European Journal of Operational Research 299 (2022) 183–194

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing, Transportation and Logistics

Integrated optimization of test case selection and sequencing for reliability testing of the mainboard of Internet backbone routers

Hanxiao Zhang, Yan-Fu Li*

Department of Industrial and Engineering, Tsinghua University, Beijing 10084, China

ARTICLE INFO

Article history:
Received 25 April 2020
Accepted 14 June 2021
Available online 24 June 2021

Keywords:
Reliability
Reliability testing plan
Test case selection
Test case sequencing
Branch-and-price

ABSTRACT

Internet backbone refers to the principal data routes between large, strategically interconnected networks and core routers on the Internet. Internet backbone router is essentially the core router of Internet backbone and its performance is mainly relevant to the reliability of its mainboard. The mainboard is an embedded system consisting of hardware and software. Its reliability testing involves executing a number of test cases, which are designed to expose potential defects, under harsh environmental conditions. The testing process is largely prolonged due to the dramatic increase of the number of test cases, mainly due to the continuous increase and upgrade of its functional modules. Thus, there is a big demand from industry to improve the reliability testing efficiency and effectiveness. In this work, we exploit the principles of regression testing in software maintenance: test case selection and prioritization, and construct two testing planning models to largely reduce the testing time as well as to improve the effectiveness of failure detections. The former is a two-step model we introduced in previous work that optimizes test case selection and test case sequencing sequentially. The latter, an integrated model is newly developed, optimizing the test case selection and sequencing simultaneously with the precedence constraints among the test cases. Moreover, we propose exact algorithms based on branch-and-price for solving these two models. Finally, we present a case study demonstrating that the integrated model outperforms the two-step method and the advantage is more significant if the sequencing objective has greater weight in the integrated objective function.

1. Introduction

Internet is a critical infrastructure for most countries on earth. Yet, the human society still faces a brisk demand for better Internet services: faster speed and more stable connection. For this reason, the reliability of the Internet backbone router, the critical device of Internet, is a vital issue. An Internet backbone refers to one of the principal data routes between large, strategically interconnected networks and core routers on the Internet. Its core routers, i.e. Internet backbone routers, are routers powerful enough to handle information on the Internet backbone and are capable of directing data to other routers in order to send it to the final destination. Thus, the processing speed and the reliability of Internet backbone router directly affect the performance of Internet services. The reliability of Internet backbone router is mainly relevant to the reliability of its main component, the mainboard, which becomes ever complex with the increasing number of functional modules. Thus, the reliability testing of the mainboard is necessary to ensure the quality of the products.

One important character of this device is that it hybrids hardware and software, as an embedded system, such that it should undergo both the reliability testing for hardware and that for software. In other words, mainboard should be tested by executing many test cases under certain accelerated stresses, e.g. temperature and voltage. A test case is a small program to diagnose if certain functions/features, e.g. connectivity and forwarding, of the mainboard operate properly. To design an effective testing plan, in reliability engineering domain most research works have focused on the accelerated conditions and studied the choice of the stress, specimen and censoring time and optimize these testing parameters (Chen, Xu & Ye, 2016; Hu, Plante & Tang, 2015; Liu & Tang, 2013; Zhu & Elsayed, 2013). The literature about the reliability testing plan is detailed in Section 2. Differently, in this work we focus on the design of the test case selection and sequencing to optimize the testing plan, as this is the particular request from our industrial partner, a multinational telecommunications equipment provider.

Due to the high-reliability requirement of the backbone router mainboard, the number of test cases increases dramatically in recent years. The testing process for each mainboard consists of tens

* Corresponding author.
E-mail addresses: zhx17@mails.tsinghua.edu.cn (H. Zhang), liyanfu@tsinghua.edu.cn (Y.-F. Li).

https://doi.org/10.1016/j.ejor.2021.06.028
0377-2217/

of thousands of test cases and lasts one days. Under the current testing plan of our industrial partner, all available test cases must be executed one time under each environmental testing condition in order to detect the product failures, which largely prolongs the testing process duration. We seek the methods to improve the testing process mainly due to the observation that there is a large number of test cases that have not exposed any failures in historical records. However, the studies on the reliability testing plan neither address the reduction of total testing time by the selection of test cases nor investigate the effectiveness of the testing plan. Thus, we propose to perform the selection and sequencing of the test cases to meet the time reduction and improve the effectiveness of reliability testing. These two operations were borrowed from the regression testing in the software development process, which is reviewed in Section 2. The details about the operations and the constraints are modified to adapt to our problem.

The testing plan design of the embedded system in our paper is operated on the selection and sequencing of test cases in software testing. As for the accelerated conditions, it is used to accelerate the degradation of related hardware and only considered as the coverage constraint in reliability testing. Note that in our conference publication (Zhang, Liu & Li, 2020), we have abstracted this problem into an optimization model and constructed a two-step method to realize the selection and sequencing of the test cases. The principle of this two-step model is to select the test cases under a testing time constraint and then reorder the selected test cases to reach the optimal criteria, which is similar to Silva et al. (2016). In this work, we develop a new design model, an integrated model, to optimize test case selection and sequencing at the same time. Krishnamoorthi and Mary (2009) proposed to deal with the prioritization with selection simultaneously in the software regression testing. They presumed that the test cases are independent and have no execution ordering dependencies. However, we consider a more practical scenario which allows the ordering dependencies (i.e. precedence constraints) to further complicate the situation.

The sequencing problem is NP-hard, such that our previous work cannot efficiently solve the model. Thus, we reformulate this test case sequencing model as a single machine scheduling problem and use the time-index formulation to reformulate this model. Then, we propose an algorithm to solve the earliness-tardiness scheduling problem considering the precedence constraints and idle time. There are few papers dealing with precedence constrained single machine scheduling problem (Correa & Schulz, 2005; Davari, Demeulemeester, Leus & Nobibon, 2016; Rostami, Creemers & Leus, 2019; Schulz & Uhan, 2011; Tanaka & Sato, 2013; Tang, Xuan & Liu, 2007). To the best of our knowledge, scheduling problem with selection, precedence constraints, idle time and total weighted earliness-tardiness penalties have not yet been studied in the literature. To address the limitations of the existing studies, the main contributions of this work are:

(1) We propose an efficient exact algorithm for the previous two-step model in order to solve a larger size of the test cases.
(2) We propose a new design model to optimize the testing plan, an integrated model that optimizes test case selection and test case sequencing simultaneously.
(3) We propose the exact algorithm rather than the heuristic algorithm to solve this integrated model via combining the column generation (CG) algorithm and branch-and-bound (B&B) methodology (i.e. branch-and-price).

The rest of this paper is organized as follows. Section 2 outlines the literature on the reliability testing, software regression testing and single machine scheduling problem. Section 3 formally describes this problem. Section 4 briefly describes the two-step model for test case selection and test case sequencing and propose the efficiently exact algorithm. Section 5 constructs the new proposed integrated model and presents the exact algorithm based on the branch-and-price algorithm. Numerical experiments and a case study are presented in Section 6. We discuss the effects of the parameters on the structural properties of the models in Section 6.1. In Section 6.2, a case study is presented which compares the solution of the integrated model with the two-step model. In Section 7, we conclude this work and discuss future work for our problem.

2. Literature review

2.1. Reliability testing plan

Reliability testing is a testing process to predict the lifetime of the products or detect potential failures in the presence of the testing conditions (Escobar & Meeker, 2006). The most common and widely used technique of reliability testing is to shorten the product testing time by accelerating their degradation with environmental stresses. The effects of environmental stresses in the testing process have been systematically studied in the literature (Nelson, 2009). Thus, the weakest products can be detected quickly and the causes of the failures can be found for correction before production. Reliability testing considered the accelerated conditions has been widely used to the high-reliability hardware in electronic, electromechanical and mechanical systems. The commonly applied stress testing processes include thermal cycles, temperature extremes, etc. For examples, Acevedo, Jackson and Kotlowitz (2006) used the thermal cycling accelerated testing for the radio unit to predict the field reliability. Choi, Joergensen and Blaabjerg (2016) applied the temperature stress to the accelerated power cycling test of the insulated-gate bipolar transistor modules.

For the successful implementation of a testing process, it is important to design an efficacious testing plan that selects and optimizes the important testing parameters, such as the stress profiles, sample size, sample allocation, test duration. In recent years, a number of publications have studied the optimal design of a reliability testing plan (Nelson, 2005, 2015) considering the type of censoring schemes (Chen et al., 2016), stress loading (Hu et al., 2015), number of stress levels (Zhu & Elsayed, 2013), size of specimens to each stress level (Liu & Tang, 2013), test constraints (Han, 2015) and so on. However, these studies have not investigated the selection and the sequencing of the test cases to reduce the testing time. It is of particular value for the embedded systems since the number of executed test cases determines the duration of the testing process. The goal of this paper is to fill this gap and to improve the efficiency and effectiveness of reliability testing.

2.2. Software regression testing

Regression testing is a testing process of software to revalidate that the modified features have not adversely affected existing functionalities. It is financially and computationally expensive if we rerun the entire existing test suite (Chittimalli & Harrold, 2009). Therefore, in the literature, several approaches have been proposed to assist the regression testing process including test case selection (TCS), test case prioritization (TCP) and a hybrid approach.

TCS intends to select a subset of test cases to test the changed code of the software instead of re-executing the entire test suite while satisfying the testing requirement denoted by a certain test criterion (Mirarab, Akhlaghi & Tahvildari, 2012). TCP (Hao et al., 2016; Srikanth, Hettiarachchi & Do, 2016) intends to improve the effectiveness of regression testing and enable a faster failure detection by re-ordering the execution schedule of the test cases so that high priority test cases, according to the preferred criterion defined by testers, are executed first. For example, the test cases might be ordered based on the number of failures detected by the test cases.

Test cases which exposed more failures have higher priorities to be scheduled than the other test cases. The failure numbers of all test cases are archived in the history record.

