A Birnbaum Importance and regression analysis based heuristic algorithm for multi-type component assignment problem

Zhiling Guan, Yida Shi and Siqi Qiu
Department of Industrial Engineering, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China
E-mail: siqiqiu@sjtu.edu.cn

Abstract. In this paper, a new kind of multi-type component assignment problem (MCAP) called the two types of component assignment problem (TCAP) is proposed. The problem is to assign two types of components to three types of positions in a system in order to obtain high system reliability. The number of each type of components equals the number of positions in the system where the components can be placed in. A new heuristic algorithm, called the REG-ZK method, based on Birnbaum Importance and regression analysis for solving the new MCAP, is illustrated in the paper. Comprehensive numerical experiments on small systems and large systems are conducted to evaluate the optimization effect of the method. Compared to the enumeration and randomization method, the REG-ZK method proves to be applicable and efficient with good optimization quality.

1. Introduction

The component assignment problem (CAP) is an important problem for optimizing the system reliability, which aims to find an optimal or near-optimal permutation in a system with \( n \) same type of positions given \( n \) components with different reliabilities. These \( n \) components are of the same type as well, and can be placed in any one of the \( n \) positions.

Suppose there is a system with \( n \) different types of positions, and each position can place a corresponding type of component. Among these positions, some positions can only place a specific type of component, while other positions can place the component of more than a type. This problem is a branch of CAP, which is called the multi-type component assignment problem (MCAP). The two types of component of component assignment problem (TCAP) is a branch of MCAP. TCAP in this paper, refers to the situation where there are only two types of component to be assigned, and three types of positions to be placed in. Two types of the positions can be placed in only one corresponding type of component, while the rest type of positions can be placed in both types of the components. It is assumed that the number of each type of components equals the total number of positions where this type of components can be placed. The paper proposes a method called REG-ZK for solving TCAP.

Existing research focused on the problem of single type component assignment problem (SCAP). In terms of SCAP, it went through a process from local optimization to global optimization. Zuo et al. [1] introduced the concept of Birnbaum Importance (BI) which describes positions’ importance to system reliability, and ZK, a local optimization method improving the system reliability of the
existing component allocation by further adjusting the permutation, was proposed. Birnbaum Importance (BI) is one of the most widely used importance measures in CAP and it assesses the contribution of a component, which is assigned to one of the positions in a system, to the whole system reliability. According to Birnbaum [2], the Birnbaum Importance of the component in position \(i\) is calculated as Equation (1).

\[
BI_i = \frac{\partial R_\pi}{\partial r_{\pi_i}}
\]  

(1)

