

# Design decision based on dependencies among product specifications

Jian Zhang<sup>1,\*</sup>, Bingbing Li<sup>1</sup>, Alessandro Simeone<sup>1</sup>, Xingpeng Chu<sup>1</sup>, Peihua Gu<sup>2,\*</sup>

<sup>1</sup> Department of Mechanical Engineering, Shantou University, Shantou, China

<sup>2</sup> School of Mechanical Engineering, Tianjin University, Tianjin, China

**Abstract:** Changes in product specifications and their combinations can be achieved through adaptations of physical components. To better satisfy the changeable product specifications, design decisions on physical components need to be made by considering the dependencies among product specifications. In this paper, a method for dependencies quantification among product specifications is defined from the statistical point of view. Since product specifications are achieved by physical components, then the statistical dependencies among product physical components are analyzed based on specifications/components relationships. A method for design decision support based on components clustering is proposed for new product development. An example is provided to illustrate the proposed methodology.

## 1 Introduction

Engineering practices in the past decades demonstrate that the global competitiveness of manufacturers is highly related to the rapid response to changeable customers' requirements of product specifications. Changes in product specifications and their combinations are achieved through adaptations of physical components. Various types of products including customized product, reconfigurable product, upgradeable product, adaptable product, and open architecture product, have been designed with adaptabilities to satisfy the changeable requirements. A comparison of those types of products are provided in Ref. [1].

To better facilitate the development of different types of products for the satisfaction of changeable requirements, dependencies among product specifications need to be analysed for design decision of physical components by considering the specification/component relationships [2]. In Axiomatic Design, product design matrix is obtained by mapping structure domain to function domain, which indirectly describes the dependencies among functional requirements on product specifications. For a coupled design, decoupling method is needed to transform the design into uncoupled or quasi-coupled design to ensure the functional independence [3]. For a complex product or system, calculation of functional dependencies among product specifications can better facilitate the decoupling strategy planning.

As regards the design dependency calculation, most of the existing works rely on expert analyses and evaluations [4-7]. The existing expert evaluation methods mainly include Individual Judgment, Expert Meeting, Brainstorming, and Delphi Method. To better support design decision, fuzzy clustering method has also been proposed [8-10]. Although the existing methods has been widely used in recent years, due to its strong applicability and good interpretability, those methods rely on expert experiences and the acquisition of dependency among product specifications is affected

by subjective factors.

In the era of internet and big data, product data including sales and specification combinations can be easily collected. Such data represent an emerging resource for the analysis of dependencies among product specifications.

To overcome the limitations of the existing experts-based method for dependency analysis, this research aims at proposing a statistic-based correlation calculation method [11] for supporting design decision of physical components. In this work, a method for measuring dependencies among product specifications is proposed from the perspective of mathematical statistics based on collected product data. Dependency among product components is obtained by considering specifications/components relationships. Hierarchical clustering of product components is carried out for product design decision support.

## 2 Proposed method

The proposed framework is based on the following assumptions:

- (1). Big data of product sales including product sales number, combinations of product specifications etc. can be obtained through market survey.
- (2). Dependencies among product specifications can be originated from both physical components and customer preferences.
- (3). Dependencies among product specifications originated from both physical components and customers preferences are embedded in big data of product sales.

Based on the assumptions, a framework for design decision support through quantifying dependencies among product specifications is proposed and illustrated in Figure 1. In general, the proposed method includes four phases. In phase 1, product specifications and their values are obtained through market survey. In phase 2, the specifications dependencies for measuring dependency among product specifications are obtained using statistical method. In phase 3, relationships

\* Corresponding author: [jianzhang@stu.edu.cn](mailto:jianzhang@stu.edu.cn); [peihua.gu@tju.edu.cn](mailto:peihua.gu@tju.edu.cn)

between product components and specifications are obtained by understanding the product physical structures. In addition, dependency matrix of components is calculated for measuring the dependency

among product components. In phase 4, components are clustered into different group by utilizing hierarchical clustering method. Design decision support is provided based on the component clustering results.

