J Syst Sci Complex (2024) 37(6): 2406–2423

Modelling and Optimizing Motherboard Functional Testing in Laptop Manufacturing*

BAI Peng · KANG Yu · WANG Kangcheng · ZHAO Yunbo · DONG Shaojie

DOI: 10.1007/s11424-024-3502-8
Received: 20 November 2023 / Revised: 7 February 2024

Abstract Functional testing is key to fulfill quality control in laptop manufacturing and is of great economic value. However, due to the unavailability of practical data, mathematical model and systematic perspective, it has barely been touched from the academic community to date. For the first time, this work provides technical understanding of the key principles of functional testing, mathematically models the general framework, elucidates existing testing strategy under the proposed framework and model, and finally proposes a specified optimization strategy which outperforms existing strategies. This work lays the model foundation for the further optimization of functional testing, and can be regarded as a good example of how a systematic approach can solve practical industrial challenges.

Keywords Laptop manufacturing, modelling, motherboard functional testing, optimization.

BAI Peng
Department of Automation, University of Science and Technology of China, Hefei 230026*, China.*
Email: baipeng@lenovo.com.
KANG Yu
Department of Automation & Institute of Advanced Technology, University of Science and Technology of China, Hefei 230026*, China; Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei* 230088*, China.* Email: kangduyu@ustc.edu.cn.
WANG Kangcheng
Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088*, China.*
Email: kcwang@iai.ustc.edu.cn.
ZHAO Yunbo (Corresponding author)
Department of Automation & Institute of Advanced Technology, University of Science and Technology of China, Hefei 230026*, China; Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei* 230088*, China.* Email: ybzhao@ustc.edu.cn.
DONG Shaojie
Institute of Advanced Technology, University of Science and Technology of China, Hefei 230026*, China.*
Email: ahudong308@163.com.
*This research was supported by the Key Research and Development Program of Anhui under Grant No. 202104a05020064.
◇ *This paper was recommended for publication by Editor XIN Bin.*

1 Introduction

Functional testing is key to fulfill quality control in laptop manufacturing[1–5], whose general workflow is depicted in Figure 1. In the so-called end-to-end functional testing workflow[6], there exist two main functional testing stages: Printed circuit boards (PCBs) first go through the surface mount technology (SMT) production line to be produced as the motherboards, which are functionally tested and the defective ones are repaired; Finished laptops are also functionally tested, with the defective ones being repaired as well. Due to harsh quality control requirement, typically 100% finished laptops are functionally tested, meaning that we have to focus on the motherboard functional testing stage for any optimizing purpose[7–9].

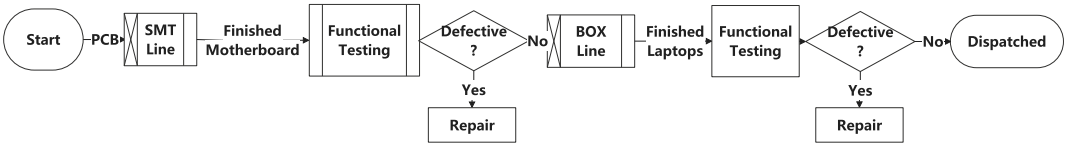

Figure 1 General framework of end-to-end functional testing in laptop manufacturing

For the motherboard functional testing stage, the testing line is installed following immediately after the SMT production line[10]. A typical motherboard to be tested involves about 30 different testing items, which have to be tested sequentially and cost several hundreds seconds if all functional items are tested. An automated testing machine can help prepare the testing and switch between different testing items. It is realized that it can be too time consuming for the production line to have all functions of all motherboards tested, but without sufficient functional testing of the motherboards it will cause more repair cost of defective finished laptops (see detailed discussions in Section 2). This creates a contradiction and is the main concern of the present work.

In a typical laptop manufacturing factory considered in this work (referred to as "Factory L" hereafter for privacy consideration), which is one of the largest all over the world, it has more than 30 such production lines, thus consequently accompanying more than 30 motherboard functional testing lines. Besides the space taken by the testing lines, the equipment of one automated testing line costs nearly 10^7 CNY, which includes 48 testing machines with each costing approximate 10^5 CNY, and necessary accessories costing approximate 4×10^6 CNY. Hence, in total the equipment only costs nearly 3×10^8 for motherboard functional testing in Factory L. This high cost justifies the economic value of optimizing functional testing in Factory L.

Though practically significant, such a seemingly urgent need remains barely touched from the academic community. The reasons are threefold.

- Data unavailability. It is understood that such a need is not public to the academic community at all, as the data are regarded as a business secret. They are open only to those trusted technical staff within Factory L, and possibly few people in academia who collaborate closely with these factories. This means, only a few people in academia may touch the problem in the first instance.

- Model unavailability. Due to the above reason, it is not surprising to find no well-formulated mathematical model for the problem at all. This creates huge challenges for the academic community to work on this problem, since anyone with interest has to first dig into the whole process and build his/her own model first, before any working solution to the problem.

- Systematic-perspective unavailability. In our more than two years of collaboration with Factory L, we find that the technical staff have not even realized that the testing lines can be optimized in a systematic way and hence have never thought of seeking for external help from the academic community. They intend to focus merely on the specific technical issues of particular equipment and operation, but not the whole process from a systematic perspective[11–18]. Their attitude is understandable, but unfortunately contributes to the separation between the industry and academia on this very problem. It is worth mentioning that our present work on the modeling of this very process, was not initiated by the invitation of Factory L, but by our own observation on the whole production line, within a larger smart manufacturing project collaborating with Factory L. This could be one key reason why our team can make an academic contribution to this industrial challenge but others do not.

By realizing the practical significance and academic absence as discussed above, in this work, we go deep into the production line of laptop manufacturing in Factory L, model the general framework of functional testing, and propose new testing strategies to optimize the testing line. Specifically, our contributions include:

- Technical understanding of the key principles in functional testing. In Section 2, three key principles, previously ambiguously comprehended by the technical staff, are explicitly stated and explained, which lay the foundation of the general motherboard functional testing framework.

- Mathematical modeling of the general framework of motherboard functional testing. In Section 3, the motherboard functional testing problem is for the first time mathematically modeled as an optimization problem, which lay the foundation of any further optimizing strategies for functional testing.

