

# A gearing forest automaton to automate a gearing system

Huazeng Zhao (✉ [zhaohuazeng@126.com](mailto:zhaohuazeng@126.com))

Shandong College of Information Technology

Zuxiao Yang

Weifang University

Qiushi Zhao

Virginia Tech: Virginia Polytechnic Institute and State University

---

## Original Article

**Keywords:** Gearing system design, Gearing tree, Gearing forest, Gearing forest automaton

**Posted Date:** November 16th, 2020

**DOI:** <https://doi.org/10.21203/rs.3.rs-108133/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

# A gearing forest automaton to automate a gearing system

Huazeng Zhao<sup>1,\*</sup>, Zuxiao Yang<sup>2,\*</sup>, Qiushi Zhao<sup>3</sup>

## Abstract

A gearing system design completely depends on the designer's skills and experiences because there is no theory or method to standardize the design process. In this paper, a gearing forest automaton is presented to implement automatic formation of design schemes from the original data without any human intervention. Firstly, a gearing tree is defined to describe a gearing system, and a two-shaft connection by  $M$ -transmission shafts ( $M \geq 0$ ) and an  $N$ -shaft connection by a transmission shaft ( $N \geq 3$ ), which can iteratively be combined into any gearing system, are denoted with data structure and automatically achieved. Secondly, the connection methods are both the input alphabet and transition functions of the gearing forest automaton. The start state comprises the original data and the accept state outputs the design schemes. Therefore, an entire design process becomes the construction and running of the automaton. The automaton is optimized to increase efficiency. Finally, according to this automaton, a computer program has been developed, and it is validated through three examples in which design quality is significantly improved.

**Keywords:** Gearing system design, Gearing tree, Gearing forest, Gearing forest automaton

## 1 Introduction

A gearing system diverts power from a single input shaft to multiple output shafts, as shown in Figure 1. It is a key part of many machines, such as a modular machine tool that is the most popular in mass-machining box parts of vehicles as it can increase production efficiency by multiple times or even by tens of times compared to a general machine tool. A major task in the design phase is when a designer connects all output shafts to an input shaft by adding new transmission shafts. The design method changes almost nothing from past [1] to present [2]:

1. Artificially specify a local layout by experience.
2. Compute the parameters, such as gears and shafts.
3. Draw the connection.
4. Improve the connection by trial and error.
5. If success, take the connection regard as a shaft.
6. Repeat the above steps until the connection becomes a gearing system.

Gear teeth must be integers and lie within a practical range, while their combinations must satisfy any gear ratio constraints and any center distance constraints of a gear system. Designing a gear system is basically a straightforward but highly repetitive exercise which has no unique solution. It is extremely difficult to ensure that every fuzzy experiential decision is correct, even optimal.

Therefore, often the transmitting component load increases too large and the gear ratio tolerance decreases too much as a gearing system design proceeds. Consequentially, some designs have to be redone because the strength is insufficient, the gear ratio requirements are not met, or interferences occur, which leads to low efficiency and unreasonable design schemes. Experience and familiarity are much more important qualities than mathematical or drafting skills. Over the past 50 years, it has been the significant object of research to improve such a situation by the design schemes which are automatically generated by a computer program.

### 1.1 development in automating a local gearing system design

To address a general algorithm, it is necessary to classify the gearing systems into certain types. Various shaft connections are reduced to nine types early in 1973 [3], and the reduction is improved [4],[5]. In addition, the special computer programs that automatically calculate the shaft coordinates according to the types after the gears are determined, have been developed. A type setting up the intermediate shaft gears as single gears and double gears, has been focused [6]. Except of the shaft coordinate calculation, an automated gear design is taken into consideration independently [7] or in a way in which the gear is connected to an automated entity by a decision structure [8]. Shafts and gears can be automatically designed and optimized after the connection forms are specified [9], [10]. After the local parameters are input, a program can check interferences [11], gear ratios and strengths [12], [13], and can produce a segmented graphical solution that is able to be further modified [14]. In recent years, an intelligent design has become a

\*Correspondence: zhaohuazeng@126.com

<sup>1</sup> Department of Information, Shandong College of Information Technology, Weifang 261061, China.

Full list of author information is available at the end of the article

research hotspot of some Chinese graduate students [15-17]. As assembly-level geometric errors such as backlash, center distance errors and shaft misalignments may adversely affect the operation of a gearing system, the tolerance is analyzed using static analogy [18]. However, a local design, a multiple solution structure, is not able to be automated when it is not described with a data structure.

Some special gearing systems have been automatically designed, such as a coaxial, input/output shaft, double speed reduction gear system [19], a manual gearing system of a tractor [20], and simple output gearing systems [21], [22]. However, it is not possible to extend the methods used in designing the invariable structure forms to a general case. A design process is automated by integrating the dimensional design and the configuration design processes in a formalized algorithm [23]. But it is impossible to formalize a complicated gearing system.

It is worth noting that design optimization is a challenge. Although genetic algorithms [24], a graph theory [25], and object parameter methods [26] have been successfully applied to optimize local designs, some constraints, such as interferences and insufficient strengths, which are seen everywhere, don't be given enough attentions. In addition, a local solution, sometimes, is optimal when viewed within a local perspective, but is a disaster from a global perspective because of interferences, gear ratio constraints, and etc.