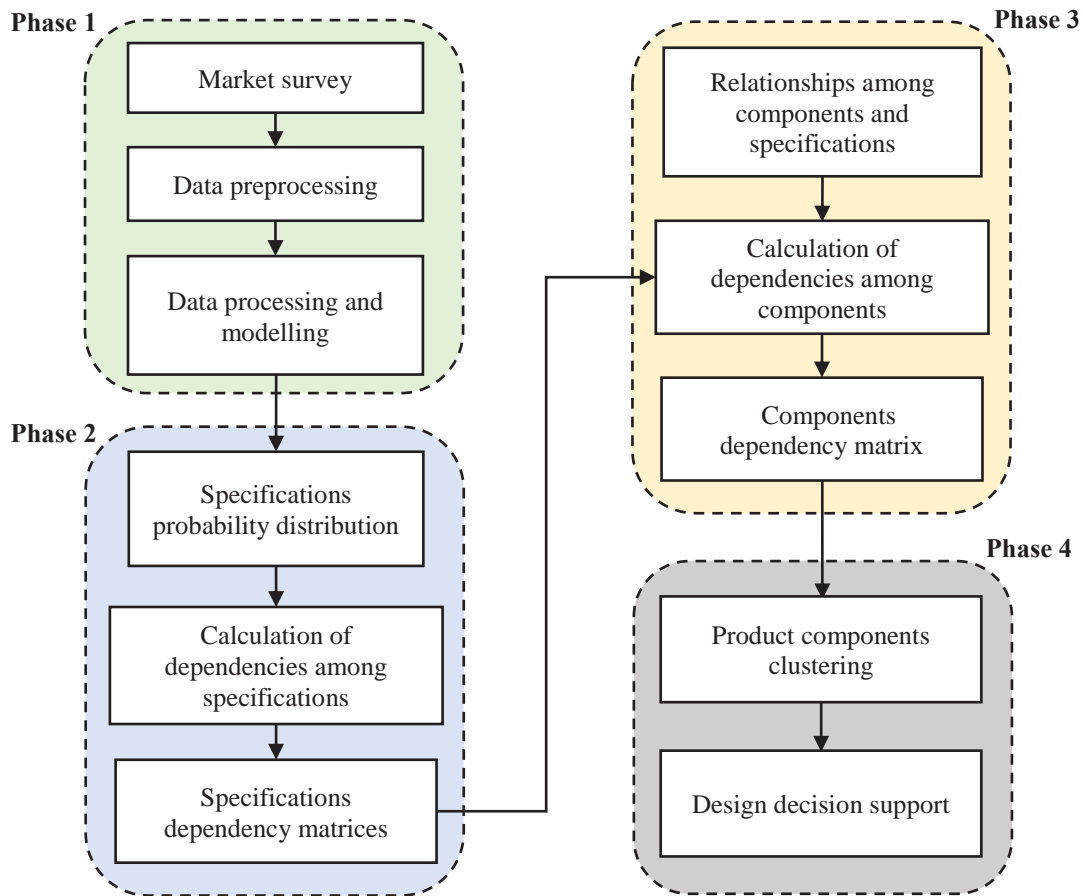


Fig. 1. Framework Flowchart.

### 2.1 Data collection and processing

The relevant data of products are collected by carrying out a market survey. The collected raw data include product sales, specifications and their values with reference to each product instance. The collected data need to be pre-processed to remove missing data and redundant information from the datasets.

Suppose  $m$  products with different specification values are obtained, the number of product sales can be arranged in array form:

$$N=[N_1, \dots, N_m] \quad (1)$$

where  $N_i$  represents the sales of the  $i$ -th product,  $i=1, 2, \dots, m$ .

All specifications are screened, and the designers need to eliminate the specifications which have the identical attributes. (e.g., price and cost, only choose one of them.) Within the collected datasets, it is assumed  $n$  specifications are considered, then the specifications  $S$  are represented by:

$$S=[S_1, \dots, S_n] \quad (2)$$

where  $S_i$  ( $i=1, \dots, n$ ) represents the  $i$ -th specification.

Specification values of each specifications are counted out. Specification values can be classified into continuous, discrete and Boolean types. For continuous variables, a discretization procedure can be applied. For example, a price ranging from 0 to 100 can be divided into four intervals:  $[0,25]$ ,  $(25,50]$ ,  $(50,75]$ ,  $(75,100]$ . Values of specification  $S_i$  are arranged as per Eq. (3):

$$V_{s_i}=[V_{i1}, \dots, V_{ik_i}] \quad (3)$$

where  $k$  indicates the number of intervals. The probability  $P_{S_{ip}}$  of the  $i$ -th specification value  $V_{ip}$  is characterized by Eq. (4):

$$\sum_{p=1}^{k_i} P_{S_{ip}}=1 \quad (4)$$

Product sales, product specifications along with values and probabilities are used in the next section for the calculation of dependencies among specifications.

### 2.2 Specifications dependencies

According to statistical method, dependencies among

variables can be calculated from the probabilities of observed sub-events of each variable.

Since values of product specifications have been discretized in Section 2.1, in this research, each product specification is considered as a discrete random variable. Every variable corresponds to a large of sub-events namely specification values. A product observed in the market is considered as a product instance. According to Eq. (1), the numbers of all observed product samples,  $N_S$ , can be calculated as:

$$N_S = \sum_{i=1}^m N_i \quad (5)$$

Suppose  $P_{s_{ip}s_{jq}}$  is the probability of  $V_{ip}$  and  $V_{jq}$  appearing in the same product, the following relationships can be obtained:

$$P_{s_{ip}s_{jq}} = \frac{f}{N_S} \quad (6)$$

where  $f$  represents the number of product samples in which  $V_{ip}$  and  $V_{jq}$  appear in the same product sample,  $i, j \in [1, n]$ ;  $p \in [1, k_i]$ ,  $q \in [1, k_j]$ .