- Practical elucidating of Factory L's testing strategy under the general framework. In Section 4, we are able to elucidate why Factory L's existing strategy is designed in such a way, under our proposed general framework and model.

- A specified optimization strategy which outperforms existing strategies. In Section 5, under the general framework, we propose a specified optimization strategy for motherboard functional testing, which is shown to outperform existing strategies with high economic value.

2 Technical Understanding of the Key Principles in Functional Testing

In this section we reveal three key principles in functional testing, which are previously ambiguously comprehended by the technical staff only. These principles are such organized that they are firstly summarized and then explained in detail. Note that these principles help the construction of the general mathematical model of functional testing to be detailed in Section 3.

Principle 2.1 Having 100% finished laptops tested is an inevitable requirement to ensure quality.

Elucidation In order to meet the harsh quality standards, it is understood that all finished laptops have to be tested for all the functions. That is, the laptop defectives in the market, is either hardly found in these functional testing, or occur after being dispatched from the factory. Such a defective rate is referred to as the Inherent Defective Rate (IDR) of laptops, denoted by r_0^l, which should be relatively small, near 0.

The above 100% functional testing operation for finished laptops are inevitable. We may interpret this point by the following simple calculations.

Denote the defective rate of all finished laptops (before testing and repairing) by r_D^l, and the ratio of the laptops to be repaired after testing by r_R^l. It is then held that

$$r_R^l = r_D^l - r_0^l. \tag{1}$$

It is reasonable to assume that most defectives can be found by functional testing, meaning that $r_D^l \approx r_R^l$. At the same time, r_D^l should be a non-neglectable value, otherwise there will be no need for functional testing in the first instance. The above argument means

$$r_D^l \approx r_R^l \gg r_0^l. \tag{2}$$

Now suppose we stop doing functional testing for 10% finished laptops, and then the defective rate of laptops in the market will become

$$\frac{0.1r_D^l + 0.9r_0^l}{r_0^l}, \tag{3}$$

times of the inherent defective rate r_0^l. Then, for a typical case of $r_D^l/r_0^l > 100$, a decrease of 10% functional testing cost causes more than 10 times defective rate increase of laptops in the market. This is simply unacceptable.

Principle 2.2 Decreasing r_D^l is a fundamental requirement to decrease repair cost.

Elucidation We now understand that 100% finished laptops have to be tested effectively, which ensures the laptop quality in the market, but also means that the testing cost for each laptop, denoted by c_T^l, is inevitable.

On the average, the overall testing and repair cost for each laptop, can be written as follows:

$$c_A^l = c_T^l + r_R^l c_R^l \approx c_T^l + r_D^l c_R^l, \tag{4}$$

where c_R^l is the average repair cost of laptops found defective in the testing operation. In practice, both r_D^l and c_R^l are relatively high, causing remarkable repair cost.

We notice that the average testing cost c_T^l and the average repair cost c_R^l should be some relatively fixed value and can not be manipulated. To decrease c_A^l, the only approach is to decrease r_R^l, and consequently to decrease the defective rate of finished laptops r_D^l, since $r_R^l \approx r_D^l$.

Principle 2.3 Motherboard functional testing is the key approach to decreasing r_D^l.

Elucidation The defective rate of finished laptops r_D^l is affected by the whole manufacturing production line. However, it is a fact that the repair cost can be much less if a defective is found right after its occurrence. Hence, having 100% testing only for finished laptops means the lowest testing cost but the highest repair cost; On the other hand, testing all laptops right after each manufacturing operation means the highest testing cost but the lowest repair cost.

The above analysis implies the existence of an optimal solution other than the above two extreme cases, which can balance between the testing and repair costs. The optimal solution should meet the following principles.

- Operations with sufficiently high yield should not have dedicated testing, since the testing cost can be much larger than the repair cost.
- Operations with sufficiently low yield should have dedicated 100% testing right after this operation, since this testing has to be part of the production line, and hence is not necessary to be partially tested. This also means that there is no further testing optimization possibility for such a low-yield operation.
- Operations with normal yield should be tested flexibly, balancing between the testing and repair costs. The above analysis already reveals that dedicated testing right after an operation should be 100%, it thus means that flexible testing should not be right after the operation, and consequently, these flexible testing should be arranged in some break point of the production line, in order not to interrupt normal production.

The above principles lead immediately the conclusion that finished motherboards should be functionally tested for the defectives caused by operations with normal yield, since firstly, the SMT production line and BOX production line are independent from each other, and secondly, the defectives of motherboards contribute the majority defectives to finished laptops.

3 Mathematical Modelling of the General Framework of Motherboard Functional Testing

The general framework of motherboard functional testing is depicted in Figure 2. In this framework, N_T different testing items are sequentially tested. For the ith testing item of a motherboard, denoted by $\mathcal{T}_i$, it is first checked whether testing is needed. If no then check the next testing item; if yes then do the testing, where the failed testing items are recorded. Only those passing all testing items or those successfully repaired go to the BOX production line.

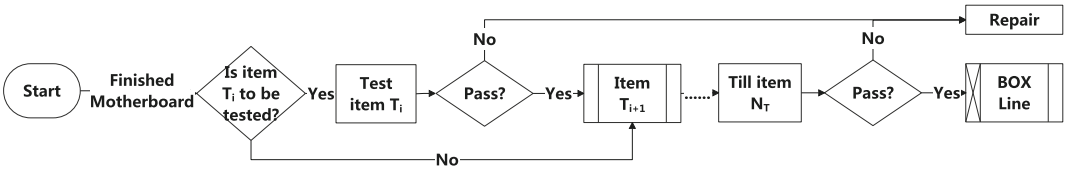

Figure 2 General framework of motherboard functional testing

In what follows, we first formulate the fundamental optimization problem of motherboard functional testing, and then detail each part of the optimization problem to form the general mathematical model. We also discuss the the basic solution to the optimization problem.

Notice first that both the testing cost c_T^m and the repair cost r_R^m are dependent on specific testing strategies: A well-designed testing strategy can find more defective items at lower testing cost, while an ill-designed one may spend more testing cost to find only few defective items. For clarity the concept of functional testing strategy, denoted by $\mathcal{S}$, is formally defined as in Definition 3.1.