A hybrid regression testing approach deals with both the selection and the prioritization of test cases. Silva et al. (2016) proposed a hybrid method for regression TCS and TCP. First, the test cases that could be executed within the time constraints are selected. Then, the test cases selected in the previous stage are ordered with TCP. Walcott, Soffa, Kapfhammer and Roos (2006) and Krishnamoorthi and Mary (2009) proposed the time-aware TCP problem that prioritizes the test cases within a given testing time constraint. They optimize TCS and TCP at the same time by GA. The assumption is that the test cases are independent. Thus, there is no preferred execution sequence of them such that they can be reordered in any sequence.

2.3. One machine scheduling problem

Test case sequencing process we proposed is similar to the single machine problem with total weighted earliness-tardiness penalties, which is NP-hard (Lawler, 1978). The time-index formulation (Sousa & Wolsey, 1992) is widely used for single and parallel machine scheduling problems since its LP-relaxation bound provided is stronger than other formulations. As for the exact methods, most methods consist of either Lagrangian relaxation (LR) or CG to get the lower bound and use either B&B or dynamic programming (DP) to get the optimality.

- DP + LR.

  Successive Sublimation Dynamic Programming (SSDP) method is efficient to calculate the lower bound of the LR. And efficient algorithms are proposed when machine idle time is forbidden (Tanaka, Fujikuma & Araki, 2009) and machine idle time is allowed (Tanaka & Fujikuma, 2008, 2012). Then they extended the above method considering the precedence constraints under the case that idle time is forbidden (Tanaka & Sato, 2013).

- B&B + LR

  There are two types for the LR formulation. First, the number of occurrences of the jobs in the schedule can be relaxed (Sourd, 2005, 2009; Sourd & Kedad-Sidhoum, 2003). The second approach is to relax the capacity constraints (Sourd & Kedad-Sidhoum, 2008).

- B&B + CG

  Van den Akker, Hurkens and Savelsbergh (2000) derived efficient lower bounds through CG combined with Dantzig-Wolfe reformulation when the idle time is allowed. Instead of adding variables with nonpositive reduced costs, the CG is modified by adding the variables whose reduced cost is less than a small value (Kedad-Sidhoum, Solis & Sourd, 2008). Due to the slow convergence of the CG process, an acceleration strategy based on a temporal decomposition is proposed to improve convergence (Bigras, Gamache & Savard, 2008). During the branching process to get the optimality, a DP procedure is developed to fix variables by Lagrangian bounds to enhance the algorithm (Pessoa, Uchoa, de Aragão & Rodrigues, 2010).

There are few papers that deal with the precedence-constrained single machine scheduling problem. First, the objective to minimize the total weighted completion time has been studied by Correa and Schulz (2005); Schulz and Uhan (2011), etc. Second,

Table 1
Summary of the publications of the single machine schduling problem.

Literature	r_j	T_j	$E_j + T_j$	C_j	IT	PRE
Tanaka et al. (2009)		√	√			
Tanaka and Fujikuma (2008)	√		√		√	
Tanaka and Fujikuma (2012)	√	√	√	√	√	
Tanaka and Sato (2013)		√	√			√
Sourd (2005)			√		√	
Sourd (2009)			√			
Sourd and Kedad-Sidhoum (2003)			√		√	
Sourd and Kedad-Sidhoum (2008)	√		√		√	
Van den Akker et al. (2000)	√	√	√	√	√	
Kedad-Sidhoum et al. (2008)	√		√		√	
Bigras et al. (2008)		√			√	
Pessoa et al. (2010)		√				
Correa and Schulz (2005)				√		√
Schulz and Uhan (2011)				√		√
Tang et al. (2007)		√				√
Davari et al. (2016)	√	√			√	√
Rostami et al. (2019)		√				√

for the weighted tardiness scheduling problem with precedence constraints, Tang et al. (2007) proposed a hybrid backward and forward DP-based LR to get the lower bound. Tanaka, S. & Sato, S. (2013) used SSDP to solve both the weighted tardiness and earliness-tardiness problem. Davari et al. (2016) proposed the B&B algorithm to solve the tardiness problem considering release time, idle time, deadlines. Rostami et al. (2019) proposed an exact DP approach with an efficient memory management technique.

All these references are summarized in Table 1. We specify the problem conditions: release time (r_j), idle time (IT), precedence constraints (PRE); the objective functions: tardiness (T_j), earliness-tardiness ($E_j + T_j$), completion time (C_j). We use the time-index formulation to model the test case sequencing process. Since our problem combines the idle time, precedence constraints and weighted earliness-tardiness penalties, all the above algorithms cannot be used directly to solve the test case sequencing process. Thus, we modify the CG method to calculate the lower bound of test case sequencing process. Moreover, the model proposed in Section 5 considers both the selection and scheduling problems and thus has a more general structure than the specific single machine scheduling problem. Note that the above CG method is also suitable for this model. Therefore, we first add the precedence constraints to the single machine scheduling problem and propose a CG algorithm to obtain the lower bound. Then, this algorithm is extended to solve a more general problem with test case selection and sequencing.

3. The statement of the problem

Table 2 shows the notations of the parameters we use in the following models.

The backbone router mainboard is an embedded system with hardware and software. Its reliability testing consists of hardware testing and software testing. In practical manufacture, the test cases of the software are executed with the associated hardware under accelerated environmental conditions. The test case in our problem denotes a class of the test cases, such as offline testing, online testing, built-in self-test (BIST), RAM test. All test cases are independent and executed to test certain functionality or component. Cycle testing is executed under environmental stresses and the weakest members exposed were recorded in history for analysis.

In our testing scheme, the reliability testing of the mainboard executes a few cycles to detect the failures shown in Table 3. During one cycle, L periods are implemented under different environmental conditions M, which concludes all possible conditions.

Table 2
Denotations of parameters in our problem.

Symbols of test cases attributes	
I	all test cases
i	test case $i \in I$
J	all periods in the testing process
j	period $j \in J$
Q	all cycles
N	number of the test cases, i.e. $N = \|I\|$
TC_{ij}	TC_{ij} means test case i in period j
p_{ij}	processing time of TC_{ij}
d_{ij}	due date, i.e. completion time of TC_{ij} in the original scheduling of the testing procedure
w_{ij}	number of failures in history exposed by TC_{ij}
SW	sparsity of the matrix of failure numbers w_{ij} in testing
Symbols of period attributes	
$T_j^{original}$	original testing time for each period $j \in J$
ρ	time reduction ratio
T_j	target time limit for each period $j \in J$, i.e. $T_j = \rho \cdot T_j^{original}$
z_{0j}	start time of each period $j \in J$
I_j^e	all critical test cases in each period $j \in J$
$\mathrm{B}_j,\ \Gamma_j, \Delta_j$	three given sets of test cases in each period $j \in J$, which are both collectively exhaustive and mutually exclusive, i.e. $\mathrm{B}_j \cup \Gamma_j \cup \Delta_j = I$, $\mathrm{B}_j \cap \Gamma_j = \mathrm{B}_j \cap \Delta_j = \Delta_j \cap \Gamma_j = \phi$.
$P_1,\ P_2,\ P_3$	priorities of the sets $\mathrm{B}_j \backslash I_j^e,\ \Gamma_j \backslash I_j^e,\ \Delta_j \backslash I_j^e$ in selection procedure individually
A_P^j	precedence relations among test cases in period $j \in J$, $A_P^j = \{(i, q) \mid TC_{ij} < TC_{qj},\ i,\ q \in I\}$
m	accelerated environmental condition, $m \in M$
M	all accelerated environmental conditions, $M = \{M_1, \ldots, M_L\}$
J_m	set of periods with the same accelerated condition $m \in M$
Symbols of decision variables	
$\tilde{x}_{ij} \in \{0,\ 1\}$	assignment variable. 1 states TC_{ij} is chosen, 0 otherwise.
$x_{it}^j \in \{0,\ 1\}$	assignment variable. 1 states TC_{ij} is chosen and starts at time $t \in \{1,\ 2,\ \ldots,\ T_j\}$ in period $j \in J$, 0 otherwise.

Table 3
An example for the testing environmental conditions and the failure numbers exposed by the test cases in testing process.

		Cycle 1				...	Cycle $\|Q\|$			
Period		1	2	...	L		$\|J\| - L + 1$	$\|J\| - L + 2$	...	$\|J\|$
Environmental condition		M_1	M_2	...	M_L		M_1	M_2	...	M_L
Test case	**Online testing**	0	3	...	1	...	0	5	...	4
	Offline testing	1	0	...	0		0	0	...	0
	BIST	0	0	...	0		2	0	...	0
	RAM test	0	0	...	0		0	2	...	0

There is a set of test cases executed in reliability testing. The index of each test case is denoted by $i \in I$. The test case $i \in I$ executed in period $j \in J$ is denoted by TC_{ij}. For example, in Table 3 "Online testing" is a test case, and "Online testing" combined with period j is a TC_{ij}. A list of test cases in each period is executed sequentially and the failures of the mainboard might be detected by certain test case(s). All failure times and corresponding test cases are recorded. The failure number w_{ij} denotes the number of failures exposed by TC_{ij} in history. A critical test case is one TC_{ij} that has exposed at least one failure in history. The sparsity SW of the matrix of failure numbers is calculated as the number of zero-valued elements divided by the total number of elements.

The goal is to reduce the reliability testing time $T_j^{original}$ to $\rho T_j^{original}$ for all periods, where ρ is the time reduction ratio ($0 < \rho \leq 1$). Since the sparsity of the failure number matrix in the testing process is high, we can select a subset of $\{TC_{ij},\ \forall i \in I,\ j \in J\}$ to execute to reduce the testing time. There are several constraints to consider when constructing this subset:

Constraint 1. For each period j, the sum of the processing times of all selected test cases must be within the time constraint $T_j = \rho T_j^{original}$.

Constraint 2. The critical test cases must be included in the subset to ensure the coverage of all historical failures.