Suppose  $E(S_i)$  and  $E(S_j)$  represent the expected value of  $S_i$  and  $S_j$ , respectively,  $E(S_i S_j)$  represent the expected value of  $S_i S_j$ . The following relationships can be obtained:

$$E(S_i S_j) = \sum_{p=1}^{k_i} \sum_{q=1}^{k_j} (V_{ip} \times V_{jq} \times P_{s_{ip}s_{jq}}) \quad (7)$$

$$E(S_i) = \sum_{p=1}^{k_i} (P_{s_{ip}} \times V_{ip}) \quad (8)$$

$$E(S_j) = \sum_{q=1}^{k_j} (P_{s_{jq}} \times V_{jq}) \quad (9)$$

where  $P_{s_{ip}}$  is the probability of the value of the  $i$ -th specification equal to  $V_{ip}$ ,  $P_{s_{jq}}$  is the probability of the value of the  $j$ -th specification equal to  $V_{jq}$ . The dependency between two product specifications  $S_i$  and  $S_j$ , i.e., the correlation  $\rho_{s_i s_j}$  can be calculated by:

$$\rho_{s_i s_j} = \frac{COV(S_i, S_j)}{\sqrt{D_{S_i}} \sqrt{D_{S_j}}} \quad (10)$$

where  $D_{S_i}$  and  $D_{S_j}$  represent the variances of  $S_i$  and  $S_j$ , respectively,  $COV(S_i, S_j)$  represents the covariance between  $S_i$  and  $S_j$ .

$$D_{S_i} = \sum_{p=1}^{k_i} [V_{ip} - E(S_i)]^2 \times P_{s_{ip}} \quad (11)$$

$$D_{S_j} = \sum_{q=1}^{k_j} [V_{jq} - E(S_j)]^2 \times P_{s_{jq}} \quad (12)$$

$$COV(S_i, S_j) = E(S_i S_j) - E(S_i)E(S_j) \quad (13)$$

Therefore, dependencies among specifications can be modelled by the matrix  $M_s$ :

$$M_s = \begin{bmatrix} 1 & \rho_{S_1 S_2} & \dots & \rho_{S_1 S_n} \\ \rho_{S_1 S_2} & 1 & \dots & \rho_{S_2 S_n} \\ \dots & \dots & \dots & \dots \\ \rho_{S_1 S_n} & \rho_{S_2 S_n} & \dots & 1 \end{bmatrix} \quad (14)$$

### 2.3 Components dependencies

To better support design decision, the degree of influence of each specification on product components need to be considered.

Based on the dependencies among specifications, the dependencies among components can be obtained according to specification/component relationships.

In this work, suppose  $t$  is the number of components considered in design, and specifications  $S_p, \dots, S_q$  are influenced by component  $C_x, x=1, \dots, t, p \neq q, p, q \in [1, n]$  Specifications  $S_u, \dots, S_v$  are influenced by component  $C_y, y=1, \dots, t, u \neq v, u, v \in [1, n]$  The dependency between components  $C_x$  and  $C_y$  can be calculated based on dependency among specifications  $S_p, \dots, S_q$  and  $S_u, \dots, S_v$ . In this work, the dependencies between components  $C_x$  and  $C_y$  are defined by the expected values of the dependency among specifications  $S_p, \dots, S_q$  and  $S_u, \dots, S_v$ . Therefore, according to Eq. (10) the dependency between components  $C_x$  and  $C_y$  can be calculated by:

$$\rho_{C_x C_y} = \frac{\sum_{j=u}^v \sum_{i=p}^q \rho_{S_i S_j}}{(q-p) \times (v-u)} \quad (15)$$

where  $x, y=1, \dots, t, p \neq q, u \neq v, p, q, u, v = 1, 2, \dots, n, n$  is the number of specifications considered in the design. The dependency between each component and itself is equal to 1. According to Eq. (15), the dependency among components can be modelled by the dependency matrix  $M_c$ , as follows:

$$M_c = \begin{bmatrix} 1 & \rho_{C_1 C_2} & \dots & \rho_{C_1 C_t} \\ \rho_{C_1 C_2} & 1 & \dots & \rho_{C_2 C_t} \\ \dots & \dots & \dots & \dots \\ \rho_{C_1 C_t} & \rho_{C_2 C_t} & \dots & 1 \end{bmatrix} \quad (16)$$

### 2.4 Components clustering and decision support

Results from the previous section leads to a matrix of product components  $M_c$ , to give out design decision support. Dependencies among product components should be find out to support the design.

Hierarchical clustering is an unsupervised learning technic which is very useful if the data has no target attribute [12]. The data is explored to find some intrinsic structures inside them [13-14]. The combination of components can be obtained according to the dependency matrices through this method. Finally, designers can modify product design according to the clustering results. Hierarchical clustering algorithm decomposes a given data set hierarchically and organizes the data into a clustering tree, which can be divided into two schemes: cohesive and split. Aggregated hierarchical clustering is a bottom-up strategy. The split hierarchical clustering is contrary to the agglomerated hierarchical clustering [12,15].

The result of hierarchical clustering is a set of components, which provides an important basis for product design reorganization. The components can be

designed as the same module when the dependencies of them are high, which is more convenient for design improvement and maintenance.

To give designers a convincing design decision support. The cluster results should be discussed to clarify the scope of application.

Clustering results can be used as a basis for various design methods such as modular design, product family design, adaptable design, and open architecture product design. Since the clustering results are obtained by calculating the product feature dependency matrix and market survey provides the source data for the matrix. This design decision support could not only take into account the preferences of customers, but also avoid the influence of individual subjective consciousness on the results of product module division.

