An application of integer programming to producing aircraft engine parts

Daniel Solow* and Qiong Wu

Department of Operations,
Case Western Reserve University,
Cleveland, OH 44106, USA
Email: daniel.solow@case.edu
Email: qxw205@case.edu
*Corresponding author

Dan Magri

PCC Airfoils,
3401 Enterprise Parkway, Suite 200,
Beachwood, OH 44122, USA
Email: Dan.Magri@PCCAIRFOILS.com

Abstract: Integer programming models are developed for optimising the production of a part used in aircraft engines. A real-world problem is solved to optimality; however, for some potentially large real-world problems, one of these models can require too much time, so appropriate heuristics are developed. These heuristics are shown computationally to be both effective and efficient using randomly generated data.

Keywords: integer programming application; integer programming heuristic; aircraft manufacturing.

Reference to this paper should be made as follows: Solow, D., Wu, Q. and Magri, D. (2021) 'An application of integer programming to producing aircraft engine parts', *Int. J. Applied Management Science*, Vol. 13, No. 1, pp.1–14.

Biographical notes: Daniel Solow is a Professor in the Department of Operations of the Weatherhead School of Management at Case Western Reserve University in Cleveland, Ohio. His research areas include deterministic optimisation, linear and nonlinear programming, combinatorial optimisation and modelling and analysis of complex systems.

Qiong Wu holds an MBA from the Weatherhead School of Management at the Case Western University. Her interest is in the area of finance and modelling, including financial models building and data analysis that provide insights and supports to make business decisions.

Dan Magri is a Product Manager and Engineer in the aerospace industry responsible for high pressure turbine blade development at the PCC Airfoils Company. A Cleveland native, he holds a degree in Mechanical Engineering from CSU and an MBA from Case Western.

1 Introduction

PCC Airfoils is a manufacturer of high-temperature rotating blades and stationary vanes for the hot sections of jet aircraft engines and industrial gas turbines. Made of nickel super alloys, PCC Airfoils' products must be manufactured with complex, internal cooling passages made to tight tolerances to withstand turbine temperatures of 2,400°F or more. As customers have designed engines for higher fuel efficiency and lower emissions, the casting technology within the hot section has become increasingly sophisticated, and new technologies must be developed to meet new requirements.

1.1 Overview of the problem and our approach

PCC Airfoils produces a particular part for an aircraft engine that is made from three separate components. The quality of the resulting part is measured numerically across five characteristics, each of which has an 'ideal' value, hereafter referred to as a *nominal value*. The value of each characteristic of the produced part depends on the corresponding values of that characteristic of the three components according to a formula to be discussed subsequently. To be *acceptable*, the value of each characteristic of the produced part must be between a given minimum and maximum value. Furthermore, the closer each characteristic is to the nominal value, the better the quality of that part. For a given production run, there are multiple copies of each component in inventory and so the overall goal is to determine how to use the components in inventory to produce as many acceptable parts of the highest overall quality as possible.

The first approach tried for solving this problem was to create an integer programming model to select a single set of components 1, 2, and 3 from inventory that would result in an acceptable part of the highest possible quality. The selected components are then removed from the inventory and the process of selecting the next-best set of the three components from the remaining inventory is repeated. Experimentation with this greedy approach on random data, however, revealed two drawbacks:

1. there is a large variation in the quality of the resulting parts, with the first few parts being of very high quality and the last few parts being of significantly lesser quality
2. as the inventory is depleted, it is possible to reach a point where it was no longer possible to produce an acceptable part with the remaining components.

As a result of the foregoing deficiencies, a different approach – described in Section 3 – was developed. A more complete problem description – including how a part's quality is determined – is presented in Section 2. The solution methodology is given in Section 3. Computational experience with this approach is presented in Section 4.