In the above equation, \(R_\pi\) represents the system reliability under the permutation of \(\pi\). \(L_i\) represents the label of the component placed in the position \(i\), and \(r_{\pi_i}\) represents the reliability of component \(L_i\). Later, Lin et al. [3] proposed a BI-based method called LK, which helps generate an initialization allocation. It made up the blanks of the previous research by finding ways to obtain a relatively reliable initial allocation. Afterwards, Yao et al. [4] suggested the two-stage BIT approach that combines LK, which generates the initial arrangement, and ZK, which further optimizes the arrangement generated by LK. So far, the local optimization of SCAP has been implemented, which means the component with the label of \(L_i\) is assigned to position \(i\). \(m+k\) represents the label of the component placed in the position \(i\), and \(r_{\pi_i}\) represents the reliability of component \(L_i\). The component preemption in the system can only place type A components is \(m\), and these positions are denoted as PA = \((P_{A1}, P_{A2}, P_{A3}, \ldots, P_{Am})\). The number of positions where can only place type B components is \(l\), and they are denoted as PB = \((P_{B1}, P_{B2}, P_{B3}, \ldots, P_{Bl})\). The number of positions where can place type A as well as type B components is \(k\), and they are denoted as PAB = \((P_{A1B}, P_{A2B}, P_{A3B}, \ldots, P_{AB})\). These positions consist of all the positions in the system, and they are described as \(P = (P_{A1}, P_{A2}, P_{A3}, \ldots, P_{Am}, P_{B1}, P_{B2}, P_{B3}, \ldots, P_{Bl}) = (P_{1}, P_{2}, \ldots, P_{n})\) \((1, 2, 3, \ldots, n)\), \(n \in Z\). There are \(m+k\) type A components with reliabilities of \(r_1, r_2, r_3, \ldots, r_{m+k}\) to be allocated, and they are labelled as LA = \((1, 2, 3, \ldots, m+k)\). Respectively, their reliabilities are RLA = \((r_1, r_2, r_3, \ldots, r_{m+k})\). There are \(k+l\) type B components with reliabilities of \(r_{m+k+1}, r_{m+k+2}, r_{m+k+3}, \ldots, r_{m+2k+l}\) to be allocated, and they are labelled as LB = \((m+k+1, m+k+2, \ldots, m+2k+l)\). Respectively, their reliabilities are RLB = \((r_{m+k+1}, r_{m+k+2}, r_{m+k+3}, \ldots, r_{m+2k+l})\). The component preemption in the system with position \(P\) is denoted as \(\pi\), \(\pi = (L_1 L_2 L_3 \ldots L_n)\), \(L_x \in Z\), which means the component with the label of \(L_1\) is placed in the position 1, and the component with the label of \(L_2\) is placed in the position 2, and so on, in which case the reliabilities of components are \(R_\pi = (r_{l1}, r_{l2}, r_{l3}, \ldots, r_{ln})\). In addition, the order of BI corresponding to \(\pi\) is denoted as \(I\), \(I = (i_1, i_2, \ldots, i_n)\), \(i_x = 1, 2, \ldots, n\). The system reliability under the permutation of \(\pi\) is presented as \(R_\pi\).

2. A two-stage method for MCAP (REG-ZK)
In this section, the REG-ZK method is proposed and described, which consists of two stages and six steps. It is a two-stage algorithm: REG method, a regression method based on TCAP, is used to generate the initial permutation, and ZK algorithm is created for the subsequent further optimization.
2.1. REG method for initialization
The location of a position in the system, the reliability of the component placed in this position, and the arrangement of the rest components in the system determine the BI of this position. Therefore, the system location where a position is located reflects the “contribution” of this position to the system reliability from one of the perspectives. However, the impact of the reliability of the component placed in this position and the arrangement of the rest components on the BI of this position is difficult to estimate. REG method is based on regression analysis, aiming to obtain a “contribution” ranking. As mentioned above, the contribution ranking has the disadvantage of ignoring the influence of the reliability of the component placed in the position and the arrangement of the rest components, but it to some extent reflects the properties of the position’s location itself. Different from BI, the contribution and the contribution ranking are only related to the position’s location in the system. The REG initialization steps are shown as follows.

Step 1. Regression analysis
A large number of random numbers are used for regression analysis. The reliability of the components placed in the system is used as the independent variables, and the system reliability (SR) is used as the dependent variable. A multivariate linear regression formula is obtained by linear fitting. Since REG method aims to reduce the impact of the component reliability placed at the position and the arrangement of the rest components, focusing only on the properties of one specific position’s location in the system, it is unnecessary to be particular about the types and reliability of the components when generating random permutations. The method of generating random permutations is as follows. A large number of sets of random numbers between 0 and 1 are considered as the reliability of components are generated first, which are assigned to the system positions in a random order then. Respectively, the order of the reliabilities of the components placed in the system is $R\pi$, and $R\pi = (r_{11}, r_{12}, \ldots, r_{1n})$.