In Axiomatic Design, functional requirements are required to be independently satisfied. The proposed method can potentially support Axiomatic Design by

evaluating the dependencies among specification clusters. Specification clusters with less functional dependency are considered as “weak point” to be decoupled in design.

### 3 Case study

Battery pack is an essential to development of new electric vehicle. Many specifications of electric vehicle are influenced/determined by battery pack. A typical structure of battery pack is shown in Figure 2. To better satisfy changeable diversified customers’ requirements of specifications, battery packs in electric vehicles need to be well designed with better adaptability. In this work, design decision support of battery pack based on dependency among product specifications was carried out to illustrate the proposed method.

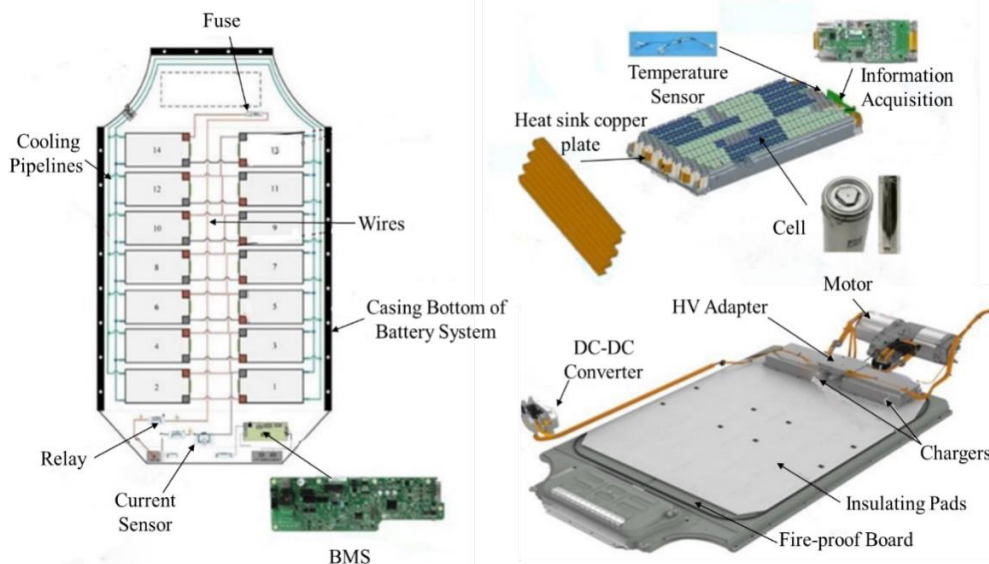


Fig. 2. A typical structure of battery pack.

#### 3.1 Data collection of battery pack

By market survey, 31 different types of battery packs for

electric vehicles are obtained. Data including sales and specification values of battery packs are collected. What needs special mention is that, only sales number for 2018 is recorded. Excerpt of the collected datasets can be seen in Table 1.

Table 1. Excerpt of the collected datasets.

ID	Sales in 2018	Specifications								
		Price (× 1000 CNY)	Electric Mileage (km)	Maximum Power (kw)	Maximum Speed (km/h)	Energy Density (Wh/kg)	Battery Warranty Time (years)	Battery Capacity (kwh)	Electricity Consumption (kwh/100km)	Model of Cell
1	65871	25	400	160	150	101.75	8	60.5	14.2	Cylindrical
2	43902	22.5	400	160	130	160.8	8	60.48	14.57	Cylindrical
3	35699	9.5	305	70	120	146.27	6	42	13.6	Cylindrical
4	31274	19.5	318	80	125	122.68	8	48	16.4	Cylindrical
5	27870	6	301	30	100	140.91	8	38	12	Cylindrical
6	15336	5	255	30	100	150	8	27	10.6	Cylindrical
7	10329	17	351	90	151	150.88	8	49	15	Cylindrical
8	8852	22	360	132	156	126.57	8	51	1606	Cylindrical
9	6508	33.5	400	90	140	125	4	82	20.5	Prismatic
10	5096	17.5	301	85	102	140.5	8	39	16	Cylindrical

### 3.2 Dependencies among specifications

Many specifications of battery packs are obtained by investigating and analysing the existing battery packs in the market. Considering that parts of the specifications have the same characteristics and the lack of data, nine of them have been sorted out for analysis and calculation. As shown in Table 2, there are the specification values corresponding to the nine specifications.

**Table 2.** Product specifications and values.

ID	Specification	Specification Values
S <sub>1</sub>	Price (×1000 CNY)	[0,0.1]; (0.1,0.2); (0.2,0.3); (0.3,0.4); (0.4, +∞)
S <sub>2</sub>	Electric Mileage (km)	[0,250]; (250,300); (300,350); (350,400); (400,+∞)
S <sub>3</sub>	Maximum Power (kw)	[0,50] ; (50,100); (100,150); (150,200); (200,+∞)
S <sub>4</sub>	Maximum Speed (km/h)	[0,100]; (100,120); (120,140); (140,160); (160,+∞)
S <sub>5</sub>	Energy Density (Wh/kg)	[0,100]; (100,120); (120,140); (140,160); (160,+∞)
S <sub>6</sub>	Fast Charging Time(h)	[0, 0.5]; (0.5,1]; (1,1.5]; (1.5,+∞); NO
S <sub>7</sub>	Battery Capacity (kwh)	[0,20]; (20,40); (40,60); (60,80); (80,+∞)
S <sub>8</sub>	Electricity Consumption of 100 KM (kwh/100km)	[0,11]; (11,14); (14,17); (17,20); (20,+∞)
S <sub>9</sub>	Model of Cell	Prismatic; Cylindrical;

Based on the case study data, pairwise dependencies among the 9 specifications are calculated according to Eqs. 5-10 and the results are reported in the 9x9 matrix  $M_s$  partially shown below for Specifications from  $S_1$  to  $S_9$ .