1.2 Literature review

In the airplane industry, mathematical models have been heavily relied on for decision making in many areas – such as component design, industrial design, supply chain management, and operations management (Stecke, 2005). However, these models are quite different from the optimisation model in this article. One difference is the type of model used. For example, many design and supply-chain problems are modelled using simulation [see, for example, Bottasso et al. (2013)] for one rotor-blade design problem while the model for the problem studied here is an integer program. Another difference is that other production problems that use raw materials to make a final product categorise those raw materials as either defective or acceptable [see Sanchez-Rodrguez and Martinez-Lorente (2004) for example]. In contrast, the raw materials for the problem studied here – that is, the inventory of components – each have a numerical measure of their quality and the goal is to decide how to use that inventory in such a way that the resulting parts that are produced collectively have the best possible quality. Surprisingly, a similar problem arises in the pharmaceutical industry (Fonteyne et al., 2014), where the quality of the final drug produced also depends on the characteristics of the constituents that make up the drug. The constituents in the pharmaceutical problem are analogous to the inventory components in the problem studied here. While Fonteyne et al. (2014) recognise that properties of the raw materials have a great influence on the quality of the final product, no practical solution was given as to how best to use the raw materials to produce the final product. We are hopeful that the type of model developed here could be applied in the pharmaceutical industry as well.

As described in Section 3, it is computationally impractical to obtain an optimal solution to one of the integer programming models presented here. As a result, an efficient heuristic was developed. This heuristic is based on the well-known 'divide-and-conquer' strategy [other examples of this type of heuristic include Benders decomposition (Benders, 1962) and sort-merge (Al-Kharabsheh et al., 2013)]. More specifically, the heuristic developed here is to decompose a problem consisting of a large number of available inventory components into a collection of smaller problems – each having only a few inventory components – for which optimal solutions can be found and then assembled into an approximate solution for the original problem. Details of this heuristic are given in Section 3.

2 Detailed problem description

As mentioned in Section 1, the quality of a produced part is measured numerically across five characteristics, labelled here as A, B, C, D, and E. Each characteristic has an ideal value – called the *nominal value* – as given for a real-world numerical example in the second column of Table 1. For a part to be *acceptable*, the value of each characteristic must be between a given minimum and maximum value, as shown in the last two columns of Table 1. The closer the values of the characteristics are to their nominal values, the better the quality of the part. Thus, an overall measure of the quality of an acceptable part is the sum, over the five characteristics, of the absolute values of the difference between the part's characteristic value and that characteristic's nominal value – the smaller this *deviation*, the higher the quality. Using the nominal values in Table 1, the deviation of a part with characteristic values A through E in Table 1 is:

$$\text{deviation} = |A - 0.034| + |B - 0.050| + |C - 0.026| + |D - 0.026| + |E - 0.026| \quad (1)$$

The numerical values of characteristics A through E for a part are determined by the corresponding values of the three components with which the part is made. To be more specific, each component in inventory has a known numerical value for the five characteristics that represents the deviation of that characteristic from the nominal value –

a positive deviation means the value of the component's characteristic is larger than the nominal value and a negative value means the value of the component's characteristic is below the nominal value. An example of an inventory consisting of five copies of each component is presented in Table 2, in which each copy is given a unique *serial* for identification purposes.

Table 1 Nominal, minimum, and maximum values of the characteristics

Characteristic	*Nominal value*	*Minimum value*	*Maximum value*
A	0.034	0.024	∞
B	0.050	0.040	∞
C	0.026	0.015	∞
D	0.026	0.015	∞
E	0.026	0.015	∞

Table 2 sample inventory with component characteristic deviations from nominal

Serial	*Deviations from nominal*				
	A	*B*	*C*	*D*	*E*
			Component 1		
L1	0.002	–0.001	0.002	0.000	–0.002
L2	0.001	0.001	0.003	0.000	0.000
L3	0.004	0.001	0.003	0.000	0.000
L4	0.002	0.000	0.004	0.000	0.000
L5	0.000	0.000	0.000	0.000	–0.001
			Component 2		
M1	–0.005	–0.001	0.011	0.002	–0.007
M2	0.009	0.009	0.012	0.006	0.003
M3	0.005	0.006	0.012	0.003	0.000
M4	0.004	0.004	0.014	0.005	–0.002
M5	0.004	0.003	0.007	0.003	–0.001
			Component 3		
T1	0.000	0.000	0.000	–0.006	0.000
T2	0.000	0.000	0.000	–0.002	0.000
T3	0.000	0.000	0.000	–0.002	0.000
T4	0.000	0.000	0.000	–0.001	0.000
T5	0.000	0.000	0.000	–0.003	0.000

