1$. This situation demonstrates that the provided value of T at this time is beyond the reasonable-value interval $[EC^*, WC^*]$. In summary, the checking segment has two different functions in the SFTAP: a null TBS at $\beta = 1$ will start the feedback adjusting mechanism, and a null TBS at $\beta > 1$ will terminate the SFTAP.

In the second stage of SFTAP, since Termination 2 is the termination condition of the inner-layer iterations, Termination 2 affects the quality of SES. If A2 is an approximate algorithm, probably not all possible feasible solutions are checked by the checking segment. Let $T_{\max}$ be the last value of T in (11). Maybe $T_{\max} > WC^*$ occurs and pseudo efficient solutions may be included into the SES. We refer to a value of T provided by the SFTAP as a pseudo reasonable value if it is in fact beyond the reasonable-value interval $[EC^*, WC^*]$ and refer to an efficient solution as a pseudo efficient solution if it is obtained under a pseudo reasonable value. Two pseudo reasonable values are illustrated in Fig. 2. Pseudo reasonable values are included in the SES because we apply the checking segment in the SFTAP instead of solving the WCP and adopt an approximate algorithm for A2. The SFTAP generates pseudo reasonable values just due to its approximate nature. We hope that the number of pseudo efficient solutions would decrease till to be zero as the approximation degree of A2 is improved. This will be proved in the following analysis in next section.

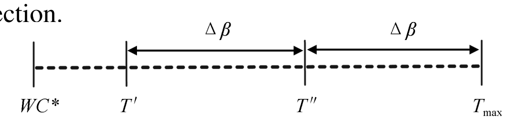

Fig. 2. Two pseudo reasonable values T' and T''.

D. The Analysis of the Approximation of SFTAP

In fact, for the SFTAP, the quality of SES and the cardinality of SES indicate the solution quality and the solution diversity of efficient solutions respectively. We discuss the approximation of the SFTAP by assessing the quality of SES and the cardinality of SES.

Let $s^b(T)$ denote the PT solution obtained by the SFTAP. $s^b(T)$ is an approximation of the optimal PT solution

$s^*(T)$, i.e., $s^*(T) = \arg\min_{s(T)\in S} PT(s(T))$ and $s^b(T) \approx \arg\min_{s(T)\in S} PT(s(T))$. The values T obtained by the SFTAP are within the interval $[T_{\min}, T_{\max}]$, then the SES can be represented by

$$SES = \{(T, PT(s^b(T))) : T = T_{\min}, \ldots, T_{\max}\} \quad (13)$$

We evaluate the quality of SES by two elements of efficient solution $(T, PT(s(T)))$ including the quality of the interval $[T_{\min}, T_{\max}]$ and the quality of $PT(s^b(T))$. Especially, we define the right gap g_r and left gap g_l between the obtained reasonable-value interval $[T_{\min}, T_{\max}]$ and the optimal reasonable-value interval $[EC^*, WC^*]$ respectively by $g_r = |T_{\max} - WC^*|$ and $g_l = |EC^* - T_{\min}|$. Of course, smaller values of both g_r and g_l indicate better quality of $[T_{\min}, T_{\max}]$. Pseudo efficient solutions in the SES are the result of the SFTAP to surrogate the PSPF. The number of pseudo efficient solutions (exactly the number of pseudo reasonable values) in the SES should be an index of approximation degree of SFTAP. In order to discuss the approximation degree of SFTAP, we give the following definition.

Definition 3: $\forall\varepsilon \geq 0$, if $|g_r| \leq \varepsilon$, $|g_l| \leq \varepsilon$, and $|PT(s^b(T)) - PT(s^*(T))| \leq \varepsilon$ are satisfied, we refer to the SFTAP as an $\varepsilon - optimal$ solution framework for the PSPF.

Thus we have the following Theorem 1.

Theorem 1: If A1 and A2 can generate ε-$optimal$ solutions with $\varepsilon = 0$ for the ECP$^\sim$ and the PSP$^\sim$ respectively, the SFTAP is an ε-$optimal$ solution framework with $\varepsilon = 0$ for the PSPF, i.e., if both A1 and A2 are exact algorithms respectively for the ECP$^\sim$ and the PSP$^\sim$, the SFTAP is an exact solution framework for the PSPF, and no pseudo effective solution exists in the SES.