$M_s =$

1	0.81	0.82	0.82	0.23	0.45	0.82	0.83	0.21
0.81	1	0.81	0.81	0.10	0.78	0.78	0.67	0.11
0.82	0.81	1	0.80	0.18	0.87	0.87	0.54	0.20
0.82	0.81	0.80	1	0.45	0.77	0.77	0.72	0.05
0.23	0.10	0.18	0.45	1	0.23	0.23	0.23	0.11
0.45	0.17	0.45	0.19	0.04	1	0.37	0.03	0.06
0.82	0.78	0.97	0.77	0.23	0.37	1	0.70	0.10
0.83	0.67	0.54	0.72	0.23	0.03	0.70	1	0.44
0.21	0.11	0.20	0.05	0.11	0.06	0.10	0.44	1

### 3.3 Components dependencies

Major components considered in the design are summarized in Table 3. By analyzing the structure of the sample battery packs, the list of components is retrieved, and then the characteristics of each component are analyzed to identify the relationships between components and specifications as shown in the matrix reported in Table 4, where the zeros indicate no influence

and the ones indicate that the components affect a certain specification. As an example, component  $C_1$  results to affect specifications  $S_1, S_2,$  and  $S_8$ .

**Table 3.** Components of battery pack.

ID	Components	ID	Components
C <sub>1</sub>	Water-cooled Connector	C <sub>14</sub>	Cell
C <sub>2</sub>	Fuse	C <sub>15</sub>	Wire Guiding Device
C <sub>3</sub>	Insulating Joint	C <sub>16</sub>	Casing Bottom
C <sub>4</sub>	Wires	C <sub>17</sub>	Heat sink copper plate
C <sub>5</sub>	Casing Bottom of Battery System	C <sub>18</sub>	Interlayer
C <sub>6</sub>	BMS	C <sub>19</sub>	HV Adapter
C <sub>7</sub>	Current Sensor	C <sub>20</sub>	Motor
C <sub>8</sub>	Relay	C <sub>21</sub>	Charging Interface
C <sub>9</sub>	Cooling Pipelines	C <sub>22</sub>	Chargers
C <sub>10</sub>	Isolation Device	C <sub>23</sub>	Insulating Pads
C <sub>11</sub>	Temperature Sensor	C <sub>24</sub>	Fire-proof Board
C <sub>12</sub>	Information Acquisition	C <sub>25</sub>	DC-DC Converter
C <sub>13</sub>	Generatrix Aggregation	...	...

**Table 4.** Specifications/Components relationship matrix.

	S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>	S <sub>4</sub>	S <sub>5</sub>	S <sub>6</sub>	S <sub>7</sub>	S <sub>8</sub>	S <sub>9</sub>
C <sub>1</sub>	1	1	0	0	0	0	0	1	0
C <sub>2</sub>	1	0	0	1	0	1	0	0	0
C <sub>3</sub>	1	0	0	0	0	0	0	0	0
C <sub>4</sub>	1	0	0	0	0	0	0	0	0
C <sub>5</sub>	1	0	0	0	0	0	0	0	0
C <sub>6</sub>	1	1	1	1	0	1	0	1	0
C <sub>7</sub>	1	1	1	1	0	1	0	0	0
C <sub>8</sub>	1	1	1	1	0	1	0	0	0
C <sub>9</sub>	1	1	0	0	0	0	0	1	0
C <sub>10</sub>	1	0	0	0	0	0	0	0	0
C <sub>11</sub>	1	1	0	1	0	0	0	0	0
C <sub>12</sub>	1	1	0	1	0	0	0	0	0
C <sub>13</sub>	1	0	0	0	0	0	0	0	0
C <sub>14</sub>	1	1	1	1	1	1	1	1	1
C <sub>15</sub>	1	0	0	0	0	0	0	0	0
C <sub>16</sub>	1	0	0	0	0	0	0	0	0
C <sub>17</sub>	1	1	0	0	0	0	0	0	0
C <sub>18</sub>	1	0	0	0	0	0	0	0	0
C <sub>19</sub>	1	0	0	0	0	1	0	0	0
C <sub>20</sub>	1	1	1	1	0	0	0	1	0
C <sub>21</sub>	1	0	0	0	0	1	0	0	0
C <sub>22</sub>	1	0	0	0	0	1	0	0	0
C <sub>23</sub>	1	0	0	0	0	0	0	0	0
C <sub>24</sub>	1	0	0	0	0	0	0	0	0
C <sub>25</sub>	1	0	0	0	0	1	0	0	0

For the 25 components identified in this work, the dependency matrix of components  $M_c$  is computed according to the Eq. (15) and partially reported in the matrix below:



$M_c =$

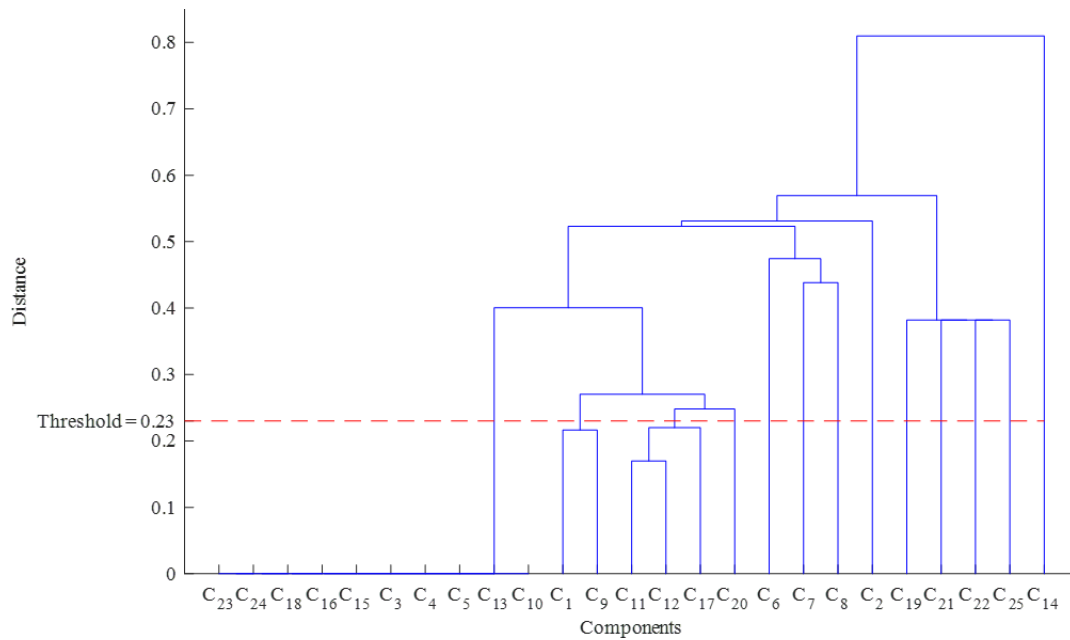
1	0.63	0.88	0.88	0.88	0.71	0.69	0.69	0.85	0.88	0.83	0.83	0.88	0.61	0.88	0.88	0.85	0.88	0.55	0.81	0.55	0.55	0.88	0.88	0.55
0.63	1	0.76	0.76	0.76	0.63	0.65	0.65	0.63	0.76	0.67	0.67	0.76	0.53	0.76	0.76	0.68	0.76	0.65	0.65	0.65	0.65	0.76	0.76	0.65
0.88	0.76	1	1	1	0.79	0.78	0.78	0.88	1	0.88	0.88	1	0.67	1	1	0.91	1	0.73	0.86	0.73	0.73	1	1	0.73
0.88	0.76	1	1	1	0.79	0.78	0.78	0.88	1	0.88	0.88	1	0.67	1	1	0.91	1	0.73	0.86	0.73	0.73	1	1	0.73
0.88	0.76	1	1	1	0.79	0.78	0.78	0.88	1	0.88	0.88	1	0.67	1	1	0.91	1	0.73	0.86	0.73	0.73	1	1	0.73
0.71	0.63	0.79	0.79	0.79	1	0.67	0.67	0.71	0.79	0.72	0.72	0.79	0.56	0.79	0.79	0.75	0.79	0.59	0.72	0.59	0.59	0.79	0.79	0.59
0.69	0.65	0.78	0.78	0.78	0.67	1	0.69	0.69	0.78	0.74	0.74	0.78	0.56	0.79	0.79	0.75	0.78	0.62	0.71	0.62	0.62	0.78	0.78	0.62
0.69	0.65	0.78	0.78	0.78	0.67	0.69	1	0.69	0.78	0.74	0.74	0.78	0.56	0.79	0.79	0.75	0.78	0.62	0.71	0.62	0.62	0.78	0.78	0.62
0.88	0.63	0.88	0.88	0.88	0.71	0.69	0.69	1	0.88	0.80	0.80	0.88	0.61	0.88	0.88	0.85	0.85	0.55	0.81	0.55	0.55	0.88	0.88	0.55
0.85	0.76	1	1	1	0.79	0.78	0.78	0.88	1	0.88	0.88	1	0.67	1	1	0.91	1	0.73	0.86	0.73	0.73	1	1	0.73
0.83	0.67	0.88	0.88	0.88	0.72	0.74	0.74	0.80	0.88	1	0.88	0.88	0.62	0.88	0.88	0.88	0.88	0.57	0.84	0.57	0.57	0.88	0.88	0.57
0.83	0.67	0.88	0.88	0.88	0.72	0.74	0.74	0.80	0.88	0.88	1	0.88	0.62	0.88	0.88	0.88	0.8	0.57	0.84	0.57	0.57	0.88	0.88	0.57
0.88	0.76	1	1	1	0.79	0.78	0.78	0.80	1	0.88	0.88	1	0.67	1	1	0.91	1	0.73	0.86	0.73	0.73	1	1	0.73
0.61	0.53	0.67	0.67	0.67	0.56	0.56	0.56	0.61	0.67	0.62	0.62	0.67	1	0.67	0.67	0.63	0.67	0.49	0.62	0.49	0.49	0.67	0.67	0.49
0.88	0.76	1	1	1	0.79	0.78	0.78	0.88	1	0.88	0.88	1	0.67	1	1	0.91	1	0.73	0.86	0.73	0.73	1	1	0.73
0.88	0.76	1	1	1	0.79	0.78	0.78	0.88	1	0.88	0.88	1	0.67	1	1	0.91	1	0.73	0.86	0.73	0.73	1	1	0.73
0.85	0.68	0.91	0.91	0.91	0.75	0.75	0.75	0.85	0.91	0.88	0.88	0.91	0.63	0.91	0.91	1	0.91	0.61	0.84	0.61	0.61	0.91	0.91	0.61
0.88	0.76	1	1	1	0.79	0.78	0.78	0.88	1	0.88	0.88	1	0.67	1	1	0.91	1	0.73	0.86	0.73	0.73	1	1	0.73
0.55	0.65	0.73	0.73	0.73	0.59	0.62	0.62	0.55	0.73	0.57	0.57	0.73	0.49	0.73	0.73	0.61	0.73	1	0.56	0.73	0.73	0.73	0.73	0.73
0.81	0.65	0.86	0.86	0.86	0.72	0.71	0.71	0.81	0.86	0.84	0.84	0.86	0.62	0.86	0.86	0.84	0.86	0.56	1	0.56	0.56	0.86	0.86	0.56
0.55	0.65	0.73	0.73	0.73	0.59	0.62	0.62	0.55	0.73	0.57	0.57	0.73	0.49	0.73	0.73	0.61	0.73	0.73	0.56	1	0.73	0.73	0.73	0.73
0.55	0.65	0.73	0.73	0.73	0.59	0.62	0.62	0.55	0.73	0.57	0.57	0.73	0.49	0.73	0.73	0.61	0.73	0.73	0.56	0.73	1	0.73	0.73	0.73
0.88	0.76	1	1	1	0.79	0.78	0.78	0.88	1	0.88	0.88	1	0.67	1	1	0.91	1	0.73	0.86	0.73	0.73	1	1	0.73
0.88	0.76	1	1	1	0.79	0.78	0.78	0.88	1	0.88	0.88	1	0.67	1	1	0.91	1	0.73	0.86	0.73	0.73	1	1	0.73
0.55	0.65	0.73	0.73	0.73	0.59	0.62	0.62	0.55	0.73	0.57	0.57	0.73	0.49	0.73	0.73	0.61	0.73	0.73	0.56	0.73	0.73	0.73	0.73	1