When serials from inventory are chosen for each of the three components, the numerical value of a characteristic of the resulting part is the nominal value plus the sum of the deviations of that characteristic for each of the three chosen components. For example, if serials L1, M2 and T3 are chosen for components 1, 2, and 3, respectively, then the value of characteristic A, whose nominal value from Table 1 is 0.034, of the resulting part is:

$$A = 0.034 + 0.002 + 0.009 + 0.000 = 0.045.$$

The values of the remaining characteristics B through E of the part are computed in the same way which, for the current serials L1, M2 and T3, result in the following values of the five characteristics of the part:

$$\begin{aligned} A &= 0.034 + 0.002 + 0.009 + 0.000 = 0.045 \\ B &= 0.050 - 0.001 + 0.009 + 0.000 = 0.058 \\ C &= 0.026 + 0.002 + 0.012 + 0.000 = 0.040 \\ D &= 0.026 + 0.000 + 0.006 - 0.002 = 0.030 \\ E &= 0.026 - 0.002 + 0.003 + 0.000 = 0.027 \end{aligned} \tag{2}$$

The foregoing part obtained by using components L1, M2, and T3 is acceptable because all of the characteristic values in (2) are between the minimum and maximum allowed values, as given in the last two columns of Table 1 – had that not been the case, this part would not have been acceptable and so components L1, M2, and T3 could not be used to make the part.

With the foregoing understanding, the overall goal is to determine how to use the components in inventory to produce as many acceptable parts as possible of the highest overall quality, as measured for each part by the deviation in (1). The approach adopted for doing so is described in the next section.

3 Solution procedure

Careful thought reveals that there are two separate goals associated with solving the problem presented in Section 2:

- *Phase 1: Determining the maximum number of acceptable parts* – in this phase, the maximum number, say K, of acceptable parts that can be produced with the given inventory is determined.
- *Phase 2: Determining how best to use the inventory* – once it is known that K acceptable parts can be produced, the next step is to determine how to use the components in inventory to produce K acceptable parts that, collectively, have the highest quality. Recalling that the quality of a part is measured numerically by the total deviation of the characteristics from their nominal values, as defined in (1), the goal of phase 2 is to determine how to produce K acceptable parts the sum of whose deviations is as small as possible.

The integer programming (IP) models in Appendix A were created for solving phase 1 and phase 2 problems. Computational experiments indicated that while the IP model for phase 1 was quite efficient for solving real-world problems – in which the inventory typically consists of between 20 and 30 serials for each component – unfortunately, the IP for phase 2 required too much time. It was therefore necessary to create a heuristic for solving phase 2.

The heuristic proposed here is based on the computational observation that solving the phase 2 IP to optimality is acceptably fast when there are no more than seven serials. This led to the following improvement heuristic that starts with the K acceptable parts obtained at the end of solving phase 1 and successively produces K acceptable parts of increasing quality, as measured by smaller total deviation:

Step 1 Divide the K acceptable parts from phase 1 into groups of four (except, possibly, for one group with fewer than four parts).

Step 2 For each group, using only the components in that group, solve the phase 2 IP to obtain a collection of those four parts that has the least total deviation.

Step 3 Sort all of the K parts obtained from step 2 in increasing order of deviation.

Step 4 Repeat steps 1, 2, and 3 so long as the total deviation of the K parts decreases.

When done, a collection of K acceptable parts with a (hopefully) small total deviation is obtained, based on working with groups of four parts. The foregoing steps are then repeated using groups consisting of 5, 6, and 7 parts, respectively, and the solution that results in the overall smallest total deviation is presented as the final solution. Of course the user is allowed to put a maximum time limit on how long the program is allowed to run.