Equation (2) shows the formula obtained by linear regression.

$$SR = c_1r_{11} + c_2r_{12} + \ldots + c_nr_{1n} + C, c_1, c_2, \ldots, c_n, C \in R.$$  \hspace{1cm} (2)

Step 2. Obtaining the contribution ranking
The position’s location corresponding contribution is measured by the position’s location corresponding slope, that is, the larger the slope, the greater the contribution; the smaller the slope, the smaller the contribution. Slopes are sorted from big to small to obtain the formula.

$$SR = c_1r_{11} + c_2r_{12} + \ldots + c_nr_{1n} + C, c_1, c_2, \ldots, c_n, C \in R.$$  \hspace{1cm} (3)

The positions corresponding to $r_{11}, r_{12}, \ldots, r_{1n}$ are $p_{c_1}, p_{c_2}, p_{c_3}, \ldots, p_{c_n}$, that is, the position of $p_{c_1}$ contributes to the system reliability the most, and the position of $p_{c_n}$ contributes the least.

Step 3. Assigning components in descending order of contribution
First, the determination which position unassigned has the largest contribution is made based on the ranking obtained in step 2. The type of the position is found out and after that among the corresponding type of components, the unassigned component with maximum reliability is selected. This component is put into the position then, and this position as well as this component are marked as assigned, which cannot enter the subsequent assigning process in step 3. Step 3 is repeated until all positions are assigned components and the initial permutation is obtained.

2.2. ZK method for further optimization
After the initialization is completed with REG method, ZK method based on the ZKC and ZKD algorithms which were proposed in the study in 2011 [4], is used in further optimization. To suit TCAP, several modifications are conducted. More precisely, modified ZKC method (MZKC) and modified ZKD method (MZKD) are designed for further optimization.

The MZKC method is a method based on the component reliability. First, the system reliability is calculated under the current permutation. Then the most reliable unsearched component and the second most reliable unsearched component in the system are found out and exchanged. The system
reliability is second calculated after the exchange. If the system reliability improves, then the exchanged permutation is adopted; otherwise, the original permutation is maintained. This most reliable component is then marked as searched, and the original second most reliable component becomes the most reliable component among the unsearched components, and after that the same process is performed, until all the components have been searched.

The MZKD method is a BI based method. First the system reliability under the current permutation is calculated. Then among the unsearched components in the system, the component at the position with the highest BI and the component at the position with the second highest BI are found out and exchanged to calculate the system reliability after the exchange. If the system reliability improves, then the exchanged permutation is adopted; otherwise, the original permutation is maintained. The component at the position with the highest BI is then marked as searched, and after that the same process is performed, until all the components have been searched.

The steps of the ZK method, that is, the MZKC-MZKD loop algorithm are shown as follows.

**Step 4. MZKC method**
Components placed in the positions where only a single type of components can be placed and their positions are considered as different “single-type small systems”. The positions where two types of components can be placed are classified into the above-mentioned single-type small systems according to the type of components placed thereon. As a result, two extended small systems are created. MZKC method is first sequentially performed on these two “extended small systems”. Then MZKC method is performed on the positions where two types of components can be placed. Repeat step 4 until the permutation no longer changes.

**Step 5. MZKD method**
Similar to step 4, the MZKD method is performed on the extended small systems and positions where two types of components can be placed in sequence. Repeat step 5 until the permutation no longer changes.

**Step 6. Returning to Step 4**
Go back to Step 4. If the permutation no longer changes, then break out of this step 4-step 5 cycle, and return the optimized permutation.

3. **Experiments and comparisons**
In this section, the application of the REG-ZK method in 10 small systems and 6 big systems is presented. Through the comparisons between REG-ZK and the enumeration/randomization method, REG-ZK is proved to be effective and efficient.

Assume that there are three types of components with different reliability distributions. The reliability is uniformly distributed among \([0.8,0.99]\), \([0,0.2]\), \([0,0.99]\), which are respectively denoted as type I, type II, and type III components.

In order to evaluate the optimization effect of the algorithm, for small systems, the optimization results obtained by REG-ZK are compared with the results obtained by the enumeration method. For large systems that are difficult to enumerate all the possibilities, the optimization results obtained by REG-ZK are compared with the best and worst solutions obtained from massive sets of random simulations. When the randomization method is used, 10,000 simulations are performed for each optimization.