Definition 3.1 (The motherboard functional testing strategy $\mathcal{S}$) A motherboard functional testing strategy is to determine, either for a specific motherboard, whether each testing item $\mathcal{T}_i$ is tested or not; or for a volume of motherboards during a specific time interval, what percentage of motherboards is tested for each testing item. Clearly, the former is a more general interpretation but the latter may be more practical. Based on the above understanding, a motherboard functional testing strategy, denoted by $\mathcal{S}$, can be written as follows:

$$\mathcal{S} : (p_1, p_2, \cdots, p_{N_T}), \quad \mathcal{S} \in \mathbb{S}, \tag{5}$$

where p_i, interpreted as the testing probability or percentage, is to be determined, and $\mathbb{S}$ is the set of all possible testing strategies.

Based on the testing strategy concept in Definition 3.1, we may then formally model the fundamental motherboard functional testing problem as follows, which is "fundamental" since no detailed testing strategy is considered.

Proposition 3.2 (The fundamental optimization problem of motherboard functional testing) *The fundamental optimization problem of motherboard functional testing can be modeled as the minimization of the effective cost $c_E(\mathcal{S})$, i.e.,*

$$\arg\min_{\mathcal{S}\in\mathbb{S}} c_E(\mathcal{S}), \tag{6}$$

where

$$c_E(\mathcal{S}) = c_T^m(\mathcal{S}) - \Delta c_R r_R^m(\mathcal{S}) \tag{7}$$

and Δc_R represents the cost difference of a faulty motherboard being repaired as a finished laptop or as still a motherboard and r_R^m is the repair rate of motherboards.

In practice, it is often the case that the functional testing is time constrained, i.e., the testing time for each motherboard is no longer than $\overline{t}_T$. In this case the optimization problem above turns to

$$\begin{aligned}&\arg\min_{\mathcal{S}\in\mathbb{S}} c_E(\mathcal{S})\\ &\text{s.t.}\quad \sum_{i=1}^{N_T} p_i \overline{t}_{Ti} \le \overline{t}_T,\end{aligned} \tag{8}$$

where $\overline{t}_{Ti}$ is the average testing time for testing item $\mathcal{T}_i$.

Proof The key to the fundamental optimization problem is to formulate the overall cost, denoted by $c_O(\mathcal{S})$ implying that it is testing strategy $\mathcal{S}$ dependent.

$c_O(\mathcal{S})$ may be written as the sum of the following items, the testing cost $c_T^m(\mathcal{S})$, the average repair cost of motherboards being faulty and repaired as still motherboards $c_R^m r_R^m(\mathcal{S})$, and the average repair cost of motherboards being faulty and repaired as finished laptops $c_R^l(r_D - r_R^m(\mathcal{S}))$, i.e.,

$$c_O(\mathcal{S}) = c_T^m(\mathcal{S}) + c_R^m r_R^m(\mathcal{S}) + c_R^l(r_D - r_R^m(\mathcal{S})),$$

where c_R^m and c_R^l are the average repair costs of motherboards being faulty and repaired as still a motherboard or as a finished laptop, respectively, and r_D is the inherent defective rate without any testing.

Hence, the fundamental optimization problem is to minimize $c_O(\mathcal{S})$, or

$$\begin{aligned}\arg\min_{\mathcal{S}\in\mathbb{S}} c_O(\mathcal{S}) &= \arg\min_{\mathcal{S}\in\mathbb{S}}\{c_T^m(\mathcal{S}) + c_R^m r_R^m(\mathcal{S}) + c_R^l(r_D - r_R^m(\mathcal{S}))\}\\ &= \arg\min_{\mathcal{S}\in\mathbb{S}}\{c_T^m(\mathcal{S}) - \Delta c_R r_R^m(\mathcal{S}) + c_R^l r_D\}\\ &= \arg\min_{\mathcal{S}\in\mathbb{S}}\{c_T^m(\mathcal{S}) - \Delta c_R r_R^m(\mathcal{S})\}\\ &:= \arg\min_{\mathcal{S}\in\mathbb{S}} c_E(\mathcal{S}),\end{aligned}$$

where $\Delta c_R := c_R^l - c_R^m > 0$, and we may drop the item $c_R^l r_D$ since both c_R^l and r_D may be regarded as some constants and hence do not affect the minimization problem.

Furthermore, the constraint in (8) may be derived from the definition of a testing strategy in Definition 3.1 straightforwardly. This thus completes the proof. ∎

Remark 3.3 Proposition 3.2 implies that the fundamental optimization problem of motherboard functional testing is to design appropriate testing strategy $\mathcal{S}$ (subject to the time constraint $\overline{t}_{Ti}$), such that the overall testing cost $c_E(\mathcal{S})$ is minimized. This modeling framework turns the motherboard functional testing problem to an optimization problem, to find the optimal testing strategy $\mathcal{S}$ which has been appropriately formulated in Definition 3.1.

Based on Proposition 3.2, we can then propose the formal mathematical model of motherboard functional testing by specifying explicitly the testing strategy $\mathcal{S}$.

Theorem 3.4 *The motherboard functional testing problem can be mathematically modeled as follows:*

$$\arg\min_{\mathcal{S}\in\mathbb{S}} c(\mathcal{S}), \tag{9}$$

where

$$c(\mathcal{S}) = \sum_{i=1}^{N_T} p_i c_{Ti}^m + \Delta c_R \prod_{i=1}^{N_T} (1 - p_i r_{Di}), \tag{10}$$

with c_{Ti} being the testing cost of each testing item $\mathcal{T}_i$, and r_{Di} being the actual defective rate of testing item $\mathcal{T}_i$.

If we consider the time constraint as in (8)*, the above optimization model* (16) *then turns to*

$$\begin{aligned} &\arg\min_{\mathcal{S}\in\mathbb{S}} c(\mathcal{S}) \\ &\text{s.t.} \quad \sum_{i=1}^{N_T} p_i \overline{t}_{Ti} \le \overline{t}_T. \end{aligned} \tag{11}$$

Proof Clearly, the key to the theorem is to formulate the effective cost $c_E(\mathcal{S})$ in Proposition 3.2. We do that by the following three steps.