Proof: According to the expression (10), $T_{\min} = EC^\sim$. If A1 can generate the ε-$optimal$ solution with $\varepsilon = 0$ for the ECP$^\sim$, it means that the obtained solution of ECP$^\sim$ can be exactly the optimal solution of ECP, then $T_{\min} = EC^\sim = EC^*$.

If A2 can generate the ε-$optimal$ solutions with $\varepsilon = 0$ for the PSP$^\sim$, it means that A2 can generate the ε-$optimal$ solution with $\varepsilon = 0$ to each PSP|T on the interval $[T_{\min}, T_{\max}]$. It demonstrates that each PSP|T is effective. For any value of T on the interval $[T_{\min}, T_{\max}]$, the optimal solution $s^*(T)$, where $s^*(T) = \arg\min_{s\in S} PT(s)$, can be obtained by performing A2 on the PSP|T. $s^*(T)$ must have been checked in the SFTAP and the result is nonempty $\Lambda_T(s^*(T))$ because if not the SFTAP must have been terminated at T and the value of T has been discarded. According to (1) and (2), there must be $PT(s^*(T)) \geq 0$. Thus any feasible solution s other than $s^*(T)$ satisfies $0 \leq PT(s^*(T)) \leq PT(s(T))$. If $PT(s^*(T)) = 0$, there must be $T = WC^*$ due to nonempty $\Lambda_T(s^*(T))$. If $PT(s^*(T)) > 0$, then $PT(s(T)) > 0$, $\Lambda_T(s)$ is bound to be nonempty, and $T < WC^*$. Anyway, for any value of T on the interval $[T_{\min}, T_{\max}]$, we have $T \leq WC^*$. The optimal solution $(T, PT(s^*(T)))$ must be an effective solution and is impossibly a pseudo effective solution. Specially for $T = T_{\max}$, we have $T_{\max} \leq WC^*$. Conversely, let $T = T_{\max} + \varepsilon$, for any $\varepsilon > 0$, the value of T is beyond the interval $[T_{\min}, T_{\max}]$, the SFTAP must have been terminated by a checking segment in an inner iteration, i.e., there must exist a feasible solution $s^\wedge$ making $\Lambda_T(s^\wedge)$ be empty, this result testifies that $T_{\max} + \varepsilon > WC^*$. Thus we have $WC^* - \varepsilon < T_{\max} \leq WC^*$ for any $\varepsilon > 0$. When $\varepsilon \to 0$, we have $T_{\max} = WC^*$. Then the interval $[T_{\min}, T_{\max}]$ is exactly $[EC^*, WC^*]$.

Due to $s^*(T) = \arg\min_{s\in S} PT(s)$ for any value of T on the interval $[T_{\min}, T_{\max}]$, it demonstrates that $\forall\varepsilon \geq 0$, $|PT(s^b) - PT(s^*(T))| \leq \varepsilon$ is satisfied for any $s^b \in S$ and $s^b \neq s^*(T)$ and for any value of T on the interval $[EC^*, WC^*]$. To summarize, the SFTAP is a ε-$optimal$ solution framework with $\varepsilon = 0$ to the PSPF, i.e., the SFTAP is an exact solution framework to the PSPF, and no pseudo effective solution exists in the SES. ■

Theorem 2: If A1 and A2 generate ε-$optimal$ solutions with $\varepsilon > 0$ to the ECP$^\sim$ and the PSP$^\sim$ respectively, the SFTAP is an ε-$optimal$ approximate framework with $\varepsilon > 0$ to the PSPF. The number of pseudo effective solutions is at most $N_{ps} = [\varepsilon/\Delta\beta]$.

Proof: If A1 generates ε-$optimal$ solutions with $\varepsilon > 0$ to the $ECP^\sim$, then there exists $\varepsilon > 0$ satisfying $|g_l| = |EC^* - T_{\min}| \leq \varepsilon$. If A2 generates ε-$optimal$ solutions with $\varepsilon > 0$ to each PSP|T on the interval $[T_{\min}, T_{\max}]$, referring to the proof of Theorem 1, we have $T_{\max} + \varepsilon > WC^*$ and it testifies $|g_r| = |T_{\max} - WC^*| \leq \varepsilon$. In addition, for any value of T on the interval $[T_{\min}, T_{\max}]$, A2 generates an ε-$optimal$ solution with $\varepsilon > 0$ for the PSP|T , it means that there exists a solution s^b to PSP|T satisfying $|PT(s^b(T)) - PT(s^*(T))| \leq \varepsilon$. Therefore, the SFTAP is a ε-$optimal$ solution framework with $\varepsilon > 0$ for the PSPF.