Constraint 3. Each test case must be executed under all the environment conditions M at least once (not necessarily in the same cycle) to guarantee the testing integrity. This constraint also supports the non-execution of unselected test cases.

Constraint 4. One precedence constraint defines a partial ordering between a pair of two test cases: for $(i,\ q) \in A_P^j,\ TC_{ij} < TC_{qj}$, means that TC_{qj} cannot start before the completion of TC_{ij}. Moreover, for any $(i,\ q) \in A_P^j$, if TC_{qj} is selected to enter the subset, then TC_{ij} must also be selected to ensure the execution of TC_{qj}. For example, there are two test cases: CPU online testing and initialization for CPU online testing. The precedence relationship between them is “initialization for CPU online testing < CPU online testing”. If CPU online testing is selected to conduct, initialization for CPU online testing must be executed before it.

Constraint 5. The test cases have different degrees of importance in testing and they are pre-divided into three levels: $\mathrm{B}_j,\ \Gamma_j,\ \Delta_j$, by domain experts. The test cases at different levels are selected with different priorities $\{P_1, P_2, P_3\}$, respectivley.

After test case selection, a subset of test cases is obtained for each period. In the second step, test case sequencing is applied to optimize a specific goal and simultaneously ensure the precedence relations between certain test cases. According to the failure physics of the mainboard, the same type of failure is more likely to occur around the same time point of the testing process. Thus, we aim to schedule the critical test cases near their original time slots and the goal of test case sequencing is to optimize the total effectiveness of test case sequence. To assess the effectiveness of different sequences, we refer to the cost-cognizant weighted average percentage of faults detected measurement for software regression testing (Huang, Peng & Huang, 2012). The idea is to minimize the total deviation between the scheduled times and the originally planned times. The deviation weighted by the number of failures in historical records is used as the effectiveness index (1).

$$E^j = \frac{\sum_{i\in I} w_{ij}\left|t_{ij} - d'_{ij}\right|}{T_j} \tag{1}$$

The time when TC_{ij} is started is denoted as t_{ij} and its original completion time in historical records is d_{ij}. In (1) the objective aims at minimizing the difference between the starting time t_{ij} to be scheduled and the starting time of the critical test case $d'_{ij} = d_{ij} - p_{ij}$ in historical records. Then, the problem is formulated to schedule the selected test cases to minimize the objective E^j with precedence constraints.

Thus, we propose two models: two-step model and integrated model to reduce the testing time and to improve the testing effectiveness.

4. Two-step model

Two-step model first selects the subset of the test cases to satisfy the time constraint and then prioritizes the selected subset based on historical information to reach the highest effectiveness index E^j. In our previous paper (Zhang et al., 2020), we formulation the test case sequencing problem as a job-based formulation. In this work, we use a time-index formulation to build this model, since its bounds provided by the solutions of relaxation are very strong (Kedad-Sidhoum et al., 2008). In this section, we briefly describe the two-step model and then propose an efficient exact algorithm to solve it. The motivations and descriptions of the model are presented in Section 4.1 and 4.2. In particular, we first formulate the test case sequencing process into a time-index formulation in Section 4.2.1. We further introduce the exact algorithm to efficiently solve it in Section 4.2.2.

4.1. Test case selection

The test case selection technique is formulated as an integer problem, solved by a linear integer programming technique. The selection of TC_{ij} is indicated as decision variable $\tilde{x}_{ij}$.

$$\tilde{x}_{ij} = \begin{cases} 1, & \text{test case } TC_{ij} \text{ is chosen} \\ 0, & \text{otherwise} \end{cases}, \ \forall\ j \in J, i \in I$$

The subset of $\{TC_{ij},\ i \in I,\ j \in J\}$ is determined as follows.

$$\max \frac{P_1 \sum_{j\in J}\sum_{i\in \mathrm{B}_j\backslash I_j^e} \tilde{x}_{ij} + P_2 \sum_{j\in J}\sum_{i\in \Gamma_j\backslash I_j^e} \tilde{x}_{ij} + P_3 \sum_{j\in J}\sum_{i\in \Delta_j\backslash I_j^e} \tilde{x}_{ij}}{N} \tag{2}$$

$$\text{s.t.} \quad \sum_{i\in I} p_{ij}\tilde{x}_{ij} \le T_j \ \forall j \in J \tag{3}$$

$$\tilde{x}_{ij} = 1 \ \forall j \in J,\ i \in I_j^e \tag{4}$$

$$\sum_{j\in J_m} \tilde{x}_{ij} \ge 1 \ \forall i \in I,\ m \in M \tag{5}$$

$$\tilde{x}_{ij} \ge \tilde{x}_{qj} \ \forall j \in J,\ (i,\ q) \in A_P^j \tag{6}$$

$$\tilde{x}_{ij} \in \{0,\ 1\} \ \forall j \in J,\ i \in I\backslash I_j^e \tag{7}$$

The constraint (3) shows that the predetermined testing time limit must be satisfied in each period $j \in J$. The constraint (4) guarantees that all critical test cases in each period must be selected into the subset. The coverage constraint (5) states that all test cases must be executed under all different environmental conditions at least once, where J_m denotes the set of periods under the same accelerated condition $m \in M$. Constraint (6) states that the former one of the precedence constraint pair must be selected if the later one is selected into the subset. Other unselected test cases in each period can be assigned into the subset based on the predefined priorities for $\mathrm{B}_j\backslash I_j^e$, $\Gamma_j\backslash I_j^e$ and $\Delta_j\backslash I_j^e$. This selection priority of different sets is guaranteed by the objective function (2) with different priority factors P_i, and $P_1 < P_2 < P_3$.

4.2. Test case sequencing

After selecting the subset of test cases, the order of selected test cases should be scheduled as close to the original plan as possible to improve the failure detection rate and the testing effectiveness. This problem is similar to the single machine scheduling problem with total weighted earliness-tardiness. We use the time-index formulation to model our problem as a binary integer programming problem. Since the idle time is allowed and the precedence constraints should be considered in our problem, we modify the CG method to calculate the lower bound.

4.2.1. Time-index formulation of test case sequencing

The time-index formulation is based on time discretization, i.e. the whole time is divided into several units, where each unit t denotes starting from time $t-1$ to time t. The time duration in each period is T_j, which means that we can only schedule all test cases in time slot $\{1,\ 2,\ \ldots,\ T_j\}$. According to the time discretization, the scheduling time of TC_{ij} is denoted as the decision variable x_{it}^j, indicating whether TC_{ij} starts at time slot t.

$$x_{it}^j = \begin{cases} 1, & \text{if } TC_{ij} \text{ starts at time } t \\ 0, & \text{otherwise} \end{cases}, \quad \forall i \in I,\ j \in J,\ t \in \left\{1,\ 2,\ \ldots,\ T_j\right\}$$

In period j, the due date d_{ij} is within the range $[(j-1)T_j^{original},\ jT_j^{original}]$ and the start time decision variable t is in the interval $[(j-1)T_j,\ jT_j]$. For simplicity of the definition, we set the range of the decision variable t and the due dates d_{ij} as $[0,\ T_j]$ and $[(j-1)(T_j^{original} - T_j),\ jT_j^{original} - (j-1)T_j]$ respectively. That is, the starting time of period j is $z_{0j} = (j-1)(T_j^{original} - T_j)$, and the due dates of test cases in each period $j \in J$ can be calculated from the starting time z_{0j} other than the sum of the previous testing times. The model for each period j is constructed as follows (8)-(12).

$$\min \quad \frac{\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1} c_{it}^j x_{it}^j}{T_j} \tag{8}$$

$$\text{s.t.} \quad \sum_{t=1}^{T_j-p_{ij}+1} x_{it}^j = 1 \ \forall i \in I \tag{9}$$

$$\sum_{i\in I}\sum_{s=est_{ij}(t)}^{t} x_{is}^{j} \le 1,\ \forall t=1,\ 2,\ \ldots,\ T_j-p_{ij}+1 \quad (10)$$

$$\sum_{s=1}^{T_j-p_{ij}+1} sx_{is}^{j} \le \sum_{t=1}^{T_j-p_{qj}+1} tx_{qt}^{j} - p_{ij},\ \forall (i,\ q)\in A_P^j \quad (11)$$

$$x_{it}^{j}\in\{0,\ 1\},\ \forall i\in I,\ t=1,2,\ldots,\ T_j-p_{ij}+1 \quad (12)$$

Let the start cost c_{it}^{j} equal to $w_{ij}|t_{ij}-(d_{ij}-p_{ij})|$ corresponding to the variable x_{it}^{j}, where $d_{ij}-p_{ij}$ is the target start time of TC_{ij}. Therefore, the effectiveness index E^j in (1) can be formulated as (8) with decision variables x_{it}^{j} to render the cost of the schedule. Let define $est_{ij}(t)$ as $max\{1,\ t-p_{ij}+1\}$ which denotes the earliest start time of TC_{ij} if it is in process in time slot t. Assignment constraint (9) means each test case in period j must be scheduled exactly once. Constraint (10), referred to capacity/resource constraint, ensures that at most one test case is scheduled at any time. And the idle time is allowed in our problem. Inequality (11) states the precedence relation between these test cases, which serves to limit the order of certain test cases in a sequence. $\sum_{s=1}^{T_j-p_{ij}+1} sx_{is}^{j}$ and $\sum_{t=1}^{T_j-p_{qj}+1} tx_{qt}^{j}$ denote the start times of TC_{ij} and TC_{qj}, respectively. So, these constraints equal to $t_{ij}\le t_{qj}-p_{ij}, \forall (i,q)\in A_P^j$.