One of the evaluation criteria is SSR. “Optimal” stands for the system reliability of the optimal solution obtained by the enumeration or randomization method, “worst” stands for the system reliability of the worst solution obtained by the enumeration or randomization method, and “heuristics” represents the system reliability of the optimal solution obtained by REG-ZK. Equation (4) shows the calculation rule of SSR.

\[
SSR = \frac{\text{heuristics} - \text{worst}}{\text{optimal} - \text{worst}}
\]
For each system, 100 experiments are performed for each type of component, and the average of the SSR of these 100 experiments is called MSSR.

The experiments involve 10 small systems and 6 big systems. The small systems are C1, C2, C3, C4, Lin/Con/2/7: G, Lin/Con/2/8: G, Lin/Con/3/7: G, Lin/Con/3/8: G, Lin/Con/3/7: F, and Lin/Con/3/8: F. The big systems are the Lin/Con/20/30: G (BS1), Lin/Con/20/30: F (BS2), Lin/Con/30/40: G (BS3), Lin/Con/30/40: F (BS4), Lin/Con/40/50: G (BS5), and Lin/Con/40/50: F (BS6). The systems of C1, C2, C3, C4 are presented in figure 1.

**Figure 1.** Diagrams of systems: (a) C1; (b) C2; (c) C3; (d) C4.

In actual calculations, the accuracy of the experiments is 12 digits after the decimal point, but for the sake of display, the paper only presents four digits after the decimal point.

### 3.1. An analytical example
Take a component allocation problem of the C1 system as an example. Assume that

\[ p_{Ax} = (1, 2), \quad p_{Bx} = (6), \quad p_{ABx} = (3, 4, 5) \]

There are now totally 9 assignable components with the reliability type of type I. There are 5 type A components with the label of \( L_A = (1, 2, 3, 4, 5) \), and their reliabilities are 0.9823, 0.9805, 0.9543, 0.8792, 0.8621 respectively. There are 4 type B components with the label of \( L_B = (6, 7, 8, 9) \), and their reliabilities are 0.9688, 0.9282, 0.8263, 0.8166 respectively. The following are detailed steps to put these components into the C1 system to obtain high system reliability using the REG-ZK method.

1) **Obtaining the contribution ranking through regression**
First, 10,000 sets of random numbers are generated, and a linear regression formula of system reliability and component reliabilities is obtained. Equation (5) shows the linear regression formula. Its t value proves that the performance is significant, indicating the results of this regression are reliable. Sorting in descending order, the contribution ranking is then obtained.

\[
SR = 0.6182r_{r_1} + 0.1831r_{r_2} + 0.2486r_{r_3} + 0.1201r_{r_4} + 0.1249r_{r_5} + 0.0616r_{r_6} - 0.3661
\]

2) **Initializing according to the assigning rules in the REG method**
According to contribution ranking, the first position contributes the most and should be allocated first, that is \( p_{A1} \). The component among the unassigned ones with the maximum reliability is the one with the label of 1, and as a result \( r_{r_1} = r_1 = 0.9823 \). Then the third position is allocated, that is \( p_{AB1} \). Among all the unassigned components, the one with the maximum reliability is the type A component with the reliability of 0.9805 and a label of 2, so \( r_{r_3} = r_2 = 0.9805 \). After that the second position is assigned. The maximum reliability of the unassigned type A components is 0.9543, so \( r_{r_2} = r_1 = 0.9543 \). This component has a label of 3. Similarly, the component allocation for the remaining three positions is performed. The initial permutation obtained is \( \pi = (1\ 3\ 2\ 7\ 6\ 8) \). Detailed operation is presented in table 1.
Table 1. The initialization process with the REG method.