If $T_{\max} \leq WC^*$, $T_{\max}$ is a reasonable value of T, then no pseudo effective solution will be generated for any value of T on the interval $[T_{\min}, T_{\max}]$. If $T_{\max} > WC^*$, due to $g_r = T_{\max} - WC^* \leq \varepsilon$, the value of T locating between WC^* and $T_{\max}$ must be pseudo effective reasonable values, as seen in Fig. 2. And the number of pseudo effective-solution pairs is the number of pseudo effective reasonable values of T, which is at most $N_{ps} = [T_{\max} - WC^*/\Delta\beta] \leq [\varepsilon/\Delta\beta]$. ■

We can conclude from Theorem 2 that A1 can affect the left gap g_l, and A2 can affect the right gap g_r as well as the solution quality of $PT(s(T))$. Especially in the second stage of SFTAP, $T_{\max}$ is approaching WC^* and the number of pseudo efficient solutions in the SES is decreasing to zero while A2 is approaching a complete enumeration. The constraint $T \leq WC^*$ is regarded as a soft constraint in the SFTAP but not a hard constraint like in the PSPF. The constraint $T \leq WC^*$ is handled through checking segments without solving the WCP while the PSP|T s are solved by the approximate algorithm A2 during the second stage. That is just the reason that the SFTAP will generate pseudo efficient solutions. We might say that generating pseudo efficient solution is exactly the cost for the SFTAP to surrogate the PSPF.

Although the value of T bigger than WC^* can make PSP|T ineffective with respect to Definition 1, pseudo efficient solution may have practical meaning in reality. Pseudo reasonable value T actually means that the corresponding PSP|T will lose the ability of differentiating elite solutions because their objective values $PT(s(T))$ are identically zero.

Seen in the expression (11), the cardinality of SES should be x. In fact, the cardinality of SES is unknown until the SFTAP is terminated. The cardinality of SES depends on the approximation degree of the interval $[T_{\min}, T_{\max}]$ and the value of $\Delta\beta$. A rougher interval $[T_{\min}, T_{\max}]$ and bigger value of $\Delta\beta$ will result in smaller cardinality of SES. Obviously, the cardinality of SES obtained by the SFTAP is $x = [T_{\max} - T_{\min}/\Delta\beta]$. We of course hope bigger SES in order to get better solution diversity. However, in fact, the SES may include not only a number of real efficient solutions but also possible pseudo efficient solutions. According to the conclusion of Theorem 2, pseudo efficient solutions will be eliminated only if $\varepsilon < \Delta\beta$. Thus, we may set bigger values of $\Delta\beta$ to avoid getting more pseudo efficient solutions into the SES for certain ε. Unfortunately, bigger values of $\Delta\beta$ will make the cardinality of SES smaller. Thus, the best way to improve the diversity of efficient solutions of SFTAP is to improve the approximation of two-stage problems of SFTAP.

As a whole, the SFTAP is actually a procedure to integrate both identification and optimization of robust scheduling model. The value of T is a variable to be determined in the identification of robust scheduling model, and the robust solution $s(T)$ is a variable to be solved in the optimization of robust scheduling model. The first stage of SFTAP undertakes the mission of providing an initial standard performance T for robust scheduling model.

There is a need to note that the proposed robust optimization framework is actually not limited to be applied in scheduling problems. In fact, the procedure proposed here can also be applied to other discrete scenario optimization problems. Specific solution algorithms for A1 and A2 should be developed for specific discrete optimization systems. Efficient solution algorithms will be developed in our future work. Due to the limitation of the length of this paper, we just applied the approximate solution framework of SFTAP to a scenario job-shop scheduling problem by using two algorithms developed before to testify the aforementioned analysis conclusions.

IV. Computational Experiment and Result Analysis

Job-shop scheduling problem (JSP) is a typical manufacturing scheduling problem, and it can be basically taken as a research example for other complex discrete optimization problems. It has been well-known as one of the hardest discrete optimization problems and numerous exact and heuristic algorithms have been proposed for deterministic JSPs [25], [26]. Much research efforts have been made on robust JSPs, which can be referred to literatures [27]−[32].