4 Computational experience

The data for a real-world problem consisting of 25 serials is given in Appendix B. Applying the integer programs in Appendix A, an optimal solution was found after 6 seconds for the phase 1 IP and 4 minutes and 38 seconds for the phase 2 IP, resulting in an optimal objective function value of 0.561. In general, however, it is expected that solving the phase 2 IP to optimality can require too much time. To determine the efficacy of the heuristic in general, a variety of different computational experiments were performed, as described next.

4.1 Computational results with problems involving random data

The first group of experiments involved problems with randomly generated data. The average optimal objective function values obtained from running the heuristic on 30 problems containing 15, 20, 25, and 30 serials for 10 minutes each are compared with those obtained from solving the same problems using the integer programming models in Appendix A for the same amount of time. Both solution procedures are solved in VBA with OpenSolver (Mason, 2012) (see also the website https://opensolver.org) which uses the COIN-OR CBC optimisation engine for solving the integer programs.

For each problem, the five characteristic deviations from their nominal values for each of the three components are obtained by dividing a uniform random integer between –10 and +10 by 1000. For a fixed number of serials (15, 20, 25, and 30), the average objective function values using the two approaches are reported in Table 3a. As seen there, the heuristic obtains a better average objective function value for all cases except 15 serials, with the amount of improvement increasing roughly as the number of serials increases. Table 3b contains the number of problems won by each of the two methods, resulting in the same conclusion that the heuristic wins on the majority of problems in all cases except for 15 serials.

Table 3 Comparison of the heuristic and the IP for different numbers of serials

a Average objective function values			*b Number of wins*		
Serials	*Heuristic*	*Original IP*	*Serials*	*Heuristic*	*Original IP*
15	0.340	0.326	15	10	20
20	0.397	0.418	20	20	10
25	0.513	0.521	25	19	11
30	0.597	0.638	30	23	7

4.2 Computational results with problems involving perfect data

The final set of experiments addresses the questions of how close the objective function value found by the heuristic is to that of the optimal solution for 30 problems, each having 35 serials. However, for a random such problem, it is not possible to find the optimal solution, so instead, problems were generated in which the optimal objective function value is known to be 0 (hereafter referred to as perfect data). To see how such data are created, suppose that for serials L1, M1 and T1, A_{L1}, A_{M1} and A_{T1} are the respective deviations of characteristic A from the nominal value. Then recall from Section 2 that the deviation of characteristic A in the finished part is:

$$A_{L1} + A_{M1} + A_{T1}. \tag{3}$$

If the sum in (3) is 0, then characteristic A in the finished part is *perfect*, that is, there is no deviation of this characteristic from the nominal value. For the computational experiments, the values of A_{L1}, A_{M1} and A_{T1} are generated so that their sum is 0, as follows:

- generate a random integer, m, between –10 and 10 and set $A_{L1} = m/1{,}000$
- generate a random integer, n, between –10 and 10 and set $A_{M1} = n/1{,}000$
- compute $A_{T1} = -A_{L1} - A_{M1}$.

The data for the remaining characteristics B, C, D, and E for each component are generated in a similar manner. When the data are generated this way, the part obtained from serials L1, M1, and T1 is perfect, that is, the deviations of all five characteristics in the finished part are 0.

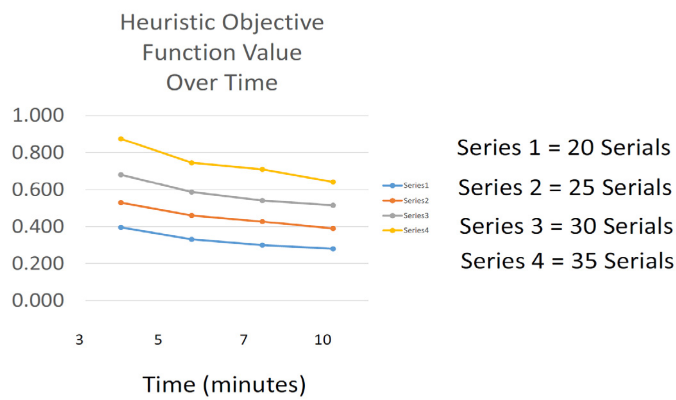
The foregoing process is used to generate all of the data for 30 problems, each containing 35 serials. Once these data are created, the serials are shuffled and the heuristic is applied for 3, 5, 7, and 10 minutes. The objective function value found is recorded and the average over all 30 problems is then compared to 0, which is the best possible value if the ideal combination of serials is chosen in each problem, providing the results shown in Table 4.