4.2.2. The CG method to solve the lower bound of test case sequencing

The number of constraints is more than $N+T_j$ and the size of the variables is greater than $N\cdot T_j$. Therefore, the CG algorithm with Dantzig-Wolf decomposition can reduce the number of constraints and obtain the lower bound. The algorithm (Van den Akker et al., 2000) is modified since we add the precedence constraints into the model. The Dantzig-Wolf decomposition has the following structure:

$$\min \quad cx$$

$$\text{s.t.} \quad Ax\ge b$$

$$x\in P$$

In the LP-relaxation, the feasible region with capacity constraints (10) and relaxations of (12) structures the polytope P and is represented with the extreme points. The extreme points are integral because the polytope P is totally unimodular, and these integral solutions are called pseudo-schedules, satisfying the capacity constraint while not assignment constraint. Let x^k denote the extreme points of $P=\{x|\sum_{i\in I}\sum_{s=est_{ij}(t)}^{t} x_{is}^{j}\le 1,\ x_{it}^{j}\ge 0,\ \forall t= 1,\ 2,\ \ldots,\ T_j-p_{ij}+1,\ i\in I\}$, and the master problem can be described as,

$$\min \quad \sum_{k=1}^{K}\left(\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1} c_{it}^{j}x_{it}^{jk}\right)\lambda_k \quad (13)$$

$$\text{s.t.} \quad \sum_{k=1}^{K}\left(\sum_{t=1}^{T_j-p_{ij}+1} x_{it}^{jk}\right)\lambda_k=1,\ \forall i\in I \quad (14)$$

$$\sum_{k=1}^{K}\left(\sum_{t=1}^{T_j-p_{qj}+1} tx_{qt}^{jk}-\sum_{s=1}^{T_j-p_{ij}+1} sx_{is}^{jk}\right)\lambda_k\ge p_{ij}, \forall (i,\ q)\in A_P^j \quad (15)$$

$$\sum_{k=1}^{K}\lambda_k=1 \quad (16)$$

$$\lambda_k\ge 0,\ \forall k=1,\ 2,\ \ldots,\ K \quad (17)$$

The corresponding pricing problem to master problem (13) – (17) is that,

$$\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1} c_{it}^{j}x_{it}^{jk}-\sum_{i=1}^{N}\pi_i\left(\sum_{t=1}^{T_j-p_{ij}+1} x_{it}^{jk}\right) + \sum_{(i,\ q)\in A_P^j}\sum_{t=1}^{T_j}\mu_{(i,\ q)}\cdot t\cdot\left(x_{it}^{jk}-x_{qt}^{jk}\right)-\alpha$$

where π_i denotes the dual variable associated with the ith constraint of (14), $\mu_{(i,\ q)}$ is the dual variable corresponding to the pair $(i,\ q)\in A_P^j$ in (15), and α is the dual variable of constraint (16). This formula can be rewritten as,

$$\bar{c}_k=\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1}\left(c_{it}^{j}-\pi_i+\sum_{q\in I}\mu_{(i,\ q)}\cdot t\cdot 1_{\{(i,\ q)\in A_P^j\}} - \sum_{q\in I}\mu_{(q,\ i)}\cdot t\cdot 1_{\{(q,\ i)\in A_P^j\}}\right)x_{it}^{jk}-\alpha \quad (18)$$

The pricing problem is to calculate whether $\bar{c}_k$ is nonnegative for all pseudo-schedules, which can be mapped into a graph with arc weight $c_{it}^{j}-\pi_i+\sum_{q\in I}\mu_{(i,\ q)}\cdot t\cdot 1_{\{(i,\ q)\in A_P^j\}}-\sum_{q\in I}\mu_{(q,\ i)}\cdot t\cdot 1_{\{(q,\ i)\in A_P^j\}}$ from node $(i,\ t)$ to node $(i,\ t+p_{ij})$. And this pricing problem is simplified into the shortest weighted path problem and can be solved with DP.

5. Integrated model

In this section, we construct the model to optimize the test case selection and sequencing at the same time. This principle is similar to the prioritization with a time constraint proposed in Krishnamoorthi and Mary (2009) for software regression testing. In the above study, the requirement of the ordering dependencies among the test cases is ignored and the solving algorithm is a heuristic method. Nonetheless, we take the precedence constraints into account and propose an exact algorithm to solve this problem.

We observe that there are conflicts between test case selection and test case sequencing. In test case selection, we prefer to select as many test cases as possible to increase our revenue within the limit time, which corresponds to the weighted sum of test cases selected. While in test case sequencing, we try to provide more possible time slots by decreasing selection to avoid the earliness and tardiness of the critical test case scheduling. So, we integrate these two models to consider the trade-off between the test case selection and test case sequencing. We can treat the integrated goal as the benefit function, where the selection revenue minus the sequencing penalty. The objective functions (2) and (8) are integrated as,

$$\frac{P_1\sum_{j\in J}\sum_{i\in B_j\backslash I_j^e}\tilde{x}_{ij}+P_2\sum_{j\in J}\sum_{i\in\Gamma_j\backslash I_j^e}\tilde{x}_{ij}+P_3\sum_{j\in J}\sum_{i\in\Delta_j\backslash I_j^e}\tilde{x}_{ij}}{N} - W_0\cdot\sum_{j\in J}\frac{\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1}c_{it}^{j}x_{it}^{j}}{T_j}$$

where parameter W_0 denotes the degree that we concern about the sequencing penalty relative to the selection revenue. The relationship between two decision variables $\tilde{x}_{ij}$ and x_{it}^{j} is $\tilde{x}_{ij}=\sum_{t=1}^{T_j-p_{ij}+1}x_{it}^{j}$.

Thus, the objective function can be rewritten as,

$$\sum_{j\in J}\sum_{t=1}^{T_j-p_{ij}+1}\left[-\sum_{i\in I_j^e}W_0\cdot\frac{c_{it}^j}{T_j}x_{it}^j+\sum_{i\in \mathrm{B}_j\setminus I_j^e}\left(\frac{P_1}{N}-W_0\cdot\frac{c_{it}^j}{T_j}\right)x_{it}^j\right.$$
$$\left.+\sum_{i\in \Gamma_j\setminus I_j^e}\left(\frac{P_2}{N}-W_0\cdot\frac{c_{it}^j}{T_j}\right)x_{it}^j+\sum_{i\in \Delta_j\setminus I_j^e}\left(\frac{P_3}{N}-W_0\cdot\frac{c_{it}^j}{T_j}\right)x_{it}^j\right]$$
$$\triangleq\sum_{j\in J}\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1}\tilde{c}_{it}^j x_{it}^j \tag{19}$$

where the cost with decision variable x_{it}^j is redefined with $\tilde{c}_{it}^j$ for simplification. And the goal is to maximize the objective function to ensure the effectiveness and efficiency of the testing process.

$$\tilde{c}_{it}^j=\begin{cases}-W_0\cdot\frac{c_{it}^j}{T_j} & when\ i\in I_j^e\\ \frac{P_1}{N}-W_0\cdot\frac{c_{it}^j}{T_j} & when\ i\in \mathrm{B}_j\setminus I_j^e\\ \frac{P_2}{N}-W_0\cdot\frac{c_{it}^j}{T_j} & when\ i\in \Gamma_j\setminus I_j^e\\ \frac{P_3}{N}-W_0\cdot\frac{c_{it}^j}{T_j} & when\ i\in \Delta_j\setminus I_j^e\end{cases},$$
$$\forall j\in J,\ i\in I, t\in\left\{1,\ 2,\ \dots,\ T_j-p_{ij}+1\right\} \tag{20}$$

5.1. Time-index formulation of integrated model

Test case selection model (2)–(7) and test case sequencing model (8)–(12) can be integrated with the transforming relationship between $\tilde{x}_{ij}$ and x_{it}^j. The model for the testing process is constructed as follows (21)–(29).

$$\max\ \sum_{j\in J}\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1}\tilde{c}_{it}^j x_{it}^j \tag{21}$$

$$\text{s.t.}\quad \sum_{t=1}^{T_j-p_{ij}+1}x_{it}^j=1,\ \forall j\in J,\ i\in I_j^e \tag{22}$$

$$\sum_{t=1}^{T_j-p_{ij}+1}x_{it}^j\le 1,\ \forall j\in J,\ i\in I\setminus I_j^e \tag{23}$$

$$\sum_{i\in I}p_{ij}\sum_{t=1}^{T_j-p_{ij}+1}x_{it}^j\le T_j,\ \forall j\in J \tag{24}$$

$$\sum_{j\in J_m}\sum_{t=1}^{T_j-p_{ij}+1}x_{it}^j\ge 1,\ \forall m\in M,\ i\in I \tag{25}$$

$$\sum_{i\in I}\sum_{s=est_{ij}(t)}^{t}x_{is}^j\le 1,\ \forall j\in J,\ t=1,\ 2,\ \dots,\ T_j-p_{ij}+1 \tag{26}$$

$$\sum_{s=1}^{T_j-p_{ij}+1}x_{is}^j\ge\sum_{t=1}^{T_j-p_{qj}+1}x_{qt}^j,\ \forall j\in J,\ (i,\ q)\in A_p \tag{27}$$

$$-\bar{M}\left(1-\sum_{s=1}^{T_j-p_{ij}+1}x_{is}^j\right)+\sum_{s=1}^{T_j-p_{ij}+1}sx_{is}^j+p_{ij}\le M\left(1-\sum_{t=1}^{T_j-p_{qj}+1}x_{qt}^j\right)$$
$$+\sum_{t=1}^{T_j-p_{qj}+1}tx_{qt}^j,\qquad \forall j\in J,\ (i,\ q)\in A_p \tag{28}$$

$$x_{it}^j\in\{0,\ 1\},\ \forall j\in J,\ i\in I,\ t=1,2,\dots,\ T_j-p_{ij}+1 \tag{29}$$

The objective function (19) is simplified represented in (21). Assignment constraints (22) and (23) ensure all critical test cases must be included in the subset selection and all test cases can be selected into the subset. Constraint (24) guarantees that the total processing time of each period is no more than the predetermined limit. Inequality (25) states the coverage constraints for all different environmental conditions. Constraint (26) is the capacity constraint, ensuring that at most one test case is scheduled at any time. And the idle time is allowable in every period. Constraint (27) ensures that the former of the precedence constraint pair must be selected if the latter is in the subset. The precedence constraint (11) is modified since the test case(s) in the precedence pairs might be unselected. For example, for the precedence pair $(i,\ q)\in A_P^j$, TC_{ij} and TC_{qj} are considered. Once one of them is unselected, the constraint (11) $\sum_{s=1}^{T_j-p_{ij}+1}sx_{is}^j\le\sum_{t=1}^{T_j-p_{qj}+1}tx_{qt}^j-p_{ij}$ is unreasonable. In constraint (28), we add the selection incidences of TC_{ij} and TC_{qj} combined with the big enough number M and $\bar{M}$. If TC_{ij} is selected while TC_{qj} is unselected, this inequality equals to $\sum_{s=1}^{T_j-p_{ij}+1}sx_{is}^j+p_{ij}\le M$. The starting time of the selected TC_{ij} is not restricted. If neither of them is selected, this constraint equals to $-\bar{M}+p_{ij}\le M$, which is always satisfied. And this constraint is simplified to the constraint (11) when they are both selected. To make the feasible region more compact, the value of M is set as $\max T_j+1$ and $\bar{M}$ is defined as $\max p_{ij}$.