The tested scenario job-shop scheduling problem (SJSP) in this paper is stated as follows: n jobs are to be processed on m machines. Each job is processed on each machine exactly one time, which is called an operation. Each machine can process only one job at a time and preemption is not allowed. A sequence in which a job visits each machine (i.e., precedence relations) is known a priori. Each operation has a processing time. Assume that processing times of all operations are uncertain. Let the scenario set Λ describe all possible processing times. Each scenario λ of Λ is designated by a vector $P^{\lambda} = (p_i^{\lambda}, i = 1, \ldots, N)$, where p_i^{λ} represents a possible realization of processing time of the operation i on the scenario λ, and $N = n \times m$ is the number of all operations. A feasible solution s represents the sequences of jobs on all machines, which are subject to precedence relations. The system performance $C(s, \lambda)$ to be minimized is the makespan, which is the completing time of the last operation to be processed. Even if all processing times of operations are deterministic, the JSP with the criterion of makespan is already strongly NP-hard [33], thus the ECP, the WCP, and the PSP|T s for the SJSP are all strongly NP-hard.

In this section, we plan to apply the SFTAP to the SJSP. The genetic simulated-annealing (GSA) algorithm [34], whose main framework come from adapting an efficient hybrid algorithm ever designed for deterministic job-shop scheduling problems [35], was adopted to solve the ECP$^{\sim}$ as alternative of A1. We set the maximal generation of evolution (denoted by $MaxGen$) as the termination condition of Termination 1, which differs from that in [35]. The reason is that an identical termination condition for different instances is required in the first stage of SMTF in order to ensure that different degrees of approximation of $EC^{\sim}$ can be obtained when Termination 1 is adjusted. The scenario-merging tabu search (SNTS) algorithm [36] specialized for the PSP|T was used as alternative of A2. A2 involves two termination conditions respectively for two layers of iterations. Termination 2 is the termination condition of the inner-layer iterations of SNTS. Once Termination 2 is satisfied, the inner-layer iterations on the current value of T will stop and an efficient solution is output into the SES. We define Termination 2 as the maximum number of iterations (denoted by $Maxiter$) of SNTS without improving the best solution obtained so far.

An extensive experiment was conducted to investigate the SFTAP for the SJSP. We tested the effectiveness of the solution framework of SFTAP by examining the efficient-solution quality and the cardinality of SES. Due to the complexity of SJSP, the tested instances are derived only from the deterministic classical JSP benchmark FT10 designed by Fisher and Thompson [37]. We made processing times of all operations of the benchmark JSP instance uncertain and use scenario sets to describe uncertain processing times. Each scenario set includes 20 possible scenarios for all operations, i.e., $|\Lambda| = 20$. Each possible processing time is generated randomly from the uniform distribution on the interval $[p_{\min}, p_{\max}] = [10\ 100]$. Ten instances were generated in each test. All tests were performed in C language under the Microsoft Visual C++ 6.0 programming environment. The experiment was conducted on the computer with Pentium G630 2.7 GHz CPU and 2.0 GB RAM.

Firstly, we plan to observe the effects of Termination 1 and 2 as well as the parameter $\Delta\beta$ on the solution quality and the solution diversity of SES. We tested an instance to observe the variation of SES as Termination 1 or Termination 2 varies respectively. Given $\Delta\beta = 0.02$, Termination 1 was given under two different conditions: $MaxGen = 30$ and $MaxGen = 60$ respectively, and Termination 2 was given under five different conditions from $Maxiter = 200$ to $Maxiter = 5000$. The

TABLE I
COMPARISONS OF SES OBTAINED UNDER DIFFERENT CONDITIONS OF TERMINATION 1 AND TERMINATION 2 ($\Delta\beta = 0.02$)

Termination 2: ($Maxiter$)	Termination 1: $MaxGen$=30 $SES : (T, PT(s(T)))$			Termination 1: $MaxGen$=60 $SES : (T, PT(s(T)))$				
200	(1268, 475.8)	(1293, 171.1)	(1318, 32.6)	(1199, 2258)	(1223, 1089)	(1247, 395.3)	(1271, 91.3)	(1295, 7.3)
500	(1268, 475.8)	(1293, 28.3)	$(1318, 12.6)^p$	(1199, 2258)	(1223, 1089)	(1247, 395.3)	(1271, 91.3)	(1295, 7.3)
1000	(1268, 416.5)	(1293, 28.3)		(1199, 2258)	(1223, 1089)	(1247, 283.7)	(1271, 52.0)	(1295, 7.3)
2000	(1268, 87.4)	$(1293, 28.3)^p$		(1199, 2258)	(1223, 1089)	(1247, 283.7)	(1271, 52.0)	$(1295, 7.3)^p$
5000	(1268, 87.4)			(1199, 1265)	(1223, 1089)	(1247, 283.7)	(1271, 52.0)	