1) The testing cost $c_T^m(\mathcal{S})$ derived from the testing strategy $\mathcal{S}$. Using testing strategy $\mathcal{S}$, the overall testing cost can be readily obtained as

$$c_T^m(\mathcal{S}) = \sum_{i=1}^{N_T} p_i c_{Ti}^m. \tag{12}$$

2) The repair rate $r_R^m(\mathcal{S})$ derived from the testing strategy $\mathcal{S}$. In order to establish the relationship between the repair rate $r_R^m(\mathcal{S})$ and the testing strategy $\mathcal{S}$, we have to make two assumptions. Firstly, a motherboard will be repaired if it fails any one of the testing items (of course, there can be more than one testing items that it fails). Secondly, no correlations of any kind can be found in these testing items, meaning that one may not deduce anything on one testing item from any other testing item(s). This second assumption may not be always held, but should be generally true, since otherwise the correlated testing items have already been ignored in practice.

Based on the above assumptions, the probability of a motherboard being tested to be non-defective, i.e., $1 - r_R^m(\mathcal{S})$, can be written as the product of the probability of each testing item being non-defective, $1 - r_{Ri}^m(\mathcal{S})$, that is,

$$1 - r_R^m(\mathcal{S}) = \prod_{i=1}^{N_T} (1 - r_{Ri}^m(\mathcal{S})). \tag{13}$$

For any testing item $\mathcal{T}_i$ of any specific motherboard, it is reasonable to assume that being tested can always find whether it is defective, but it remains unknown without testing. Hence,

$$r_{Ri}^m(\mathcal{S}) = p_i r_{Di} \tag{14}$$

and therefore,

$$r_R^m(\mathcal{S}) = 1 - \prod_{i=1}^{N_T} (1 - p_i r_{Di}), \tag{15}$$

which builds the relationship between $r_R^m(\mathcal{S})$ and $\mathcal{S}$.

3) The model for motherboard functional testing. From (8), (12) and (15), it can be derived that

$$\begin{aligned}\arg\min_{\mathcal{S}\in\mathbb{S}} c_E(\mathcal{S}) &= \arg\min_{\mathcal{S}\in\mathbb{S}}\{c_T^m(\mathcal{S}) - \Delta c_R r_R^m(\mathcal{S})\}\\ &= \arg\min_{\mathcal{S}\in\mathbb{S}}\left\{\sum_{i=1}^{N_T} p_i c_{Ti}^m - \Delta c_R\left(1-\prod_{i=1}^{N_T}(1-p_i r_{Di})\right)\right\}\\ &= \arg\min_{\mathcal{S}\in\mathbb{S}}\left\{\sum_{i=1}^{N_T} p_i c_{Ti}^m + \Delta c_R\prod_{i=1}^{N_T}(1-p_i r_{Di})\right\}\\ &= \arg\min_{\mathcal{S}\in\mathbb{S}} c(\mathcal{S}). \end{aligned} \tag{16}$$

This completes the proof. ∎

Proposition 3.5 *An optimal testing strategy should consider the following design principle: The probability of testing item $\mathcal{T}_i$ being tested p_i should be inversely proportional to its actual defective rate r_{Di}, and is adjusted by its testing cost c_{Ti}^m.*

Proof To obtain the solution to (10), we may let

$$\frac{\partial c(\mathcal{S})}{\partial p_i} = 0,$$

which yields

$$\frac{c_{Ti}^m}{\Delta c_R r_{Di}} = \prod_{j\neq i}^{N_T}(1-p_j r_{Dj})$$

or

$$\frac{c_{Ti}^m}{\Delta c_R r_{Di}}(1-p_i r_{Di}) = \prod_{j=1}^{N_T}(1-p_j r_{Dj}) = 1 - r_R^m(\mathcal{S})$$

and further,

$$p_i = \frac{1}{r_{Di}} - \frac{\Delta c_R}{c_{Ti}^m}(1-r_R^m(\mathcal{S})). \tag{17}$$

Core to the testing strategy $\mathcal{S}$ is to determine the testing probability p_i of each testing item $\mathcal{T}_i$. (17) means that for the optimal testing strategy, the probability of testing item $\mathcal{T}_i$ being tested p_i should be inversely proportional to its actual defective rate r_{Di}, and is adjusted by its testing cost c_{Ti}^m, which is consistent with our intuitions. ∎

4 Elucidating Factory L's Testing Strategy Under the General Framework

The framework of the motherboard functional testing strategy of Factory L is depicted in Figure 3. In this strategy, firstly, certain testing items are must-test due to their importance, i.e.,

being 100% tested regardless of other conditions; secondly, fully testing for a certain percentage of motherboards (typically 20%) are carried out, while for the left motherboards (typically 80%) the testing items are optional. An optional testing item is not tested, if 1) it is not a must-test item; 2) the estimated defective rate of this testing item is sufficiently low; and 3) the number of motherboards that can be used to estimate its defective rate is more than some threshold (typically 20000).

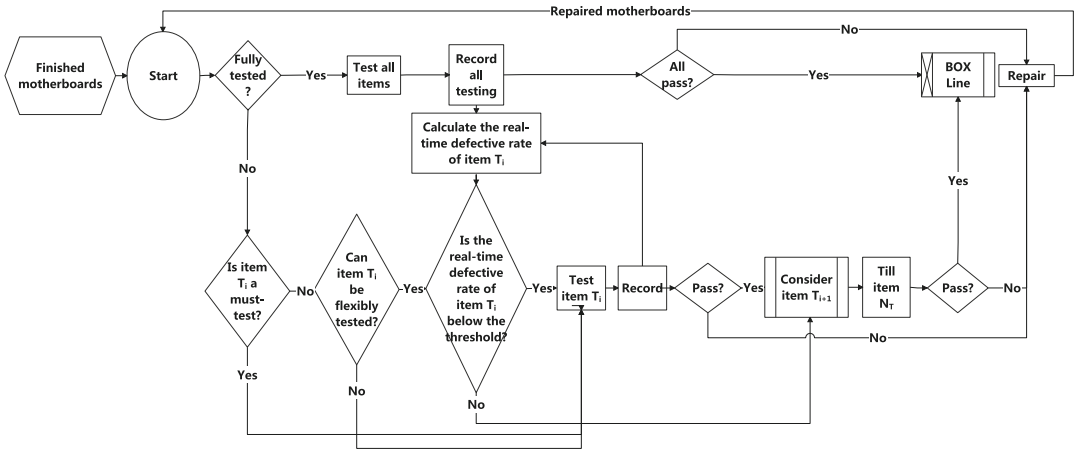

Figure 3 General framework of Factory L's motherboard functional testing strategy

The following theorem makes it clear that the testing strategy in Factory L can be elucidated under our proposed framework and model in Section 3.