### 3.4 Component clustering and discussion on design decision

The modular design [15] of battery pack provides an important basis for battery pack adaptations. The dependency matrix of components  $M_c$  is used as the basis for hierarchical division. To fit the algorithm, every item in the matrix is replaced

by the difference between the original value and 1.

From Fig.3, different combinations of components can be obtained when different thresholds are selected. Considering the existing product structure, when the threshold is set to 0.23, the clustering results, as shown in Table 5, can meet the physical constraints of product components to the greatest extent.



**Fig. 3.** Results of Components Clustering.

**Table 5.** Battery pack components clustering.

Clusters	Components
Cluster 1	C <sub>1</sub> , C <sub>9</sub>
Cluster 2	C <sub>11</sub> C <sub>12</sub> C <sub>17</sub>
Cluster 3	C <sub>20</sub>
Cluster 4	C <sub>6</sub>
Cluster 5	C <sub>14</sub>
Cluster 6	C <sub>7</sub>
Cluster 7	C <sub>8</sub>
Cluster 8	C <sub>2</sub>
Cluster 9	C <sub>19</sub>
Cluster 10	C <sub>21</sub>
Cluster 11	C <sub>22</sub>
Cluster 12	C <sub>25</sub>
Cluster 13	C <sub>23</sub> C <sub>24</sub> C <sub>18</sub> C <sub>16</sub> C <sub>15</sub> C <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>13</sub> C <sub>10</sub>

From Table 5, components within Cluster 1, Cluster 2 and Cluster 13 are suggested to be designed as modules because relative high dependencies among components can be

observed in these clusters.

In Cluster 1, C<sub>1</sub> and C<sub>9</sub> can be designed as a modular, which has been already realized in current designs. This shows the rationality of the existing design of battery pack.

In Cluster 2, C<sub>11</sub>, C<sub>12</sub> and C<sub>17</sub> are suggested to be designed as a module. However, in the current design of battery pack, only C<sub>11</sub> and C<sub>12</sub> are designed as a modular, C<sub>17</sub> has not yet been considered in the module because the dependencies among specifications have not been well investigated. To better support adaptations of battery packs, design modification is suggested to form C<sub>11</sub>, C<sub>12</sub> and C<sub>17</sub> as a module for the new battery pack development.