The superscript^p: pseudo efficient solution

obtained efficient solutions of SES under different conditions of Termination 1 and 2 are presented in details in Table I.

Table I shows that more efficient solutions were obtained under a higher condition of Termination 1 because a better value of $EC^{\sim}$ was obtained under $MaxGen = 60$ and it enlarged the interval $[T_{\min}, T_{\max}]$. Comparing the results under different conditions of Termination 2, we can also notice that the solution quality was improved for any provided values of T under $MaxGen = 30$ when Termination 2 got a higher condition. Consistent results were observed under $MaxGen = 60$. Moreover, it is shown that the cardinality of SES was consistently getting smaller as Termination 2 got higher conditions under both $MaxGen = 30$ and $MaxGen = 60$. The reason may be that pseudo efficient solutions obtained under lower conditions of Termination 2 were identified and removed from the SES when more feasible solutions were checked under higher conditions of Termination 2. Until $Maxiter = 5000$, we got the best result of SES, which includes the best solution quality of PT objective and the least number of pseudo efficient solutions. It demonstrates that the number of pseudo efficient solutions decreased as the quality of efficient solutions was improved by A1 and A2. In summary, the results of Table I testified the conclusions of Theorem 2.

Further, the variation of cardinality of SES under different conditions of Termination 1 and Termination 2 was investigated in all ten instances and the computational results are presented in Table II. It is shown that the cardinality of SES was consistently getting smaller in ten instances as Termination 2 was given higher conditions of Termination 1, and the bigger cardinality of SES under $MaxGen = 60$ than $MaxGen = 30$ was obtained for all conditions of Termination 2 in ten instances. Until $Maxiter$= 5000, the smallest SES was gotten respectively under two conditions of Termination 1, where at least one efficient solution was obtained under $MaxGen = 30$ and more efficient solutions were obtained under $MaxGen = 60$ in all ten instances. The results of Table II testified that the SFTAP is effective for decision maker to generate at least one pair of efficient solution. Promoting Termination 1 can make the SFTAP enlarge the interval $[T_{\min}, T_{\max}]$ and generate more number of efficient solutions for a specific value of $\Delta\beta$. Promoting Termination 2 can make the SFTAP improve the quality of PT robust solution and reduce the cardinality of SES due to eliminating pseudo efficient solutions.

Finally, we tested all ten instances to observe the influences of different values of $\Delta\beta$ on the cardinality of SES. Given $MaxGen = 60$ and $Maxiter = 5000$, $\Delta\beta$ is given by three

TABLE II
COMPARISONS OF THE CARDINALITY OF SES UNDER DIFFERENT CONDITIONS OF TERMINATION 1 AND TERMINATION 2 ($\Delta\beta = 0.02$)

Instances	$MaxGen$	$Maxiter$ 200	500	1000	2000	5000
1	30	2	2	2	2	2
	60	5	5	5	4	4
2	30	3	3	3	3	2
	60	5	5	5	5	4
3	30	3	3	2	2	1
	60	5	5	5	5	4
4	30	2	2	2	2	1
	60	5	4	4	4	4
5	30	3	3	2	2	2
	60	5	4	4	4	4
6	30	2	2	1	1	1
	60	4	3	3	3	3
7	30	2	2	2	2	1
	60	4	4	4	4	4
8	30	3	2	2	2	2
	60	5	5	5	5	5
9	30	2	2	2	2	2
	60	5	5	5	5	5
10	30	3	2	2	1	1
	60	5	5	5	4	4
Average	30	2.5	2.3	2.0	1.9	1.5
	60	4.8	4.5	4.5	4.3	4.1

TABLE III
COMPARISONS OF THE CARDINALITY OF SES OBTAINED UNDER DIFFERENT VALUES OF $\Delta\beta$

Instances	$\Delta\beta$ 0.010	0.015	0.020
1	8	6	4
2	8	7	4
3	7	5	4
4	6	5	4
5	8	5	4
6	7	4	3
7	6	5	4
8	9	6	5
9	6	5	5
10	8	4	4

different values: 0.01, 0.015, and 0.02. The computational results are presented in Table III. The results of Table III testified that the value of $\Delta\beta$ affects the solution diversity of SES and a smaller value of $\Delta\beta$ will generate better diversity of SES. Among ten instances, the instance got the biggest cardinality of SES at $\Delta\beta = 0.01$, where the frontier of nine efficient solutions is presented in Fig. 3.