Theorem 4.1 *Factory L's testing strategy can be elucidated under the general framework in Theorem* 3.4*, i.e., its strategy, denoted by* $\mathcal{S}_0$*, is designed by optimizing the following average effective cost,*

$$c(\mathcal{S}_0) = \sum_{\mathcal{T}_i \in \mathbb{T}_0} \mu c_{Ti}^m + \sum_{\mathcal{T}_i \in \mathbb{T}_1} c_{Ti}^m + \Delta c_R \prod_{\mathcal{T}_i \in \mathbb{T}_0} (1 - \mu r_{Di}) \prod_{\mathcal{T}_i \in \mathbb{T}_1} (1 - r_{Di}), \tag{18}$$

where μ *is the percentage of the motherboards being fully tested, and* σ_{Ti} *is the indicator that shows whether the optional testing item* $\mathcal{T}_i$ *is tested, that is,* $\sigma_{Ti} = 1$ *if* $\mathcal{T}_i$ *is tested, and* $\sigma_{Ti} = 0$ *if* $\mathcal{T}_i$ *is not tested, and*

$$\mathbb{T}_0 = \{\mathcal{T}_i : \sigma_{Ti} = 0\}, \tag{19a}$$

$$\mathbb{T}_1 = \{\mathcal{T}_i : \sigma_{Ti} = 1\}. \tag{19b}$$

Proof We may formulate the testing strategy of Factory L, denoted by $\mathcal{S}_0$, as a special case of Definition 3.1, i.e.,

$$\mathcal{S}_0 : (p_i = \mu + \sigma_{Ti}(1 - \mu), i = 1, 2, \cdots, N_T). \tag{20}$$

From (16) and (20), the effective average cost of testing strategy $\mathcal{S}_0$ is

$$c(\mathcal{S}_0) = \sum_{i=1}^{N_T} p_i c_{Ti}^m + \Delta c_R \prod_{i=1}^{N_T}(1 - p_i r_{Di})$$
$$= \sum_{\mathcal{T}_i \in \mathbb{T}_0} \mu c_{Ti}^m + \sum_{\mathcal{T}_i \in \mathbb{T}_1} c_{Ti}^m + \Delta c_R \prod_{\mathcal{T}_i \in \mathbb{T}_0} (1 - \mu r_{Di}) \prod_{\mathcal{T}_i \in \mathbb{T}_1} (1 - r_{Di}), \tag{21}$$

where $\mathbb{T}_0$ and $\mathbb{T}_1$ are defined as in (19).

This completes the proof. ∎

Remark 4.2 By definition σ_{Ti} is determined by whether the estimated defective rate r_{Ti} of testing item $\mathcal{T}_i$ exceeds certain threshold r_{Ti}^0, i.e.,

$$\sigma_{Ti} = \begin{cases} 1, & r_{Ti} \geq r_{Ti}^0, \\ 0, & r_{Ti} < r_{Ti}^0, \end{cases} \tag{22}$$

r_{Ti} can be estimated as follows:

$$r_{Ti} = \frac{N_{Ti}(\mu)}{\mu M}, \tag{23}$$

where $N_{Ti}(\mu)$ is the number of defective testing item $\mathcal{T}_i$ being found among M motherboards with the fully testing percentage being μ. It is understood that r_{Ti} approaches r_{Di} with the increase of μ (and sufficiently large M).

5 A Specified Optimization Strategy Validating the General Framework and Model for Functional Testing

In this section, we specify the general optimization framework for functional testing and verify its effectiveness numerically.

5.1 A Specified Optimization Strategy

Consider the optimization strategy for an already founded testing line in Factory L. Suppose, under strategy $\mathcal{S}$, there are in total of $q(\mathcal{S})$ testing machines, each costing u_S, and the current cost for all other accessories in the factory is u_A. Let the current strategy of the factory be $\mathcal{S}_0(\mu_0)$. Then, the average cost for each testing machine (with consideration of its accessories) is

$$\overline{u}_{SA} = u_S + \frac{u_A}{q(\mathcal{S}_0(\mu_0))}. \tag{24}$$

Let the average usage life for the testing machines be l_{SA}, then for testing item $\mathcal{T}_i$, its cost in (10) can be specified as follows:

$$c_{T_i}^m = \frac{\overline{u}_{SA}}{l_{SA}} \overline{t}_{Ti}. \tag{25}$$

The average testing time per motherboard under strategy S is

$$\overline{t}_T(\mathcal{S}) = \sum_{i=1}^{N_T} p_i \overline{t}_{T_i}. \tag{26}$$

Then the average testing time per motherboard using $q(\mathcal{S})$ testing machines in parallel is

$$\overline{t}_T^a(\mathcal{S}) = \frac{\overline{t}_T(\mathcal{S})}{q(\mathcal{S})}. \tag{27}$$

Suppose that when using the current strategy and the current testing machines to test in parallel, the functional testing operation can just keep pace with the motherboard manufacturing operation. Then, the average manufacturing time for each motherboard can be specified as follows:

$$\overline{t}_{manuf} = \overline{t}_T^a(\mathcal{S}_0(\mu_0)). \tag{28}$$

In order to prevent motherboard accumulation before the functional testing operation, the average testing time per motherboard under strategy S should not exceed the average manufacturing time per motherboard, i.e.,

$$\overline{t}_T^a(\mathcal{S}) \leq \overline{t}_{manuf}. \tag{29}$$

Let the running time for the testing machines be w_L^m per week. Typically, the testing machines run automatically without interruption. Let the working time and wage for the workers be w_R per week and v_R, respectively. The average repair costs (time) for the defective motherboard and laptop are t_R^m and t_R^l, respectively. Then, the two costs can be specified as follows:

$$c_R^m = \frac{v_R}{w_R} t_R^m, \tag{30}$$

$$c_R^l = \frac{v_R}{w_R} t_R^l. \tag{31}$$

Moreover, other than the constraint in (29), the rework operation for defective motherboards and laptops should be not slower than the testing operation. Let the workers for repairing defective motherboards and laptops be $n_R^m(\mathcal{S})$ and $n_R^l(\mathcal{S})$, respectively. Then, the constraints on the rework operation can be derived as follows.