5.2. The CG method to solve the lower bound of integrated model

The time-index formulation model (21) - (29) also uses the Dantzig-Wolfe decomposition to reduce the size of the constraints with a huge number of variables, and the CG method is applied to deal with the size of variables and get the lower bound of the integrated model. Like in Section 4, the capacity constraint (26) and the relaxation of constraint (29) structure the polytope P, i.e., $P=\{x\ |\sum_{i\in I}\sum_{s=est_{ij}(t)}^{t}x_{is}^j\le 1,\ \ x_{it}^j\ge 0,\ \ \forall j\in J,\ \ i\in I,\ \ t=1,2,\dots,\ \ T_j-p_{ij}+1\}$. The feasible region P can be represented by the convex hull of its extreme points $\{x^k\}$, that is, for $x\in P$, $x=\sum_k\lambda_k x^k$ with $\sum_k\lambda_k=1$, $\lambda_k\ge 0,\forall k$. The master problem of the model (21)-(29) is,

$$\max\ \sum_{k=1}^{K}\left(\sum_{j\in J}\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1}\tilde{c}_{it}^j x_{it}^{jk}\right)\lambda_k \tag{30}$$

$$\text{s.t.}\quad \sum_{k=1}^{K}\left(\sum_{t=1}^{T_j-p_{ij}+1}x_{it}^{jk}\right)\lambda_k=1,\ \forall j\in J,\ i\in I_j^e \tag{31}$$

$$\sum_{k=1}^{K}\left(\sum_{t=1}^{T_j-p_{ij}+1}x_{it}^{jk}\right)\lambda_k\le 1,\ \forall j\in J,\ i\in I\setminus I_j^e \tag{32}$$

$$\sum_{k=1}^{K}\left(\sum_{i\in I}p_{ij}\cdot\sum_{t=1}^{T_j-p_{ij}+1}x_{it}^{jk}\right)\lambda_k\le T_j,\ \forall j\in J \tag{33}$$

$$-\sum_{k=1}^{K}\left(\sum_{j\in J_m}\sum_{t=1}^{T_j-p_{ij}+1}x_{it}^{jk}\right)\lambda_k\le -1,\ \forall m\in M,\ i\in I \tag{34}$$

$$\sum_{k=1}^{K}\left(\sum_{t=1}^{T_j-p_{qj}+1}x_{qt}^j-\sum_{s=1}^{T_j-p_{ij}+1}x_{is}^j\right)\lambda_k\le 0,\ \forall j\in J,\ (i,\ q)\in A_P^j \tag{35}$$

$$\sum_{k=1}^{K}\left(\sum_{s=1}^{T-p_{ij}+1}\left(\bar{M}+s\right)x_{is}^{jk}+\sum_{t=1}^{T-p_{qj}+1}(M-t)x_{qt}^{jk}\right)\lambda_k \le M+\bar{M}-p_{ij},$$
$$\forall j\in J,\ (i,\ q)\in A_P^j \tag{36}$$

$$\sum_{k=1}^{K}\lambda_k = 1 \tag{37}$$

$$\lambda_k \ge 0,\ \forall k=1,\ 2,\ \ldots,\ K \tag{38}$$

Corresponding to the master problem (30)–(38), the pricing problem is that,

$$-\sum_{j\in J}\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1}\tilde{c}_{it}^{j}x_{it}^{jk}+\sum_{j\in J}\sum_{i\in I}\pi_{i,\ j}\left(\sum_{t=1}^{T_j-p_{ij}+1}x_{it}^{jk}\right)$$
$$+\sum_{j\in J}\xi_j\left(\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1}p_{ij}x_{it}^{jk}\right)$$
$$-\sum_{m\in M}\sum_{i\in I}\nu_{m,\ i}\left(\sum_{j\in J_m}\sum_{t=1}^{T_j-p_{ij}+1}x_{it}^{jk}\right)$$
$$-\sum\nolimits_{j\in J}\sum\nolimits_{(i,\ q)\in A_P^j}\theta_{(i,\ q)}^{j}\sum\nolimits_{t=1}^{T_j-p_{ij}+1}x_{it}^{jk}$$
$$+\sum\nolimits_{j\in J}\sum\nolimits_{(q,\ i)\in A_P^j}\theta_{(q,\ i)}^{j}\sum\nolimits_{t=1}^{T_j-p_{ij}+1}x_{it}^{jk}$$
$$+\sum_{j\in J}\sum_{(i,\ q)\in A_P^j}\mu_{(i,\ q)}^{j}\left(\sum_{t=1}^{T_j-p_{ij}+1}\left(\bar{M}+t\right)x_{it}^{jk}\right)$$
$$+\sum_{j\in J}\sum_{(q,i)\in A_P^j}\mu_{(q,\ i)}^{j}\left(\sum_{t=1}^{T_j-p_{ij}+1}(M-t)x_{it}^{jk}\right)+\alpha$$

where $\pi_{i,j}$ denotes the dual variable associated with the constraints (31) and (32) for TC_{ij}. ξ_j is the dual variable corresponding to the jth constraint of (33). $\nu_{m,\ i}$ is the dual variable for the constraint (34) ensuring the test case i to cover the environment condition J_m. $\theta_{(i,\ q)}^{j}$ and $\mu_{(i,\ q)}^{j}$ are the dual variable related to the pair $(i,\ q)\in A_P^j$ in (35) and (36), and α is the dual variable of constraint (37). This formula can be written as,

$$\bar{c}_k=\alpha+\sum_{j\in J}\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1}x_{it}^{jk}\cdot\left[-\tilde{c}_{it}^{j}+\pi_{i,j}+\xi_j p_{ij}-\sum_{m\in M}\nu_{m,i}\cdot 1_{\{m:j\in J_m\}}\right.$$
$$+\sum_{q\in I}\left(\mu_{(i,\ q)}^{j}\left(\bar{M}+t\right)-\theta_{(i,\ q)}^{j}\right)\cdot 1_{\left\{(i,\ q)\in A_P^j\right\}}$$
$$\left.-\sum_{q\in I}\left(\mu_{(q,\ i)}^{j}(M-t)+\theta_{(q,\ i)}^{j}\right)\cdot 1_{\left\{(q,\ i)\in A_P^j\right\}}\right]$$
$$\triangleq\sum\nolimits_{j\in J}\bar{c}_k^j+\alpha \tag{39}$$

where $\bar{c}_k^j$ denotes $\sum_{i\in I}\sum_{t=1}^{T_j-p_{ij}+1}(-\tilde{c}_{it}^{j}+\pi_{i,\ j}+\xi_j p_{ij}-\sum_{m\in M}\nu_{m,i}\cdot 1_{\{m:j\in J_m\}}+\sum_{q\in I}(\mu_{(i,\ q)}^{j}(\bar{M}+t)-\theta_{(i,\ q)}^{j})\cdot 1_{\{(i,\ q)\in A_P^j\}}-\sum_{q\in I}(\mu_{(q,\ i)}^{j}(M-t)+\theta_{(q,\ i)}^{j})\cdot 1_{\{(q,\ i)\in A_P^j\}})$. The pricing problem is to calculate whether $\bar{c}_k$ is nonnegative for all pseudo-schedules in whole testing periods, which can be mapped into a graph with arc weight $-\tilde{c}_{it}^{j}+\pi_{i,\ j}+\xi_j p_{ij}-\sum_{m\in M}\nu_{m,i}\cdot 1_{\{m:j\in J_m\}}+\sum_{q\in I}(\mu_{(i,\ q)}^{j}(\bar{M}+t)-\theta_{(i,\ q)}^{j})\cdot 1_{\{(i,\ q)\in A_P^j\}}-\sum_{q\in I}(\mu_{(q,\ i)}^{j}(M-t)+\theta_{(q,\ i)}^{j})\cdot 1_{\{(q,\ i)\in A_P^j\}}$ from node $(i,\ t)^j$ to node $(i,\ t+p_{ij})^j$. The shortest weighted path problem $\bar{c}_k$ can be decomposed into $|J|$ subproblems $\bar{c}_k^j,\ \forall j\in J$ and each sub-problem is solved with DP. The optimal solution of the two-step model is used as the initial solution of the CG algorithm to produce the lower bound.

6. Computational experiment and case study

In this section, we present the numerical studies and compare the performance of the above two models. In Section 6.1, the two-step model is solved with the efficient exact approach we proposed. We further demonstrate the effects of two parameters on the structural properties of this model, that is, the time decreasing ratio ρ and the sparsity SW of the matrix of failure numbers. In Section 6.2, we present a case study modified from the real-world problem to compare the integrated model with the two-step model and analyze the effect of the parameter W_0 on the performance.