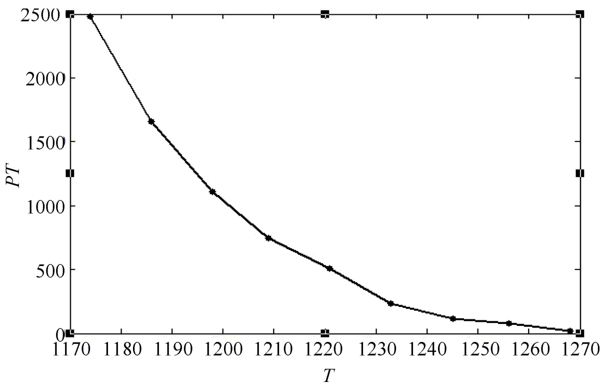

Fig. 3. The frontier of efficient solutions $\Delta\beta = 0.01$.

V. Conclusions

A robust scheduling formulation PSPF for uncertain discrete optimization system under various scenarios is proposed in this paper. This formulation aims at hedging against the risk of system performance degradation while keeping excellent expected system performance. In fact, the PSPF is a bi-objective problem, which represents a robust scheduling framework. In order to solve the bi-objective problem, a surrogate approximate approach with two stages called SFTAP is proposed to obtain the SES of PSPF instead of an exact approach. Two problems are approximately handled instead of handling three problems with the same intractability. Approximate search algorithm structures are developed for two-stage problems of SFTAP. Due to the approximate nature, pseudo efficient solutions may appear in the SES. The approximation degree of SFTAP is analyzed. The number of pseudo efficient solutions in the SES can indicate the approximation degree of SFTAP. We conclude that the approximation degree of SFTAP is dependent on the approximation degree of two-stage approximate problems. The framework SFTAP was applied to a scenario job-shop scheduling problem. The computational results testified the conclusions of analysis. It demonstrates that the framework SFTAP is effective. It can provide multiple selections for the decision maker and is able to obtain better robust solutions for the bi-objective problem as long as two approximate problems obtain better solutions. The robust scheduling framework studied in this paper can provide a specific paradigm for formulating and solving robust discrete optimization problems.

References

[1] W. Herroelen and R. Leus, "Project scheduling under uncertainty: Survey and research potentials," *Eur. J. Oper. Res.*, vol. 165, no. 2, pp. 289−306, Sep. 2005.

[2] I. Sabuncuoglu and S. Goren, "Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research," *Int. J. Comput. Integr. Manuf.*, vol. 22, no. 2, pp. 138−157, Feb. 2009.

[3] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios, "Robust optimization of large-scale systems," *Oper. Res.*, vol. 43, no. 2, pp. 264−281, Apr. 1995.

[4] P. Kouvelis and G. Yu, *Robust Discrete Optimization and Its Applications*. Norwell, MA: Kluwer Academic Publisher, 1997.

[5] Z. K. Li and M. G. Ierapetritou, "Robust optimization for process scheduling under uncertainty", *Ind. Eng. Chem. Res.*, vol. 47, no. 12, pp. 4148−4157, May 2008.

[6] A. D. Che, J. Feng, H. Chen, and C. Chu, "Robust optimization for the cyclic hoist scheduling problem," *Eur. J. Oper. Res.*, vol. 240, no. 3, pp. 627−636, 2015.

[7] Z. Q. Chang, S. J. Song, Y. L. Zhang, "Distributionally robust single machine scheduling with risk aversion," *Eur. J. Oper. Res.*, to be published.

[8] M. Pinedo, "Minimizing the expected makespan in stochastic flow shops," *Oper. Res.*, vol. 30, no. 1, pp. 148−162, Feb. 1982.