| Current maximum slope | Position to be assigned | Unassigned components of the corresponding type | π |
|-----------------------|-------------------------|-----------------------------------------------|---|
| 0.6182                | \( A_1 \) =1           | \( 1 \ 2 \ 3 \ 4 \ 5 \)                       | \( 1 \) |
| 0.2486                | \( A_{AB_1} \) =3      | \( 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \)           | \( 1 \ 2 \) |
| 0.1831                | \( B_2 \) =2           | \( 3 \ 4 \ 5 \)                                | \( 1 \ 3 \ 2 \) |
| 0.1249                | \( A_{AB_3} \) =5      | \( 4 \ 5 \ 6 \ 7 \ 8 \ 9 \)                   | \( 1 \ 3 \ 2 \ 6 \) |
| 0.1201                | \( A_{AB_2} \) =4      | \( 4 \ 5 \ 7 \ 8 \ 9 \)                       | \( 1 \ 3 \ 2 \ 7 \ 6 \) |
| 0.0616                | \( B_1 \) =6           | \( 8 \ 9 \)                                   | \( 1 \ 3 \ 2 \ 7 \ 6 \ 8 \) |

3) Optimizing with MZKC method

According to the existing initial permutation \( \pi \), two extended small systems can be obtained, namely extended small system 1 \( (P_{A_1}, P_{A_2}, P_{AB_1}) \) and extended small system 2 \( (P_{A_2}, P_{AB_3}, P_{A_1}) \). First, MZKC method is performed on the extended small system 1 and the result of the maintenance of the original permutation is obtained. Then MZKC method is performed on the extended small system 2. The detailed operation is presented in table 2.

Table 2. Process of MZKC method performed on the extended small system 2.

| Permutation | System reliability | After exchange | Permutation | System reliability | Exchange? |
|-------------|--------------------|----------------|-------------|--------------------|-----------|
| Cycle 1     |                    |                | Cycle 1     |                    |           |
| \( 7 \ 6 \ 8 \) | .981001791113395 | (6 7 8)        | .981001791113396 | Yes               |
| \( \setminus 7 \ 8 \) | .981001791113396 | \( \setminus 8 \ 7 \) | .980943777854950 | No                |
| Cycle 2     |                    |                | Cycle 2     |                    |           |
| \( 6 \ 7 \ 8 \) | .981001791113396 | (6 7 8)        | .981001791113395 | No                |
| \( \setminus 7 \ 8 \) | .981001791113396 | \( \setminus 8 \ 7 \) | .980943777854950 | No                |

* If the permutation is same as the one generated by the last cycle, the processing ends.

The same MZKC method is performed on the positions where two type components \( (P_{AB_1}, P_{AB_2}, P_{AB_3}) \) can be placed. The permutation after using the MZKC method is \( \pi' = (1 \ 3 \ 2 \ 6 \ 7 \ 8) \).

4) Optimizing with MZKD method

Table 3. Process of MZKD method performed on the extended small system 1.

| Permutation | System reliability | I | After exchange | Permutation | System reliability | Exchange? |
|-------------|--------------------|---|----------------|-------------|--------------------|-----------|
| Cycle 1     |                    |   | Cycle 1        |             |                    |           |
| \( 1 \ 3 \ 2 \) | 0.9810             |   | \( 0.9987, 0.0191, 0.0467 \) | \( 2 \ 3 \ 1 \) | 0.9793 | No          |
| \( 3 \ 2 \)  | 0.9810             |   | \( \setminus 0.0191, 0.0467 \) | \( \setminus 2 \ 3 \) | 0.9810 | No          |

* If the permutation is same as the one generated by the last cycle, the processing ends.

Since the two components exchanged are both type B components, the expanded small system 1 and the expanded small system 2 are not changed at this time. First, MZKD method is performed on the extended small system 1. Detailed operation is presented in table 3.