6.1. Computation experiment

We set that the reliability testing has two cycles and eight periods, and the accelerated environmental conditions in each cycle are different. There are 20 test cases during the testing process. The precedence constraints are $A_P^j=\{TC_1<TC_i,\ \forall i\neq 1, i\in I\},\forall\ j\in J$. The given testing sets are $B_j=\{3,\ 4,\ 5,\ 6\}$, $\Gamma_j=\{7,\ 8,\ 9,\ 10\}$, $\Delta_j=I\backslash(B_j\cup\Gamma_j),\ \forall\ j\in J$, the priority factors for these three sets $P_1,\ P_2,\ P_3$ are set as 3, 2, 1 respectively. The processing time of each test case is generated from the integer uniform distribution U[1, 50], and the same test cases in different periods have the same processing times for simplicity, i.e., $p_{ij}=p_{ik},\ \forall j,k\in J$. Then the original time for each period is $T_j^{original}=\sum_{i\in I}p_{ij},\ \forall j\in J$ and the due date d_{ij} of TC_{ij} can be obtained by summing up the processing times of all previous test cases in sequence. The time decreasing ratio is denoted as ρ, and the total processing time of all periods is reduced by equal multiples, so the time limit for each period is $T_j=\lfloor\rho\cdot T_j^{original}\rfloor$. The number of failures w_{ij} in history exposed by TC_{ij} is generated from the integer uniform distribution U[1, 10]. And the sparsity of w_{ij} is denoted as SW, which states that there are $\lfloor(1-SW)\cdot(8\cdot 20)\rfloor$ critical test cases detecting the failures in the historical records.

The structural property of the model is associated with two parameters: the time decreasing ratio ρ and the sparsity SW of the matrix of failure numbers. To analysis the effects of these two parameters on the computational complexity of the two-step model, we compare the computation time by computing the lower bound of the model with different values of $\rho=0.7,\ 0.8,\ 0.9$ and $SW=0.5,\ 0.7,\ 0.9$. Then, 50 instances are generated for each combination of $(\rho,\ SW)$, which consists of 450 instances. This information is represented as R20.50.10.$(10*SW)$.$(10*\rho)$, corresponding to the number of test cases, the maximal processing time, the upper bound of w_{ij}'s distribution, sparsity and decreasing time ratio. And these instances as denoted as R20.50.10.$(10*SW)$.$(10*\rho)$.m, where m is the number of instances. The computation has been performed on a computer with Core i5 7200 (2.5 gigahertz) CPU and 8 gigabytes RAM and the experiments have been implemented with python 3.6/CPLEX 12.8.

As for any branch-and-price algorithm, the computation times of two different instances generated by the same set of parameters may differ considerably (Sourd & Kedad-Sidhoum, 2008). Therefore, the average computation time is not relevant and we prefer to use the distribution of computation time to present our results. Fig. 1 illustrates such distributions, where for example, Fig. 1(a) contains the curves whose x-axis value is the CPU computation time t and y-axis value is the percentage of the instances solved within the time t. To illustrate the influence of ρ and SW on computational complexity of two-step model, Fig. 1 and Fig. 2 com-

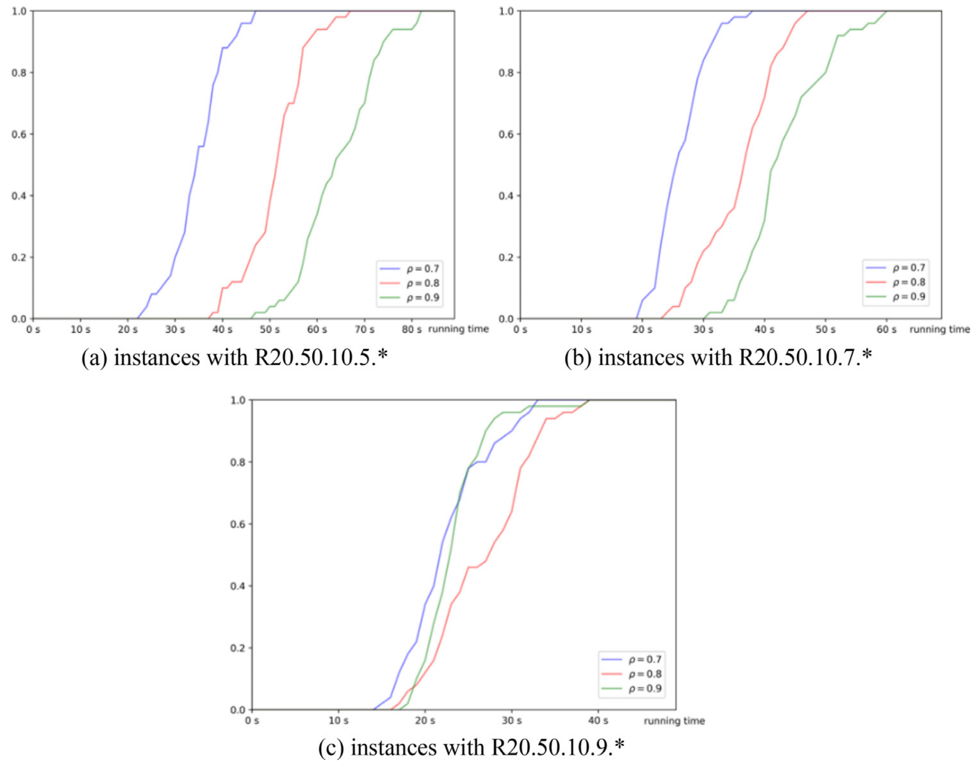

Fig. 1. Computation time distributions with ρ varying given different *SW*.

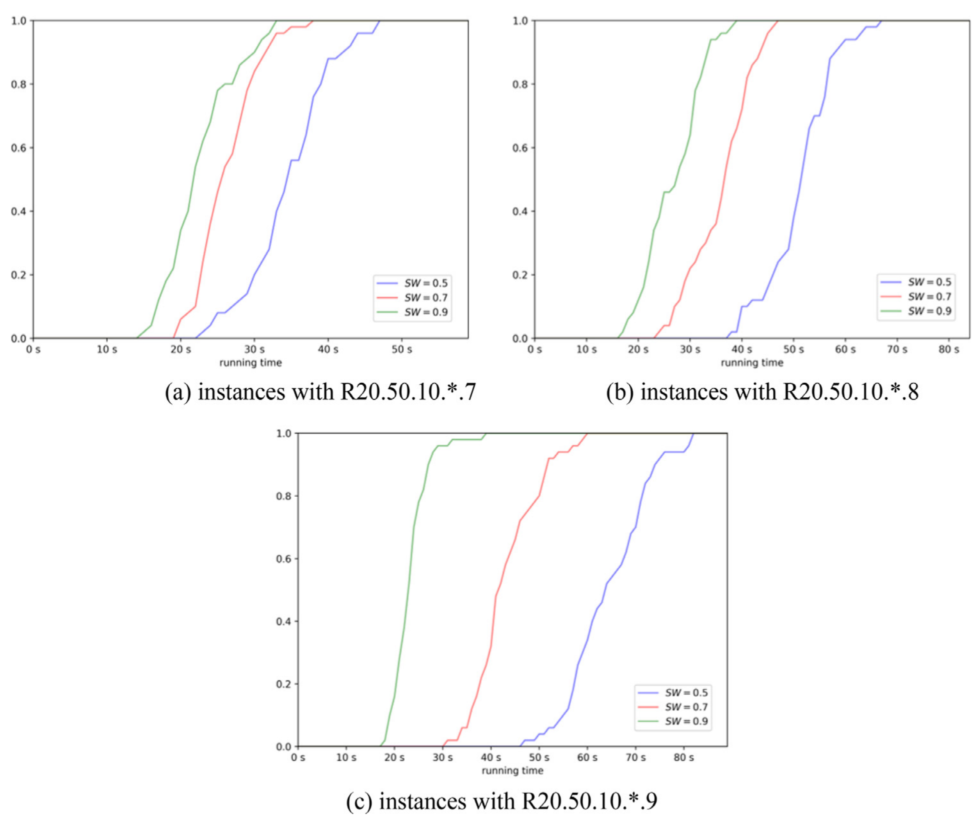

Fig. 2. Computation time distributions with *SW* varying given different ρ.

pares the computation time distribution to solve the instances for different values of ρ and SW, respectively.

Fig. 1 shows the computation time distribution under different ρ. We fix the parameter SW and vary ρ among the set $\{0.7,\ 0.8,\ 0.9\}$. That is, the instances R20.50.10.5.*, R20.50.10.7.* and R20.50.10.9.* are considered under different ρ respectively. As shown in Fig. 1(a) and (b), when $SW = 0.5$ or 0.7, the computation time to obtain a solution increases with ρ increasing. It indicates that the problem is generally more complicated when ρ is bigger. There are two possible reasons for this characteristic. For the first reason, as ρ increases, the reliability testing time $\sum_{j\in J} T_j$ increases, and the number of test cases selected in test case selection procedure also increases. Secondly, the range of time decision variables in model (8) - (12) is also monotonically increasing with ρ. Both these two reasons result that the number of decision variables x_{it}^{j} increases with ρ increasing. Therefore, the problem gets difficult with ρ increasing. However, we observe that in Fig. 1(c) the computation time of R20.50.10.9.9 (i.e. $\rho = 0.9$) is generally less than R20.50.10.9.8 (i.e. $\rho = 0.8$), which does not match the characteristic mentioned above. It seems that the number of decision variables may not be the major factor influencing the problem complexity when the matrix of failure numbers is sparse enough. The instances with R20.50.10.9.9 (or R20.50.10.9.8) mean that the sparsity is 0.9 and the reliability testing time is decreased by 10% (or 20%). When the failure number matrix is sparse enough, the critical test cases are more easily to be scheduled around the target time slots. The major factor affecting the model complexity might be whether the starting times $\{0, T_j,\ 2T_j, \ldots\}$ of all periods in scheduled reliability testing process are not distant from their original starting times $\{0,\ T_j^{original}, 2\ T_j^{original}, \ldots\}$. When $\rho = 0.9$, the scheduled starting times are closer to the original starting times than when $\rho = 0.8$, since $T_j^{original} = \rho T_j$.