[9] S. D. Wu, E. S. Byeon, and R. H. Storer, "A Graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness," *Oper. Res.*, vol. 47, no. 1, pp. 113−124, Jan. 1999.

[10] X. X. Lin, S. L. Janak, and C. A. Floudas, "A new robust optimization approach for scheduling under uncertainty: I. bounded uncertainty," *Comp. Chem. Eng.*, vol. 28, no. 6−7, pp. 1069−1085, Jun. 2004.

[11] A. L. Soyster, "Convex programming with set-inclusive constraints and applications to inexact linear programming," *Oper. Res.*, vol. 21, no. 5, pp. 1154−1157, Oct. 1973.

[12] A. Ben-Tal and A. Nemirovski, "Robust solutions of uncertain linear programs," *Oper. Res. Lett.*, vol. 25, no. 1, pp. 1−13, Aug. 1999.

[13] D. Bertsimas and M. Sim, "Robust discrete optimization and network flows," *Math. Program.*, vol. 98, no. 1−3, pp. 49−71, Sep. 2003.

[14] R. L. Daniels and P. Kouvelis, "Robust scheduling to hedge against processing time uncertainty in single-stage production," *Manag. Sci.*, vol. 41, no. 2, pp. 363−376, Feb. 1995.

[15] P. Kouvelis, R. L. Daniels, and G. Vairaktarakis, "Robust scheduling of a two-machine flow shop with uncertain processing times," *IIE Trans.*, vol. 32, no. 5, pp. 421−432, May 2000.

[16] C. Artigues, J. C. Billaut, and C. Esswein, "Maximization of solution flexibility for robust shop scheduling," *Eur. J. Oper. Res.*, vol. 165, no. 2, pp. 314−328, Sep. 2005.

[17] R. L. Daniels and J. E. Carrillo, "beta-robust scheduling for single-machine systems with uncertain processing times," *IIE Trans.*, vol. 29, no. 11, pp. 977−985, Nov. 1997.

[18] B. Wang, X. F. Yang, and Q. Y. Li, "Bad-scenario set based risk-resisting robust scheduling model," *Acta Automat. Sin.*, vol. 38, no. 2, pp. 270−278, Feb. 2012.

[19] C. W. Wu, K. N. Brown, and J. C. Beck, "Scheduling with uncertain durations: modeling beta-robust scheduling with constraints," *Comput. Oper. Res.*, vol. 36, no. 8, pp. 2348−2356, Aug. 2009.

[20] P. Kouvelis, A. A. Kurawarwala, and G. J. Gutierrez, "Algorithms for robust single and multiple period layout planning for manufacturing systems," *Eur. J. Oper. Res.*, vol. 63, no. 2, pp. 287−303, Feb. 1992.

[21] T. Assavapokee, M. J. Realff, J. C. Ammons, and I. H. Hong, "Scenario relaxation algorithm for finite scenario-based minCmax regret and minCmax relative regret robust optimization," *Comput. Oper. Res.*, vol. 35, no. 6, pp. 2093−2102, Jun. 2008.

[22] M. A. Al-Fawzan and M. Haouari, "A bi-objective model for robust resource-constrained project scheduling," *Int. J. Prod. Econ.*, vol. 96, no. 2, pp. 175−187, May 2005.

[23] R. Qing-dao-er-ji, Y. P. Wang, and X. L. Wang, "Inventory based two-objective job shop scheduling model and its hybrid genetic algorithm," *Appl. Soft Comput.*, vol. 13, no. 3, pp. 1400−1406, Mar. 2013.

[24] D. M. Lei, "Simplified multi-objective genetic algorithms for stochastic job shop scheduling," *Appl. Soft Comput.*, vol. 11, no. 8, pp. 4991−4996, Dec. 2011.

[25] A. S. Jain and S. Meeran, "Deterministic job-shop scheduling: past, present and future," *Eur. J. Oper. Res.*, vol. 113, no. 2, pp. 390−434, Mar. 1999.

[26] C. Y. Zhang, P. G. Li, Z. L. Guan, and Y. Q. Rao, "A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem," *Comput. Oper. Res.*, vol. 34, no. 11, pp. 3229−3242, Nov. 2007.

[27] 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, Sep. 1994.

[28] E. S. Byeon, S. D. Wu, and R. H. Storer, "Decomposition heuristics for robust job-shop scheduling," *IEEE Trans. Rob. Automat.*, vol. 14, no. 2, pp. 303−313, Apr. 1998.