Table 4 Average deviation for 30 problems of 35 serials with perfect data

	3 min.	*5 min.*	*7 min.*	*10 min.*
Heuristic average deviation	0.829	0.724	0.680	0.632

Computational experiments also show that similar results are obtained when there are fewer serials. In these experiments, perfect data are generated for 10 problems each having 20, 25, 30, and 35 serials. These problems are solved by the heuristic using 3, 5, 7, and 10 minutes of time. The averages of the optimal objective function values over the 10 problems for each number of serials are shown in Figure 1 where it is seen that the overall pattern is similar.

Figure 1 Heuristic objective function value over time for different numbers of serials (see online version for colours)



5 Summary

In this work, an optimisation approach is developed to identify how best to use an inventory of three components to produce as many acceptable aircraft parts having the best overall quality, as defined across five separate characteristics. To that end, a two-phase approach is proposed, in which an integer programming model in phase 1 is solved to identify the maximum number, K, of acceptable parts that can be produced with the given inventory of components. Then, in phase 2, the solution to another integer programming model determines how to use that inventory to produce K parts having the highest quality – defined as the sum over the K parts, of the total amount by which each part's characteristics deviate from given nominal values. Due to the fact that the computational time needed to solve the phase 2 IP could be excessive for some real-world applications, an improvement heuristic is created and shown in practice – on both real-world and random problems – to produce a collection of K parts of good quality in reasonable amounts of computer time. When there are more than 15 serials, computational results show that the quality of the parts obtained by the heuristic is increasingly better as the size of the inventory increases than that obtained by solving the integer programs in Appendix A using the same amount of time.

Acknowledgements

We thank Nishant Barlinge for running some of the preliminary computational experiments that took many hours and led to the results that are included here.

References

Al-Kharabsheh, K.S., AlTurani, I.M., AlTurani, A.M.I. and Zanoon, N.I. (2013) 'Review on sorting algorithms a comparative study', *International Journal of Computer Science and Security*, Vol. 7, No. 3, p.120.

Benders, J.F. (1962) 'Partitioning procedures for solving mixed-variables programming problems', *Numerische Mathematik*, Vol. 4, No. 4, pp.238–252.

Bottasso, C., Campagnolo, F., Croce, A. and Tibaldi, C. (2013) 'Optimization-based study of bend-twist coupled rotor blades for passive and integrated passive/active load alleviation', *Wind Energy*, Vol. 16, pp.1149–1166, Doi:10.1002/we.1543.

Fonteyne, M., Wickstrm, H., Peeters, E., Vercruysse, J., Ehlers, H., Peters, B.H. and Rantanen, J. (2014) 'Influence of raw material properties upon critical quality attributes of continuously produced granules and tablets', *European Journal of Pharmaceutics and Biopharmaceutics*, Vol. 87, No. 2, pp.252–263.

Mason, A.J. (2012) Opensolver – an open source add-in to solve linear and integer progammes in Excel', in Schmedders, K., Klatte, D. and Luthi, H.J. (Eds.) *Operations Research Proceedings 2011. Operations Research Proceedings (GOR (Gesellschaft fur Operations Research e.V.))*, pp.401–406, Springer, Berlin, Heidelberg.

Sanchez-Rodrguez, C. and Martnez-Lorente, R. (2004) 'Quality management practices in the purchasing function: an empirical study', *International Journal of Operations and Production Management*, Vol. 24, No. 7, pp.666–687.

Stecke, K.E. (2005) 'Using mathematics to solve some problems in industry', *INFORMS Transactions on Education*, Vol. 5, No. 2, pp.1–8.