Fig. 2 compares the computation time distribution for instances with different SW varying among $\{0.5,\ 0.7,\ 0.9\}$. We observe that the computation time to solve the problem increases with SW decreasing. It indicates that the problem is harder to solve with lower sparsity SW of the matrix of failure numbers. The objective in (8) is actually to schedule all critical test cases to expected positions as much as possible. The number of critical test cases increases as SW decreasing. Therefore, it is obvious that when the parameter SW decreases, more critical test cases are expected to be scheduled to minimize the sequencing objective function and the model is becoming more complicated.

6.2. Case study

The practical case study of our industrial partner, a multinational telecommunication equipment provider, is presented in this section. The environmental conditions used in practice are temperature and voltage, which alternate between two extremes to apply the mechanical stresses to the mainboard. Combine these two accelerated factors: high/low temperature (HT/LT) and high/low voltage (HV/LV), thus all environmental conditions are $M = \{\text{HTHV},\ \text{HTLV},\ \text{LTHV},\ \text{LTLV}\}$. Reliability testing has two cycles and eight periods. The test programs are blocked into L test cases and thus $8L$ TCs are considered in the reliability testing. Due to the confidentiality reasons, the real data about the mainboard testing process was not disclosed. Instead, artificial numbers are used to illustrate the two models and compare the performances.

We generated the instances with the number of test cases during the testing process as $N \in \{4,\ 5\}$, since the integrated model takes into account the test cases for all periods simultaneously which incorporates the selection and sequencing. The precedence constraints are $A_P^j = \{TC_1 < TC_i,\ \forall i \neq 1, i \in I\}, \forall\ \ j \in J$. The given testing sets are $\mathrm{B}_j = \{1,\ 2\},\ \Gamma_j = \{3\},\ \Delta_j = I \backslash (\mathrm{B}_j \cup \Gamma_j),\ \ \forall\ \ j \in J$,

Table 4
The objective value for two-step model and integrated model.

Instance	Two-step model						Integrated model					
	$W_0 = 0.2$		$W_0 = 0.4$		$W_0 = 0.6$		$W_0 = 0.2$		$W_0 = 0.4$		$W_0 = 0.6$	
	Selection & sequencing obj.	Integrated obj.	Selection & sequencing obj.	Integrated obj.	Selection & sequencing obj.	Integrated obj.	Selection & sequencing obj.	Integrated obj.	Selection & sequencing obj.	Integrated obj.	Selection & sequencing obj.	Integrated obj.
R4.10.10.7.8.1	9.25	−2.93	9.25	−15.11	9.25	−27.29	**9.25**	**−2.63**	**7.25**	**−14.23**	**7.25**	**−24.97**
	60.9		60.9		60.9		**59.4**		**53.7**		**53.7**	
R4.10.10.7.8.2	10.25	3.45	10.25	−3.35	10.25	−10.15	**10.25**	**3.65**	**10.25**	**−2.95**	**10.25**	**−9.55**
	34		34		34		**33**		**33**		**33**	
R4.10.10.7.8.3	10.25	5.5206	10.25	0.7912	10.25	−3.9382	10.25	5.5206	**8.75**	**1.5971**	**8.0**	**−1.8824**
	23.6471		23.6471		23.6471		23.6471		**17.8824**		**16.4706**	
R4.10.10.7.8.4	10.25	5.9643	10.25	1.6786	10.25	−2.6071	**10.25**	**6.9357**	**10.25**	**3.6214**	**8.75**	**0.6929**
	21.4286		21.4286		21.4286		**16.5714**		**16.5714**		**13.4286**	
R4.10.10.7.8.5	7.25	−2.0	7.25	−11.25	7.25	−20.5	**7.25**	**−1.8875**	**7.25**	**−11.025**	**7.25**	**−20.1625**
	46.25		46.25		46.25		**45.6875**		**45.6875**		**45.6875**	
R5.10.10.7.8.1	9.8	−6.1652	9.8	−22.1304	9.8	−38.0957	**9.8**	**−6.1130**	**9.8**	**−22.0261**	**9.8**	**−37.9391**
	79.8261		79.8261		79.8261		**79.5652**		**79.5652**		**79.5652**	
R5.10.10.7.8.2	8.0	−2.4947	8.0	−12.9894	8.0	−23.4842	**8.0**	**−2.4316**	**7.4**	**−12.2842**	**7.2**	**−22.0737**
	52.4737		52.4737		52.4737		**52.1579**		**49.2105**		**48.7895**	
R5.10.10.7.8.3	9.6	−6.06	9.6	−21.72	9.6	−37.38	9.6	−6.06	**6.0**	**−20.28**	**5.6**	**−33.28**
	78.3		78.3		78.3		78.3		**65.7**		**64.8**	
R5.10.10.7.8.4	5.4	−5.6857	5.4	−16.7714	5.4	−27.8571	5.4	−5.6857	5.4	−16.7714	5.4	−27.8571
	55.4286		55.4286		55.4286		55.4286		55.4286		55.4286	
R5.10.10.7.8.5	10.2	−5.1125	10.2	−20.425	10.2	−35.7375	**10.2**	**−4.9375**	**10.2**	**−20.075**	**8.4**	**−34.425**
	76.5625		76.5625		76.5625		**75.6875**		**75.6875**		**71.375**	

Table 5
The deviations of integrated objective values between two-step model and integrated model under different W_0.

Instance	$W_0 = 0.2$	$W_0 = 0.4$	$W_0 = 0.6$
R4.10.10.7.8.1	0.3	0.88	2.32
R4.10.10.7.8.2	0.2	0.4	0.6
R4.10.10.7.8.3	0	0.8059	2.0559
R4.10.10.7.8.4	0.9714	1.9429	3.3
R4.10.10.7.8.5	0.1125	0.225	0.3375
R5.10.10.7.8.1	0.0522	0.1043	0.1565
R5.10.10.7.8.2	0.0632	0.7053	1.4105
R5.10.10.7.8.3	0	1.44	4.1
R5.10.10.7.8.4	0	0	0
R5.10.10.7.8.5	0.175	0.35	1.3125

the priority factors for these three sets P_1, P_2, P_3 are set as 3, 2, 1 respectively. Like the random instance generation method in Section 5.1, the processing time of each test case is uniformly distributed in [1, 10], and the same test cases in different periods have the same processing times. The time decreasing ratio ρ is set as 0.8. The number of failures w_{ij} is uniformly distributed in [1, 10]. And the sparsity of w_{ij} is set as $SW = 0.7$. Five instances are generated for both $N = 4$ and $N = 5$. The trade-off parameter W_0 in the objective function of integrated model varies in $\{0.2, 0.4, 0.6\}$. Ten different types of mainboard are studied and all these historical records are complete. The two-step method and integrated model are implemented and analyzed. The optimal solutions are solved by two-step model and integrated model, where CG methods we proposed are used to obtain the lower(upper) bound and branch-and-price algorithm is implemented to reach the integer optimal solution. Then, the selection, sequencing and integrated objectives of these two models are obtained with the optimal solutions.

Table 4 presents that the optimal solutions solved by the two-step model are compared with the solutions by the integrated model, including the selection objective values (2), sequencing objective values (8) and integrated objective values (21). We observe that the sequencing objective and the integrated objective value of the integrated model outperform the two-step model and the integrated model improves the sequencing and integrated objective values by sacrificing the selection value. For instance R5.10.10.7.8.2, when $W_0 = 0.4/0.6$, the selection objective value of the integrated model is worse than the two-step model while its integrated objective value is better. The integrated model increases the integrated revenue by paying more attention to the sequencing penalty. When $W_0 = 0.2$, we observe the selection objective values of the integrated model are the same as the two-step model while the sequencing objective values are not. This is because there are multiple optimal solutions in test case selection process and a different optimal solution in selection is obtained to get a better integrated value. Furthermore, the sequencing value is decreasing with the parameter W_0 increasing, which means we care more about the sequencing penalty of the critical test cases. The strictly greater solutions of the integrated objective value are shown in bold. As shown in Table 5, if we are more concerned about the effects of sequencing penalty, i.e. W_0 is increasing, the outperformance of the integrated model becomes more obvious.

6.3. Comparison of integrated model with time-aware TCP scheme

We compare two schemes of reliability testing: our proposed reliability testing scheme and the scheme with time-aware TCP technique proposed in Walcott et al. (2006) and Krishnamoorthi and Mary (2009). The time-aware TCP scheme aims to optimize the regression software testing under the testing time constraint. Since they did not consider the preferred execution sequence of test cases, we omit the precedence constraints and consider the constraint 1 - 3 and 5 to compare their performances. The time-aware TCP scheme is adjusted to our required fault-based objective (21) and other constraints. The integrated model is applied with (21) - (26) and (29). The instances (R4.10.10.7.8.1 - R4.10.10.7.8.5 and R5.10.10.7.8.1 – R5.10.10.7.8.5) and parameter settings of reliability testing in Section 6.2 are used in this experiment except the precedence constraints. We apply the time-aware TCP scheme and the integrated model for these ten instances and obtain the optimal integrated objective values (21) under different W_0 shown in Table 6. It is obvious that the objective values of integrated model are no less than time-aware TCP scheme and our proposed integrated model outperforms the time-aware TCP scheme. The reason is that the idle time during the reliability testing is allowed in our integrated model while forbidden in the time-aware TCP scheme. The idle time between test cases is helpful to improve the effectiveness of failure detections.