The same MZKD method is performed on the expanded small system 2 and the positions where two type components can be placed. The permutation after using the MZKD method is \( \pi' = (1 \ 3 \ 2 \ 6 \ 7 \ 8) \).
5) Cyclically performing MZKC and MZKD method
Since $\pi'$ does not equals $\pi$, return to 3) until the permutation no longer changes. The permutation obtained finally is $\pi' = (1 \ 3 \ 2 \ 6 \ 7 \ 8)$.

The maximum value of the system reliability obtained by the enumeration method is 0.9812, and the minimum value is 0.8410. SSR for this experiment is calculated as equation (6).

$$SSR = \frac{0.9810 - 0.8410}{0.9812 - 0.8410} = 0.9986.$$  (6)

3.2. Numerical examples
Experiments on small systems and large systems are conducted to evaluate the optimization performance of REG-ZK, compared with the enumeration / randomization method. Results of the experiments on small systems and large systems are listed separately.

3.2.1. Experiments on small systems. The results are shown in table 4.

| System | $p_{AX}$ | $p_{BX}$ | $p_{ABX}$ | MSSR(I) | MSSR(II) | MSSR(III) |
|--------|----------|----------|-----------|---------|----------|-----------|
| C1     | (1,2)    | (6)      | (3,4,5)   | 0.9989  | 0.9836   | 0.9889    |
| C2     | (1,2)    | (5)      | (3,4)     | 0.9904  | 0.9513   | 0.9785    |
| C3     | (1,2,3)  | (8,9)    | (4,5,6,7) | 0.9936  | 0.9530   | 0.9936    |
| C4     | (1)      | (7)      | (2,3,4,5,6) | 1.0000   | 0.9996   | 0.9977    |
| G1(2/7: G) | (1) | (7) | (2,3,4,5,6) | 0.9990 | 0.9906 | 0.9955 |
| G2(2/8: G) | (1,2) | (7,8) | (3,4,5,6) | 0.9985 | 0.9896 | 0.9955 |
| G3(3/7: G) | (1) | (7) | (2,3,4,5,6) | 0.9990 | 0.9932 | 0.9968 |
| G4(3/8: G) | (1,2) | (7,8) | (3,4,5,6) | 0.9963 | 0.9848 | 0.9870 |
| F1(3/7: F) | (1) | (7) | (2,3,4,5,6) | 0.9995 | 0.9960 | 0.9990 |
| F2(3/8: F) | (1,2) | (7,8) | (3,4,5,6) | 0.9965 | 0.9899 | 0.9948 |

It can be seen that the optimization effect of most experiments is good, with MSSR $\geq 0.98$. The two poor performances with MSSR = 0.9513 and MSSR = 0.9530 are also good in the specific values. In other words, in these 200 worst-case scenarios, the system reliability after performing REG-ZK optimization is only 0.01-0.1 less than the highest achievable system reliability in value.

3.2.2. Experiments on large systems. In addition to MSSR, the computation time for optimization is also used as a criterion for evaluating REG-ZK. Numerical experiments mainly involve 6 large systems, namely BS1, BS2, BS3, BS4, BS5, and BS6. The results of REG-ZK compared with the randomization method is shown in table 5. It can be seen that the system reliability results obtained by using the REG-ZK method are all not worse than those obtained by a large number of random simulations, and are even much better than those obtained by random simulations. At the same time, in terms of computation time, in most cases, experiments using REG-ZK are more time-saving than experiments using the randomization method. For those more time-consuming experiments, the huge improvement in system reliability optimization could make up for the losses caused by the time-consuming to some extent.

While saving time, the REG-ZK method can still maintain a high optimization effect, which is especially obvious in the experiments in the large systems with a huge quantity of possible permutations. To sum up, REG-ZK is applicable, effective and efficient.