Appendix A

The integer programming models for solving the phase 1 and phase 2 problems in Section 3 are presented here for a general problem in which the data are:

M the number of components ($M = 3$ for the application here)

N the number of serials in inventory for each component

C the number of characteristics of each part ($C = 5$ for the application here)

δ_{ij}^{c} the deviation of characteristic c for serial i of component j ($i = 1, \ldots, N$; $j = 1, \ldots, M$; $c = 1, \ldots, C$)

$\underline{\mu}^{c}, \mu^{c}, \overline{\mu}^{c}$ the minimum, nominal, and maximum values of characteristic c ($c = 1, \ldots, C$).

An integer program for solving the phase 1 problem

The integer program given below for solving the phase 1 problem uses the following variables, in which serial $i = 1, \ldots, N$; component $j = 1, \ldots, M$; and part $k = 1, \ldots, N$:

- $x_{ij}^{k} = \begin{cases} 1, & \text{if serial } i \text{ of component } j \text{ is used to make part } k \\ 0, & \text{otherwise} \end{cases}$

- $z^k = \begin{cases} 1, & \text{if part } k \text{ is produced} \\ 0, & \text{otherwise} \end{cases}$
- w^{kc} = the value of characteristic c in part k ($c = 1, \ldots, C$).

With the foregoing $N(NM + C + 1)$ variables included as logical constraints, the solution to the following integer program – that has $N(2M + C)$ constraints other than bounds – provides the maximum number, K, of feasible parts that can be produced with the given inventory (when $M = 3$, $N = 35$, and $C = 5$, for example, there are 3,885 variables and 385 constraints):

$$\max K = \sum_{k=1}^{N} z^k$$

s.t.

- Each serial of each component is used at most once

$$\sum_{k=1}^{N} x_{ij}^k \le 1 \qquad (i = 1, \ldots, N;\ j = 1, \ldots, M)$$

- A produced part must use each of the M components

$$z^k = \sum_{i=1}^{N} x_{ij}^k \qquad (k = 1, \ldots, N;\ j = 1, \ldots, M) \tag{4}$$

- Definition of the characteristic values for each part

$$w^{kc} = \mu^c + \sum_{j=1}^{M} \sum_{i=1}^{N} \delta_{ij}^c x_{ij}^k \qquad (k = 1, \ldots, N;\ c = 1, \ldots, C)$$

- Acceptability constraints for the characteristics of each part

$$\underline{\mu}^c \le w^{kc} \le \overline{\mu}^c \qquad (k = 1, \ldots, N;\ c = 1, \ldots, C).$$

An integer program for solving the phase 2 problem

As a result of solving the phase 1 problem, it is possible to produce K acceptable parts with the given inventory. The nonlinear program given below uses the following variables to determine how to use the inventory to produce K acceptable parts having the highest quality (that is, least total deviation), in which serial $i = 1, \ldots, N$; component $j = 1, \ldots, M$; part $k = 1, \ldots, K$; and characteristic $c = 1, \ldots, C$:

- $x_{ij}^k = \begin{cases} 1, & \text{if serial } i \text{ of component } j \text{ is used to make part } k \\ 0, & \text{otherwise} \end{cases}$
- w^{kc} = the value of characteristic c in part k.

With the foregoing at most $NC(M + 1)$ variables included as logical constraints, the solution to the following nonlinear program – that has at most $2N(M + C)$ constraints other than bounds – determines how to use the inventory to produce K acceptable parts having the highest quality (that is, least total deviation):

A nonlinear model to determine inventory use

$$\min \sum_{k=1}^{N} \sum_{c=1}^{C} \left| \mu^c - w^{kc} \right|$$

s.t.

- Each serial of each component is used at most once

$$\sum_{k=1}^{N} x_{ij}^{k} \le 1 \qquad (i = 1, \ldots, N;\ j = 1, \ldots, M)$$

- Each part uses one serial for each component

$$\sum_{i=1}^{N} x_{ij}^{k} = 1 \qquad (k = 1, \ldots, K;\ j = 1, \ldots, M)$$

- Definition of the characteristic values for each part

$$w^{kc} = \mu^c + \sum_{j=1}^{M} \sum_{i=1}^{N} \delta_{ij}^{c} x_{ij}^{k} \qquad (k = 1, \ldots, K;\ c = 1, \ldots, C)$$

- Acceptability constraints for the characteristics of each part

$$\underline{\mu}^c \le w^{kc} \le \overline{\mu}^c \qquad (k = 1, \ldots, K;\ c = 1, \ldots, C)$$

- Logical constraints

  All x_{ij}^{k} variables 0 or 1; all w^{kc} variables continuous.