7. Conclusions

In conclusion, first the two-step model is reformulated. Then, the efficiently exact algorithm is proposed to solve the large-scale problem. Numerical experiments showed that the parameters: decreasing time ratio ρ and sparsity SW may influence the model complexity, which is measured by the computation time of the lower bound. The sparsity has a negative effect on the computation time. The decreasing time ratio increases the computation time when the sparsity is small, while the effect is opposite when the matrix of failure numbers is sparse enough. We also constructed the integrated model optimizing the test case selection and test case sequencing simultaneously, which produces the optimally selected and scheduled subsets of test cases to optimize the effectiveness of the testing process with the precedence constraints and reduce the testing time to the desired level. The exact algorithm is also proposed based on the branch-and-price technique. A case

Table 6
The comparison of objective value (21) for integrated model and time-aware TCP scheme under different W_0.

Instance	time-aware TCP			Integrated model		
	$W_0 = 0.2$	$W_0 = 0.4$	$W_0 = 0.6$	$W_0 = 0.2$	$W_0 = 0.4$	$W_0 = 0.6$
R4.10.10.7.8.1	0.51	−8.23	−16.97	1.01	−7.23	−15.47
R4.10.10.7.8.2	5.87	1.49	−2.89	6.27	2.2900	−1.69
R4.10.10.7.8.3	8.8853	7.5206	6.1559	9.0029	7.7559	6.5088
R4.10.10.7.8.4	8.65	7.05	5.5429	8.8214	7.3929	5.9643
R4.10.10.7.8.5	−0.1625	−7.575	−14.9875	0.725	−5.8	−12.325
R5.10.10.7.8.1	−4.3913	−18.5826	−32.7739	−3.1652	−16.1304	−29.0957
R5.10.10.7.8.2	−0.0211	−8.0421	−16.0632	−0.0211	−8.0421	−16.0632
R5.10.10.7.8.3	0	−9.6	−19.2	0.33	−8.94	−18.21
R5.10.10.7.8.4	−4.4	−14.2	−24	−2.5143	−10.4286	−18.3429
R5.10.10.7.8.5	−1.05	−12.3	−23.55	−0.5	−11.2	−21.9

study is presented to demonstrate that the integrated model outperforms the two-step model and the improvement is more evident when the sequencing part has greater weight in the integrated objective.

The objectives of test case selection and test case sequencing are competing and conflicting. The proposed integrated model conflates these two objectives into a single objective with a trade-off parameter. In future work, we can consider a Pareto optimization approach for these conflicting objectives.

Acknowledgments

The authors thank the anonymous referees for their valuable suggestions that significantly improved this paper.

Funding

This research was supported by the National Natural Science Foundation of China under a key project grant No. 71731008 and Natural Science Foundation of Beijing Municipality under Grant No. L191022.

References

Acevedo, P. E., Jackson, D. S., & Kotlowitz, R. W. (2006). Reliability growth and forecasting for critical hardware through accelerated life testing. *Bell Labs Technical Journal, 11*(3), 121–135.

Bigras, L. P., Gamache, M., & Savard, G. (2008). Time-indexed formulations and the total weighted tardiness problem. *INFORMS Journal on Computing, 20*(1), 133–142.

Chen, P., Xu, A., & Ye, Z. (2016). Generalized fiducial inference for accelerated life tests with Weibull distribution and progressively type-II censoring. *IEEE Transactions on Reliability, 65*(4), 1737–1744.

Chittimalli, P. K., & Harrold, M. J. (2009). Recomputing coverage information to assist regression testing. *IEEE Transactions on Software Engineering, 35*(4), 452–469.

Choi, U. M., Joergensen, S., & Blaabjerg, F. (2016). Advanced accelerated power cycling test for reliability investigation of power device modules. *IEEE Transactions on Power Electronics, 31*(12), 1 –1.

Correa, J. R., & Schulz, A. S. (2005). Single-machine scheduling with precedence constraints. *Mathematics of Operations Research, 30*(4), 1005–1021.

Davari, M., Demeulemeester, E., Leus, R., & Nobibon, F. T. (2016). Exact algorithms for single-machine scheduling with time windows and precedence constraints. *Journal of Scheduling, 19*(3), 309–334.

Escobar, L. A., & Meeker, W. Q. (2006). A review of accelerated test models. *Statistical Science, 21*(4), 552–577.

Han, D. (2015). Time and cost constrained optimal designs of constant-stress and step-stress accelerated life tests. *Reliability Engineering & System Safety, 140*, 1–14.

Hao, D., Zhang, L., Zang, L., Wang, Y., Wu, X., & Xie, T. (2016). To be optimal or not in test-case prioritization. *IEEE Transactions on Software Engineering, 42*(5), 490–505.

Hu, C. H., Plante, R. D., & Tang, J. (2015). Equivalent step-stress accelerated life tests with log-location-scale lifetime distributions under Type-I censoring. *IIE Transactions, 47*(3), 245–257.

Huang, Y. C., Peng, K. L., & Huang, C. Y. (2012). A history-based cost-cognizant test case prioritization technique in regression testing. *Journal of Systems and Software, 85*(3), 626–637.

Kedad-Sidhoum, S., Solis, Y. R., & Sourd, F. (2008). Lower bounds for the earliness–tardiness scheduling problem on parallel machines with distinct due dates. *European Journal of Operational Research, 189*(3), 1305–1316.

Krishnamoorthi, R., & Mary, S. A. (2009). Regression test suite prioritization using genetic algorithms. *International Journal of Hybrid Information Technology, 2*(3), 35–52.

Lawler, E. L. (1978). Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Annals of Discrete Mathematics, 2*, 75–90.

Liu, X., & Tang, L. C. (2013). Planning accelerated life tests under scheduled inspections for log-location-scale distributions. *IEEE Transactions on Reliability, 62*(2), 515–526.

Mirarab, S., Akhlaghi, S., & Tahvildari, L. (2012). Size-constrained regression test case selection using multicriteria optimization. *IEEE Transactions on Software Engineering, 38*(4), 936–956.

Nelson, W. B. (2005). A bibliography of accelerated test plans. *IEEE Transactions on Reliability, 54*(2), 194–197.

Nelson, W. B. (2009). *Accelerated testing: Statistical models, test plans, and data analysis*. New York: John Wiley & Sons.

Nelson, W. B. (2015). An updated bibliography of accelerated test plans. In *2015 annual reliability and maintainability symposium (RAMS)* (pp. 1–6). Palm Harbor, FL, USA: IEEE.

Pessoa, A., Uchoa, E., de Aragão, M. P., & Rodrigues, R. (2010). Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. *Mathematical Programming Computation, 2*, 259–290.

Rostami, S., Creemers, S., & Leus, R. (2019). Precedence theorems and dynamic programming for the single-machine weighted tardiness problem. *European Journal of Operational Research, 272*(1), 43–49.

Schulz, A. S., & Uhan, N. A. (2011). Near-optimal solutions and large integrality gaps for almost all instances of single-machine precedence-constrained scheduling. *Mathematics of Operations Research, 36*(1), 14–23.

Silva, D., Rabelo, R., Campanha, M., Neto, P. S., Oliveira, P. A., & Britto, R. (2016). A hybrid approach for test case prioritization and selection. In *2016 IEEE congress on evolutionary computation (CEC)* (pp. 4508–4515). Vancouver, BC, Canada: IEEE.

Sourd, F. (2005). Earliness-tardiness scheduling with setup considerations. *Computers and Operations Research, 32*(7), 1849–1865.

Sourd, F. (2009). New exact algorithms for one-machine earliness-tardiness scheduling. *INFORMS Journal on Computing, 21*(1), 167–175.

Sourd, F., & Kedad-Sidhoum, S. (2003). The one-machine problem with earliness and tardiness penalties. *Journal of Scheduling, 6*(6), 533–549.

Sourd, F., & Kedad-Sidhoum, S. (2008). A faster branch-and-bound algorithm for the earliness-tardiness scheduling problem. *Journal of Scheduling, 11*(1), 49–58.

Sousa, J. P., & Wolsey, L. A. (1992). A time indexed formulation of non-preemptive single machine scheduling problems. *Mathematical Programming, 54*(1–3), 353–367.

Srikanth, H., Hettiarachchi, C., & Do, H. (2016). Requirements based test prioritization using risk factors: An industrial study. *Information and Software Technology, 69*, 71–83.

Tanaka, S., & Fujikuma, S. (2008). An efficient exact algorithm for general single-machine scheduling with machine idle time. In *4th IEEE conference on automation science and engineering* (pp. 371–376). Arlington, VA, USA: IEEE.

Tanaka, S., & Fujikuma, S. (2012). A dynamic-programming-based exact algorithm for general single-machine scheduling with machine idle time. *Journal of Scheduling, 15*(3), 347–361.

Tanaka, S., Fujikuma, S., & Araki, M. (2009). An exact algorithm for single-machine scheduling without machine idle time. *Journal of Scheduling, 12*(6), 575–593.

Tanaka, S., & Sato, S. (2013). An exact algorithm for the precedence-constrained single-machine scheduling problem. *European Journal of Operational Research, 229*(2), 345–352.

Tang, L., Xuan, H., & Liu, J. (2007). Hybrid backward and forward dynamic programming based Lagrangian relaxation for single machine scheduling. *Computers and Operations Research, 34*(9), 2625–2636.

Van den Akker, J. M., Hurkens, C. A. J., & Savelsbergh, M. W. P. (2000). Time-indexed formulations for machine scheduling problems: Column generation. *INFORMS Journal on Computing, 12*(2), 111–124.

Walcott, K. R., Soffa, M. Lou, Kapfhammer, G. M., & Roos, R. S. (2006). Time-aware test suite prioritization. In *Proceedings of the 2006 international symposium on software testing and analysis* (pp. 1–12). Portland Maine USA: ACM.

Zhang, H., Liu, S., & Li, Y. F. (2020). Optimizing the efficiency of accelerated reliability testing for the internet router motherboard. In *2020 annual reliability and maintainability symposium (RAMS)*. Palm Springs, CA, USA: IEEE.

Zhu, Y., & Elsayed, E. A. (2013). Optimal design of accelerated life testing plans under progressive censoring. *IIE Transactions, 45*(11), 1176–1187.