In Cluster 13, C<sub>16</sub> and C<sub>24</sub> are apparently physically unrelated. They are clustered into the same cluster is due to the simplification of the specification data collection in the case study. From the dendrogram as shown in Fig. 3, components in Cluster 13 are all zero, this is because there is a lack of specifications to distinguish components C<sub>16</sub> and C<sub>24</sub>. In real design, sufficient specification data need to be collected to improve the accuracy of the proposed method.

## 4 Conclusions

Big data of product sales is an essential resource for the analysis of dependency among product specifications. In this work, a statistic-based method is proposed for measuring dependency among product specifications through the analysis of collected data of product sales. By considering the components/specifications relationships, a matrix for measuring the dependency among product components is obtained for Component cluster analysis. Clustering results of product components are used for support design decision. A case study of battery pack in electric vehicle is used to illustrate the proposed method.

Ongoing research activities in this scope include (1) integrations of the dependency analysis method for facilitating adaptable design, product family design, product platform design and open architecture product design, etc. [17-19] (2) investigation of the evolution trends of product specifications in the market by using historical data of product sales.

## Acknowledgment

The authors wish to thank the National Key R&D Program of China (No: 2018YFB1701701), the National Natural Science Foundation of China (Nos: 51375287, 51505269), the Sailing Talent Program, Pearl River Scholar Program, and the Leading Talent Project of Guangdong Province, China for providing financial supports to this research.

## References

- [1] J. Zhang, G. Xue, H.L. Du, A. Garg, Q.J. Peng, P.H. Gu, Enhancing interface adaptability of open architecture products. *Res.Eng. Des.* **28**, 545-560 (2017)
- [2] J.X. Liu, J.Y. Li, C.F. Liu, Y.F. Discover Dependencies from Data—A Review. *IEEE Trans. Knowl. Data Eng.* **24**, 251-264 (2012)
- [3] R. Sheikh, M. Abbasi, A.A. Talaei, M. Tahmasbi, Dynamic Axiomatic Design: Applying the Independence Axiom in the Design of Social Systems. *Procedia CIRP.* **34**, 125-130 (2015)
- [4] Z. Fatemeh, A method for quantitative evaluation of expert systems. *Eur. J. Oper. Res.* **48**, 136-147 (1990)
- [5] C.H. Yang, J. Wei, Z. Xie, F. Li, Research on evaluation method of maintainability design based on QFD and D-S theory. *J. Eng. Des.* **16**, 93-97 (2009)
- [6] M. Roubens, Preference relations an actions and criteria in multicriteria decision making. *Eur. J. Oper. Res.* **10**, 51-55 (1982)
- [7] H. Pastijn, J. Leysen, Constructing an outranking relation with ORESTE. *Math.*

- Comput. Model.* **12**, 1255-1268 (1989)
- [8] L. Li, J.Z. Zhou, X.L. An, Y.H. Li, Q. Liu, Improved Fuzzy Clustering Method Based on Entropy Coefficient and Its Application. *International Symposium on Neural Networks.* **5264**, 11-20 (2008)
- [9] R.M. Rodríguez, B. Bedregal, H. Bustince, Y.C. Dong, B. Farhadinia, C. Kahraman, L. Martínez, V. Torra, Y.J. Xu, Z.S. Xu, F. Herrera, A position and perspective analysis of hesitant fuzzy sets on information fusion in decision making. *Inform. Fusion.* **29**, 89-97 (2016)
- [10] H.X. Li, Multifactorial Fuzzy Sets and Multifactorial Degree of Nearness. *Fuzzy Set. Syst.* **19**, 291-297 (1986)
- [11] C. Richard, Aster, B. Brian, H. Clifford Thurber, Appendix B Review of probability and statistics. *International Geophysics.* **11**, 251-272 (2005)
- [12] S.C. Johnson, Hierarchical clustering schemes. *Psychometrika.* **32**, 241-254 (1967)
- [13] H.B. Barlow, Unsupervised Learning. *Neural Comput.* **1**, 295-311 (1989)
- [14] K.A. Ciecierski, T. Mandat, Unsupervised Machine Learning in Classification of Neurobiological Data. *Intelligent Methods and Big Data in Industrial Applications.* **40**, 203-212 (2019)
- [15] Y.Z. Sun, J.W. Han, P. Zhao, Z. Yin, H. Cheng, T. Wu, Rankclus: Integrating clustering with ranking for heterogenous information network analysis. *Acm Sigkdd International Conference on Knowledge Discovery & Data Mining ACM (EDBT, Saint Petersburg, 2009)*
- [16] E. Spooner, A.C. Williamson, G. Catto, Modular design of permanent-magnet generators for wind turbines. *IEEE P - Elect. Pow. Appl.* **143**, 388-395 (1996)
- [17] P.H. Gu, M. Hashemian, A.Y.C. Nee, Adaptable Design. *CIRP Ann. - Manuf. Tech.* **53**, 539-557. (2004)
- [18] Y. Koren, S.J. Hu, P.H. Gu, M. Shpitalni, Open-architecture products. *CIRP Ann. - Manuf. Tech.* **62**, 719-729 (2013)
- [19] Q.J. Peng, Y.H. Liu, P.H. Gu, Development of an Open-architecture Electric Vehicle Using Adaptable Design. *Advances in Sustainable and Competitive Manufacturing Systems, Lecture Notes in Mechanical Engineering.* 79-90 (2013)