By introducing the following new at most $C * N$ continuous variables, it is possible to replace the nonlinear absolute-value terms in the objective function of the foregoing model so as to create an integer program with linear constraints:

$$y^{kc} = \left| \mu^c - w^{kc} \right| \text{ for characteristic } c \text{ in part } k \quad (c = 1, \ldots, C;\ k = 1, \ldots, K).$$

With the foregoing variables, together with at most $2 * N * C$ constraints to ensure that each $y^{kc} \ge \mu^c - w^{kc}$ and $y^{kc} \ge w^{kc} - \mu^c$, the foregoing nonlinear model becomes the linear integer program given below (having, for example, 875 variables and 735 constraints when $M = 3$, $N = 35$ and $C = 5$). The optimal solution to this problem determines how to use the inventory to produce K acceptable parts having the highest quality (that is, least total deviation):

A linear model to determine inventory use

$$\min \sum_{k=1}^{N} \sum_{c=1}^{C} y^{kc} \quad \left(= \sum_{k=1}^{N} \sum_{c=1}^{C} \left| \mu^c - w^{kc} \right| \right)$$

s.t.

- Each serial of each component is used at most once

$$\sum_{k=1}^{N} x_{ij}^{k} \le 1 \qquad (i = 1, \dots, N;\ j = 1, \dots, M)$$

- Each part uses one serial for each component

$$\sum_{i=1}^{N} x_{ij}^{k} = 1 \qquad (k = 1, \dots, K;\ j = 1, \dots, M)$$

- Definition of the characteristic values for each part

$$w^{kc} = \mu^{c} + \sum_{j=1}^{M} \sum_{i=1}^{N} \delta_{ij}^{c} x_{ij}^{k} \qquad (k = 1, \dots, K; c = 1, \dots, C)$$

- Acceptability constraints for the characteristics of each part

$$\underline{\mu}^{c} \le w^{kc} \le \overline{\mu}^{c} \qquad (k = 1, \dots, K; c = 1, \dots, C)$$

- Absolute value constraints

$$y^{kc} \ge \mu^{c} - w^{kc} \qquad (k = 1, \dots, K; c = 1, \dots, C)$$

$$y^{kc} \ge w^{kc} - \mu^{c} \qquad (k = 1, \dots, K, c = 1, \dots, C)$$

- Logical constraints

  All x_{ij}^{k} variables 0 or 1; all w^{kc} and y^{kc} variables continuous.

Appendix B

The characteristic deviations for the three components used in the real-world application solved in Section 4 are given in the following tables.

Table B1 Component 1 characteristics

Serial	*Deviations from nominal*				
	A	*B*	*C*	*D*	*E*
L1	0.002	–0.001	0.002	0.000	–0.002
L2	0.001	0.001	0.003	0.000	0.000
L3	0.004	0.001	0.003	0.000	0.000
L4	0.002	0.000	0.004	0.000	0.000
L5	0.000	0.000	0.000	0.000	–0.001
L6	0.002	–0.001	0.003	0.000	–0.001
L7	0.002	0.001	0.002	0.000	0.000
L8	0.003	0.001	0.002	0.000	0.000