[29] E. Kutanoglu and I. Sabuncuoglu, "Experimental investigation of iterative simulation-based scheduling in a dynamic and stochastic job shop," *J. Manuf. Syst.*, vol. 20, no. 4, pp. 264−279, Jan. 2001.

[30] 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.

[31] S. V. Mehta and R. M. Uzsoy, "Predictable scheduling of a job shop subject to breakdowns," *IEEE Trans. Rob. Automat.*, vol. 14, no. 3, pp. 365−378, 1998.

[32] M. Sakawa and R. Kubota, "Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms," *Eur. J. Oper. Res.*, vol. 120, no. 2, pp. 393−407, Jan. 2000.

[33] J. K. Lenstra and A. H. G. R. Kan, "Computational complexity of discrete optimization problems," *Ann. Discrete Math.*, vol. 4, pp. 121−140, Dec. 1979.

[34] B. Wang, X. F. Yang, and Q. Y. Li, "Genetic simulated-annealing algorithm for robust job shop scheduling," in *Fuzzy Information and Engineering, Vol. 2: Advances in Intelligent and Soft Computing*, B. Y. Cao, T. F. Li, and C. Y. Zhang, Eds. Berlin Heidelberg: Springer, 2009, pp. 817−827.

[35] L. Wang and D. Z. Zheng, "An effective hybrid optimization strategy for job-shop scheduling problems," *Comput. Oper. Res.*, vol. 28, no. 6, pp. 585−596, May 2001.

[36] B. Wang, J. Liu, and Y. Y. Yu, "Scenarios based Tabu search algorithm for robust job-shop scheduling," in *The Proc. Chinese Control Conf.*, Xi'an, China, 2013, pp. 2491−2496.

[37] J. F. Muth and C. L. Thompson, *Industrial Scheduling*, Englewood Cliffs, NJ: Prentice-Hall, 1963.

Bing Wang received her B. E. degree in precision mechanism and precision instrument from University of Science and Technology of China in 1989, her Master degree in operations research and control theory from Shangdong University, China, in 2001, and her Ph. D. degree in control theory and control engineering from Shanghai Jiao Tong University, China, in 2005. From March 2008 to March 2009, she was a visiting scholar with the Department of Industry and System Engineering, University of Florida, USA. Before January 2011, she was a faculty member of Shandong University at Weihai, and currently she is a professor at the School of Mechatronic Engineering and Automation, Shanghai University, China. Bing Wang's current research interests include scheduling theory and algorithm, production planning and scheduling, robust discrete optimization theory and application, and health-care management and optimization. She is the author or coauthor of over 50 published papers. Corresponding author of this paper.

Xuedong Xia received his B. E. degree from the School of Physics and Electronic Electrical Engineering, Huaiyin Normal University, China, in 2013. He is currently a master student at the School of Mechatronic Engineering and Automation, Shanghai University. His research interests include production planning and scheduling.

Hexia Meng received her B. E. degree from the School of Electric Power, North China University of Water Resources and Electric Power, China, in 2012. She is currently a master student at the School of Mechatronic Engineering and Automation, Shanghai University. Her research interests include production planning and scheduling.

Tao Li received the B. E. degree in automation from Nankai University, Tianjin, China, in 2004, and the Ph. D. degree in systems theory from the Academy of Mathematics and Systems Science (AMSS), Chinese Academy of Sciences (CAS), Beijing, China, in 2009. From July 2009 to December 2013, he was a faculty member of AMSS, CAS. Since December 2013, he has been with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China, where now he is a professor.

His current research interests include stochastic systems, networked control, multi-agent systems and sensor networks. He was mentioned as one of the five finalists for the Young Author Prize of the 17th IFAC World Congress, Seoul, 2008. He received the Special President Prize of Chinese Academy of Sciences in 2009, the Best Paper Award of the 7th Asian Control Conference with coauthors in 2009, the 2009 Singapore Millennium Foundation Research Fellowship and the 2010 Endeavor Research Fellowship from Australian Government. He was entitled Dongfang Distinguished Professor by Shanghai Municipality in 2012 and received National Science Found for Outstanding Young Scholars from National Natural Science Foundation of China in 2015. He is a member of IFAC Technical Committee on Networked Systems and a council member of Shanghai Association of Automation. He is a Senior Member of IEEE.