On one hand, the average testing time for each motherboard is

$$\overline{t}_R^m(\mathcal{S}) = \frac{t_R^m}{n_R^m(\mathcal{S})}. \tag{32}$$

The number of defective motherboards that can be repaired per week is

$$M_R^m(\mathcal{S}) = \frac{w_R}{\overline{t}_R^m(\mathcal{S})}. \tag{33}$$

On the other hand, the total number of motherboards detected by the testing machines per week is

$$M_L^m(\mathcal{S}) = \frac{w_L^m}{\overline{t}_T^a(\mathcal{S})}. \tag{34}$$

The number of defective motherboards detected by the testing machines per week can be determined using the motherboard rework rate $r_R^m(\mathcal{S})$ as follows:

$$M_{LD}^m(\mathcal{S}) = r_R^m(\mathcal{S}) M_L^m(\mathcal{S}). \tag{35}$$

To prevent motherboard accumulation before the rework operation, the number of motherboards requiring rework should not be less than the number of defective motherboards detected by the testing machines, i.e.,

$$M_R^m(\mathcal{S}) \geq M_{LD}^m(\mathcal{S}). \tag{36}$$

Similarly, the average time for repairing each defective laptop is

$$\overline{t}_R^l(\mathcal{S}) = \frac{t_R^l}{n_R^l(\mathcal{S})}. \tag{37}$$

The number of defective laptops that can be repaired per week is

$$M_R^l(\mathcal{S}) = \frac{w_R}{\overline{t}_R^l(\mathcal{S})}. \tag{38}$$

Among the $M_L^m(\mathcal{S})$ motherboards tested by the testing machines per week, a proportion of $r_R^m(\mathcal{S})$ defective motherboards are reworked after testing, and a proportion of approximately $r_R^l(\mathcal{S})$ defective motherboards are assembled as laptops and then reworked after laptop testing. Therefore, the number of defective motherboards assembled in laptops is approximately,

$$M_{LD}^l(\mathcal{S}) = r_R^l(\mathcal{S}) M_L^m(\mathcal{S}) \approx (r_D - r_R^m(\mathcal{S})) M_L^m(\mathcal{S}), \tag{39}$$

where r_D denotes the defect rate of the laptops without the motherboard functional testing operation.

To prevent the accumulation of tested laptops before the rework operation, the number of defective laptops requiring rework should be equal to or greater than the number of defective laptops assembled after motherboard testing, i.e.,

$$M_R^l(\mathcal{S}) \geq M_{LD}^l(\mathcal{S}). \tag{40}$$

Furthermore, an additional constraint on the number of rework workers should be considered to ensure the automation level of the functional testing operation.

Let the total number of rework workers using strategy S be

$$n_R^{ml}(\mathcal{S}) = n_R^m(\mathcal{S}) + n_R^l(\mathcal{S}). \tag{41}$$

To maintain the level of automation in the testing operation, the total number of rework workers using the new strategy should not exceed the number using the current strategy, i.e.,

$$n_R^{ml}(\mathcal{S}) \leq n_R^{ml}(\mathcal{S}_0(\mu_0)). \tag{42}$$

From (16), (29), (36), (40), (42), the optimization problem turns to

$$\min_{\mathcal{S}\in\mathbb{S}} c(\mathcal{S}) \tag{43a}$$

$$\text{s.t.} \quad \overline{t}_T^a(\mathcal{S}) \leq \overline{t}_{manuf}, \tag{43b}$$

$$M_{LD}^m(\mathcal{S}) \leq M_R^m(\mathcal{S}), \tag{43c}$$

$$M_{LD}^l(\mathcal{S}) \leq M_R^l(\mathcal{S}), \tag{43d}$$

$$n_R^{ml}(\mathcal{S}) \leq n_R^{ml}(\mathcal{S}_0(\mu_0)). \tag{43e}$$

To further validate the effectiveness of the established general framework and the model for functional testing, an additional optimization problem is formulated as follows to provide a specified strategy for Factory L. In this problem, only the fully testing percentage μ of optional testing items is optimized within the factory's framework, while the must-test items remain unchanged. From (21), (29), (36), (40), (42), the strategy can be specified as follows:

$$\min_{\mu} c(\mathcal{S}_0(\mu)) \tag{44a}$$

$$\text{s.t.} \quad \overline{t}_T^a(\mathcal{S}_0(\mu)) \leq \overline{t}_{manuf}, \tag{44b}$$

$$M_{LD}^m(\mathcal{S}_0(\mu)) \leq M_R^m(\mathcal{S}_0(\mu)), \tag{44c}$$

$$M_{LD}^l(\mathcal{S}_0(\mu)) \leq M_R^l(\mathcal{S}_0(\mu)), \tag{44d}$$

$$n_R^{ml}(\mathcal{S}_0(\mu)) \leq n_R^{ml}(\mathcal{S}_0(\mu_0)). \tag{44e}$$

With specified parameters, the above optimization problems can be solved in the following process. First, determine the ranges of $q(\mathcal{S})$, $n_R^m(\mathcal{S})$, and $n_R^l(\mathcal{S})$ based on practical experience. Then, by trying different values within the ranges, one may determine the values of $q(\mathcal{S})$, $n_R^m(\mathcal{S})$, and $n_R^l(\mathcal{S})$, corresponding to the optimal objective value.

5.2 Validating the Specified Strategy

Due to business privacy requirement, we are not allowed to show the real data here. We confirm that our mathematical model as well as its solution have been implemented in the real production line in Factory L for several months and have proven its effectiveness so far, see Figure 4 for the implementation scene. Here, to show the effectiveness of our approach, we adopt a modified data set from the real one, which should be sufficient for the validation purpose.