Table B1 Component 1 characteristics

Serial	*Deviations from nominal*				
	A	*B*	*C*	*D*	*E*
L9	0.007	0.002	0.003	0.000	0.002
L10	0.003	0.004	0.006	0.000	0.003
L11	0.001	0.000	0.004	0.000	0.000
L12	0.001	0.001	0.002	0.000	0.001
L13	0.004	0.000	0.003	0.000	0.001
L14	0.004	0.002	0.003	0.000	0.001
L15	0.006	0.001	0.005	0.000	0.001
L16	0.001	0.002	0.000	0.000	0.002
L17	0.001	0.001	0.002	0.000	0.001
L18	0.001	0.001	0.002	0.000	0.000
L19	0.004	0.001	0.003	0.000	0.000
L20	–0.001	–0.001	–0.001	0.000	–0.002
L21	0.004	0.001	0.002	0.000	0.000
L22	0.000	–0.002	0.001	0.000	–0.003
L23	–0.001	–0.001	–0.001	0.000	–0.001
L24	0.001	0.001	0.002	0.000	0.002
L25	0.001	0.001	0.002	0.000	0.000

Table B2 Component 2 characteristics

Serial	*Deviations from nominal*				
	A	*B*	*C*	*D*	*E*
M1	–0.005	–0.001	0.011	0.002	–0.007
M2	0.009	0.009	0.012	0.006	0.003
M3	0.005	0.006	0.012	0.003	0.000
M4	0.004	0.004	0.014	0.005	–0.002
M5	0.004	0.003	0.007	0.003	–0.001
M6	0.003	0.003	0.008	0.002	–0.001
M7	0.003	0.003	0.008	0.002	–0.001
M8	0.004	0.004	0.010	0.002	–0.001
M9	0.000	0.001	0.007	0.002	–0.003
M10	–0.001	0.001	0.012	0.004	–0.004
M11	0.002	0.003	0.009	0.005	–0.001
M12	0.003	0.003	0.007	0.003	–0.001
M13	0.003	0.003	0.008	0.002	–0.002
M14	0.002	0.004	0.009	0.003	–0.001
M15	0.004	0.003	0.011	0.004	–0.002
M16	0.002	0.002	0.007	0.002	–0.001

Table B2 Component 2 characteristics

Serial	*Deviations from nominal*				
	A	*B*	*C*	*D*	*E*
M17	0.005	0.004	0.010	0.000	0.000
M18	0.002	0.003	0.007	0.001	–0.001
M19	0.002	0.003	0.007	0.001	–0.001
M20	0.004	0.004	0.010	0.002	–0.001
M21	0.002	0.003	0.009	0.005	–0.001
M22	0.003	0.003	0.008	0.002	–0.001
M23	0.002	0.004	0.009	0.003	–0.001
M24	0.002	0.002	0.007	0.002	–0.001
M25	–0.001	0.001	0.012	0.004	–0.004

Table B3 Component 3 characteristics

Serial	*Deviations from nominal*				
	A	*B*	*C*	*D*	*E*
T1	0.000	0.000	0.000	–0.006	0.000
T2	0.000	0.000	0.000	–0.002	0.000
T3	0.000	0.000	0.000	–0.002	0.000
T4	0.000	0.000	0.000	–0.001	0.000
T5	0.000	0.000	0.000	–0.003	0.000
T6	0.000	0.000	0.000	0.000	0.000
T7	0.000	0.000	0.000	0.000	0.000
T8	0.000	0.000	0.000	–0.001	0.000
T9	0.000	0.000	0.000	–0.002	0.000
T10	0.000	0.000	0.000	–0.006	0.000
T11	0.000	0.000	0.000	0.000	0.000
T12	0.000	0.000	0.000	–0.001	0.000
T13	0.000	0.000	0.000	0.000	0.000
T14	0.000	0.000	0.000	–0.004	0.000
T15	0.000	0.000	0.000	–0.002	0.000
T16	0.000	0.000	0.000	0.000	0.000
T17	0.000	0.000	0.000	–0.001	0.000
T18	0.000	0.000	0.000	–0.003	0.000
T19	0.000	0.000	0.000	–0.001	0.000
T20	0.000	0.000	0.000	0.000	0.000
T21	0.000	0.000	0.000	–0.001	0.000
T22	0.000	0.000	0.000	–0.002	0.000
T23	0.000	0.000	0.000	–0.001	0.000
T24	0.000	0.000	0.000	–0.001	0.000
T25	0.000	0.000	0.000	–0.001	0.000