4. Conclusion
Aiming at TCAP, a heuristic algorithm, called the REG-ZK algorithm, is proposed. The REG-ZK algorithm is a two-stage optimization method. The first stage is to obtain the initial permutation using REG method based on massive random numbers and regression analysis. The second stage is to use
ZK method to improve the initial permutation generated by REG method. Through experiments in small and large systems, REG-ZK is proved to be effective and efficient, compared with the enumeration / randomization method. Future research can be conducted in the following directions:

1) Consider more types of components
2) Application of genetic algorithm on TCAP / MCAP, etc.

Table 5. The results of REG-ZK compared with randomization method.

| System | $p_{Ax}$ | $p_{Bx}$ | $p_{ABx}$ | Reliability type | MSSR | Computation time (seconds) |
|--------|----------|----------|-----------|------------------|------|---------------------------|
|        |          |          |           |                  |      | Randomization REG-ZK      |
| BS1    | [1:10]   | [21:30]  | [11:20]   | I                | 1.5671 | 24.2489   | 5.8503   |
|        |          |          |           | II               | 39.2111 | 23.2656   | 3.5643   |
|        |          |          |           | III              | 51.6218 | 23.6108   | 4.1069   |
| BS2    | [1:10]   | [21:30]  | [11:20]   | I                | 1.0000  | 24.8290   | 1.4461   |
|        |          |          |           | II               | 1.1091  | 25.6275   | 6.6170   |
|        |          |          |           | III              | 1.0000  | 24.9443   | 5.7308   |
| BS3    | [1:13]   | [28:40]  | [14:27]   | I                | 1.9982  | 26.8740   | 6.2714   |
|        |          |          |           | II               | 214.1240| 26.9961   | 18.2719  |
|        |          |          |           | III              | 1.0000  | 27.4886   | 4.7687   |
| BS4    | [1:13]   | [28:40]  | [14:27]   | I                | 1.0000  | 27.2320   | 3.7959   |
|        |          |          |           | II               | 1.1133  | 27.5928   | 17.706   |
|        |          |          |           | III              | 1.0000  | 27.4886   | 4.7687   |
| BS5    | [1:17]   | [26:50]  | [18:25]   | I                | 1.6946  | 28.8682   | 54.9876  |
|        |          |          |           | II               | 26.3519 | 28.8199   | 59.1625  |
|        |          |          |           | III              | 271.1743| 28.7277   | 59.1412  |
| BS6    | [1:17]   | [26:50]  | [18:25]   | I                | 1.0000  | 29.4080   | 7.6596   |
|        |          |          |           | II               | 1.3172  | 29.5279   | 7.5229   |
|        |          |          |           | III              | 1.0000  | 29.4298   | 7.4784   |

Acknowledgments
This work is supported by National Natural Science Foundation of China (Grant No. 51805326 and 71632008) and National Science and Technology Major Project (Grant No. 2017-I-0007-0008).

References
[1] Zuo M and Kuo W. 1990 Design and performance analysis of consecutive-K-out-of-n structure. Naval Research Logistics, 37(2) 203–30
[2] Birnbaum Z W. 1969 On the importance of different components in a multi-component system. Journal of Multivariate Analysis
[3] Lin F and Kuo W. 2002 Reliability Importance and Invariant Optimal Allocation. Journal of Heuristics 8, 155–71
[4] Yao Q, Zhu X and Kuo W. 2011 Heuristics for component assignment problems based on the Birnbaum importance, IIE Transactions, 43(9) 633-46
[5] Yao Q, Zhu X and Kuo W. 2014 A Birnbaum-importance based genetic local search algorithm for component assignment problems[J]. Annals of Operations Research, 212(jan.) 185-200
[6] Cai Z, Si S, Sun S and et al. 2016 Optimization of linear consecutive-k-out-of-n system with a Birnbaum importance-based genetic algorithm[J]. Reliability Engineering & System Safety, 152(Aug.) 248-58
[7] Zhu X, Fu Y, Yuan T and et al. 2017 Birnbaum importance based heuristics for multi-type component assignment problems[J]. Reliability Engineering & System Safety, 165 209-21