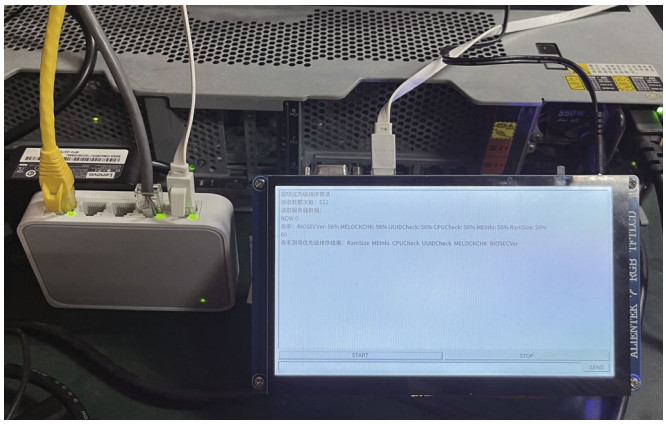

Figure 4 Our mathematical model as well as its solution has been implemented in the production line in Factory L

Let the number of testing items be $N_T = 30$, and $M = 10^5$ testing data. The mean testing time for each testing item, denoted as $\overline{t}_{T_i}$, is a uniformly distributed random number between $[0.2, 10]$, while the standard deviation is set as $\overline{t}_{T_i}/100$. Random numbers following

the Gaussian distribution are generated using the mean and standard deviation to represent the testing time of each motherboard, which yields an average testing time for a motherboard of 186.7708 seconds, close to the real value. The native defective rate of each testing item is a uniformly distributed random number between $[0.998, 1]$.

Other parameters are determined as follows. The threshold of defective rate in Factory L is $r_{Ti}^0 = 0.0005$. Out of all motherboards, $\mu_0 = 20\%$ are fully tested. It has $q(\mathcal{S}_0(\mu_0)) = 24$ testing machines, with each costing $u_S = 10^5$ CNY, and $u_A^a = 2 \times 10^6$ CNY. The average usage life $l_{SA} = 5$. The running time $w_L^m = 168$ hours per week. The number of workers $n_R^m(\mathcal{S}_0(\mu_0)) = 10$ and $n_R^l(\mathcal{S}_0(\mu_0)) = 2$. The wages $v_R = 1200$. The work time $w_R = 40$ hours per week. The average repair time $t_R^m = 900$ seconds, and $t_R^l = 910$ seconds.

The simulation for this study was conducted on a Windows 10 PC equipped with an Intel Core 2.5 GHz CPU and 32 GB of RAM. The simulation was performed using Matlab R2023a software. The `fmincon` function is utilized to solve the optimization problem in (43a–43e) and (44a–44e), with the optimization interval being $[0, 1]$, and the initial value being random within this interval.

The ranges of $q(\mathcal{S})$, $n_R^m(\mathcal{S})$, and $n_R^l(\mathcal{S})$ are set at 12 to 24 sets, 12 to 18 persons, and 0 to 6 persons, respectively. Denote the strategy designed within the framework of Factory L as $\mathcal{S}_0(\mu^*)$, and the optimal one be $\mathcal{S}^*$. The time required to derive $\mathcal{S}_0(\mu^*)$ and $\mathcal{S}^*$, along with their corresponding values of $q(\mathcal{S})$, $n_R^m(\mathcal{S})$, and $n_R^l(\mathcal{S})$ are 74.4 seconds and 55.9 seconds, respectively. The results are shown in Table 1, where the percentage of fully tested motherboards of $\mathcal{S}_0(\mu^*)$ is $\mu^* = 4.4448 \times 10^{-6}$.

From Table 1, it is seen that strategy $\mathcal{S}_0(\mu^*)$ requires a low percentage of testing for testing items with high yield rate; $\mathcal{S}^*$ effectively balances between the testing time and defective rate of all testing items, with the focus on those items that can either be tested fast or are of high defective rate.

Table 1 Optimal solution vs. Existing solution

Testing Item	$\overline{t}_{Ti}$	r_{Ti}	$\mathcal{S}_0(\mu_0)$	$\mathcal{S}_0(\mu^*)$	$\mathcal{S}^*$
1	8.4132	0.0004	0.2000	0.0122	0.0034
2	5.6211	0.0001	0.2000	0.0122	0.0040
3	4.9850	0.0015	1.0000	1.0000	0.9951
4	6.6460	0.0001	0.2000	0.0122	0.0034
5	5.7984	0.0006	1.0000	1.0000	0.0082
6	3.3483	0.0019	1.0000	1.0000	0.9974
7	9.3973	0.0014	1.0000	1.0000	0.0213
8	5.2658	0.0009	1.0000	1.0000	0.5744
9	6.0082	0.0002	0.2000	0.0122	0.0040
10	9.0318	0.0001	0.2000	0.0122	0.0023
11	8.9430	0.0018	1.0000	1.0000	0.9866
12	5.6829	0.0001	0.2000	0.0122	0.0039
13	7.1624	0.0001	0.2000	0.0122	0.0031

Table 1 (Continued) Optimal solution vs. Existing solution

Testing Item	$\overline{t}_{Ti}$	r_{Ti}	$\mathcal{S}_0(\mu_0)$	$\mathcal{S}_0(\mu^*)$	$\mathcal{S}^*$
14	1.4172	0.0010	1.0000	1.0000	0.9956
15	2.1517	0.0003	0.2000	0.0122	0.0841
16	8.7972	0.0018	1.0000	1.0000	0.9880
17	6.4684	0.0009	1.0000	1.0000	0.0133
18	8.8059	0.0002	0.2000	0.0122	0.0026
19	6.4338	0.0004	0.2000	0.0122	0.0051
20	9.8463	0.0001	0.2000	0.0122	0.0021
21	2.8260	0.0007	1.0000	1.0000	0.9836
22	2.5504	0.0016	1.0000	1.0000	0.9970
23	8.9067	0.0002	0.2000	0.0122	0.0027
24	4.7249	0.0001	0.2000	0.0122	0.0049
25	3.9446	0.0007	1.0000	1.0000	0.7639
26	8.0647	0.0006	1.0000	1.0000	0.0043
27	3.0057	0.0006	1.0000	1.0000	0.9513
28	3.9100	0.0012	1.0000	1.0000	0.9940
29	9.9752	0.0007	1.0000	1.0000	0.0034
30	8.6386	0.0016	1.0000	1.0000	0.9732

Let Δ_0^* and Δ^* be the relative change in the two strategies $\mathcal{S}_0(\mu^*)$ and $\mathcal{S}^*$ compared to $\mathcal{S}_0(\mu_0)$ (unit: Percentage). The essential economic indexes for each strategy are calculated and shown in Table 2.

Table 2 Other optimization indexes

Index	Unit	$\mathcal{S}_0(\mu_0)$	$\mathcal{S}_0(\mu^*)$	$\mathcal{S}^*$	$\Delta_0^*(\%)$	$\Delta^*(\%)$
$c(\mathcal{S})$	CNY/board	0.2157	0.1962	0.1456	−9.0355	−32.4790
$t^m(\mathcal{S})$	seconds/board	115.2229	98.4304	54.6085	−14.5739	−52.6062
$q(\mathcal{S})$	set	24	21	12	−12.5000	−50.0000
$n_R^{ml}(\mathcal{S})$	person	18	18	18	0.0000	0.0000

From Table 2, it is shown that strategies $\mathcal{S}_0(\mu^*)$ and $\mathcal{S}^*$ optimize at the indexes of average effective cost, average testing time and number of required testing machines without increasing the amount of repair workers. In particular, strategy $\mathcal{S}^*$ decreases the testing time and cost by as more as 50% and 30%, respectively. Besides, only half of the testing machines are needed for testing. This means, by applying the designed strategy $\mathcal{S}^*$ to the production line, the testing costs can be significantly reduced. Moreover, the expenses related to the space occupied by the testing machines and the accessories of testing machines are expected to be further reduced, which is of great economic value.

6 Conclusions

Functional testing in laptop manufacturing is systematically addressed from the academic community for the first time. This includes the technical understanding of the key principles

in functional testing, the mathematical modeling of the general framework, the practical elucidating of the existing strategy of a leading factory, and finally a specified optimization strategy outperforming existing ones.

To actually solve this problem, one may realize that the key challenge is not the optimization problem itself, but the accurate estimate of the defective rate under scarce labelled data. The latter is one core challenge for the-state-of-the-art deep learning as well, receiving much attentions from the academic community but having not reached any universally accepted solutions to date. We are working on solutions to this challenge, not from pure deep learning perspective, but relying on our deep understanding on the industrial process and the systematic modeling and design.

We believe that this work is a good example showing that a systematic perspective and modeling can solve complex industrial challenge, even beyond the imagination of the industry themselves. We hope that this work can inspire people of interest in academia to contribute to the industry, provided a real deep collaboration relationship can be built between both sides.

Conflict of Interest

ZHAO Yunbo is a youth editorial board member for Journal of Systems Science & Complexity and was not involved in the editorial review or the decision to publish this article. All authors declare that there are no competing interests.

References

[1] Ribeiro J, Lima R, Eckhardt T, et al., Robotic process automation and artificial intelligence in industry 4.0 — A literature review, *Procedia Computer Science*, 2021, **181**: 51–58.

[2] Ait-El-Cadi A, Gharbi A, Dhouib K, et al., Integrated production, maintenance and quality control policy for unreliable manufacturing systems under dynamic inspection, *International Journal of Production Economics*, 2021, **236**: 108140.

[3] Jwo J S, Lin C S, and Lee C H, Smart technology-driven aspects for human-in-the-loop smart manufacturing, *The International Journal of Advanced Manufacturing Technology*, 2021, **114**(5): 1741–1752.

[4] Serban M, Vagapov Y, Chen Z, et al., Universal platform for PCB functional testing, 2014 *Int. Conf. Actual Probl. Electron Devices Eng. APEDE*, IEEE, Saratov, Russia, 2014, 402–409.

[5] Houdek C, Inspection and testing methods for PCBs: An overview, technical report 401, Caltronics Design & Assembly, Inc., 1–7.

[6] Li D, Wang L, and Huang Q, A case study of SOS-SVR model for PCB throughput estimation in smt production lines, 2019 *International Conference on Industrial Engineering and Systems Management* (*IESM*), 2019, 1–6.

[7] Kundu S, Mak T, and Galivanche R, Trends in manufacturing test methods and their implications, 2004 *International Conference on Test*, 2004, 679–687.

[8] Yang J, Li S, Wang Z, et al., Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges, *Materials*, 2020, **13**(24): 5755, DOI: 10.3390/ma13245755.

[9] Liang Y, Sun Z, Song T, et al., Lenovo schedules laptop manufacturing using deep reinforcement learning, *INFORMS Journal on Applied Analytics*, 2022, **52**(1): 56–68.

[10] Li Y, Wang K, Kang Y, et al., Board-level functional test selection based on fault tree analysis, 2023 6*th International Symposium on Autonomous Systems* (*ISAS*), IEEE, Nanjing, China, 2023, 1–6.

[11] Qian F, Smart and optimal manufacturing: The key for the transformation and development of the process industry, *Engineering*, 2017, **3**(2): 151, DOI: 10.1016/J.ENG.2017.02.016.

[12] Bogle I D L, A perspective on smart process manufacturing research challenges for process systems engineers, *Engineering*, 2017, **3**(2): 161–165.

[13] Qian F, Zhong W, and Du W, Fundamental theories and key technologies for smart and optimal manufacturing in the process industry, *Engineering*, 2017, **3**(2): 154–160.

[14] Jiyuan Z, Baicun W, Liu M, et al., Brief analysis on three basic paradigms of intelligent manufacturing, *Chinese Journal of Engineering Science*, 2018, **20**(4): 13–18.

[15] Evjemo L D, Gjerstad T, Grøtli E I, et al., Trends in smart manufacturing: Role of humans and industrial robots in smart factories, *Current Robotics Reports*, 2020, **1**(2): 35–41.

[16] Wang B, Tao F, Fang X, et al., Smart manufacturing and intelligent manufacturing: A comparative review, *Engineering*, 2021, **7**(6): 738–757.

[17] Wang C, Wang L, Xue Y, et al., Revealing spatial spillover effect in high-tech industry agglomeration from a high-skilled labor flow network perspective, *Journal of Systems Science & Complexity*, 2022, **35**(3): 839–859.

[18] Yang Y, Zhao L, Chen S, et al., Fiducial approach for the storage reliability assessment of complex repairable systems, *Journal of Systems Science & Complexity*, 2024, **37**(4): 1653–1671.