### **1.2 Development in automating a gearing system design from local to global**

Early in 1970s, it is proposed that a gearing system is described with a small gearing tree based on graph theory [27] and is established with gearing tree graphics from the tips [28]. A key item specializing in the theories and methods for the computer aided design of a gearing system in the Chinese machinery industry, takes 100 person-years, and develops a program that processes the production technological parameters into design data, draws the design, and checks the strength and the interference [29], [30]. The item is known as the first step to an automatic design [31]. But persistent research seems to have been thrown.

To avoid the empirical decision-making of local layouts, a model by which the layouts are directly obtained, is investigated. An expansion method that finds positions of the packing items by placing expansion and repulsion operations among the packing items, is proposed [32], [33]. The interference measurement between different meshing gears is regarded as their index of dissimilarity, and then a cluster analysis algorithm is employed [34]. However, the two models all are too rough. The authors

consider these works are a contribution to the progress of this field and not suit for practicing in engineering practices.

### **1.3 Methods used in designing automated systems**

A gearing system is a topological structure. Research on an automatic structural design can date back to early 1980 when a computer software which automatically designs steel pitched-roof portal frame buildings was developed using the methods used in manual design [35]. Various technologies for carrying out inferences on an automatic gearing design have been widely studied to achieve a variety of automatic structural design systems, such as a CAD-based methodology in the automated modular fixture design system [36], a suitable CAD modeling approach of the compound die assembly in an automated design system for compound washer dies [37], a 3D geometric modeling method by means of the API function of the CAD system in the automated design system of model sets [38], and a building information modeling method focused on pre-engineered building as a case study in the automated modelling system to facilitate integrated structural design and wind engineering analysis [39].

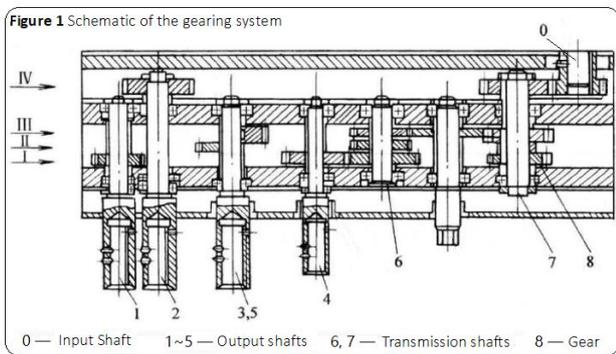
Early in 1994, a cellular automaton (CA) is first presented to automate the structural design since it can simulate the behavior of dynamic systems and can generate a clear framed structure for a mechanical condition [40]. Using the concept of a CA, a design domain is divided into small square cells and then the thicknesses of the individual cells are taken as the design variables [41]. In-plane response of composite laminae has been improved within the context of structural design using CA's iterative numerical techniques that use local rules to update both field and design variables to satisfy equilibrium and optimality conditions [42]. A CA tries to be translated into a specific design tool from a generic system in architectural design [43].

However, although the above researches focused on the problem-solving technologies for an automated structural design, a situation in which adding new parameters are prerequisite when an old state becomes into a new state, is ignored. Although an approach can automatically add new features to the model and subsequently allow the use of the newly added parameters, along with the existing parameters, the new parameters are wholly dependent on the existing ones [44].

To the author knowledge, all efforts to date have not resulted in a determination of methods to automate the entire design process of the gearing system.

This paper focuses on automating an entire design

process by the automation of a local design and extending the automation from local to global using a gearing forest automaton. Firstly, local shaft connections are classified into two types that are a two-shaft connection by M-transmission shafts ( $M \geq 0$ ) and an N-shaft connection by a transmission shaft ( $N \geq 3$ ), and the connection solutions are automatically generated as a gearing tree is defined to describe a gear system. Secondly, a gearing forest automaton is devised to iteratively realize an automatic design from local to global, and optimized to improve efficiency. Finally, according to this automaton, we have developed a program that automatically creates design schemes for a gearing system without human intervention as soon as the original data are inputted.



## 2 A two-shaft connection by M-transmission shafts ( $M \geq 0$ ) and an N-shaft connection by a transmission shaft ( $N \geq 3$ )

A gearing system can be achieved by recursively using the two connections that are a two-shaft connection by M-transmission shafts ( $M \geq 0$ ) and an N-shaft connection by a transmission shaft ( $N \geq 3$ ). In this section, we review the two connections. Note that an output shaft does not drive other shafts.

### 2.1 A two-shaft connection by M-transmission shafts ( $M \geq 0$ )

A two-shaft connection by M-transmission shafts is one in which two local designs are assembled into a new local design by adding M intermediate transmission shafts, as shown in Figure 2. Of course, it is possible that there is only a single shaft in a local design.

The blue, green, gold, and red circles indicate the gears at columns I, II, III, and IV, respectively, the crosses denote the positions of the shafts, and the Arabic numerals are the shaft sequence numbers. Note that the input shaft gear is at the column IV, as shown in Figure 1. For narrative convenience, the Arabic numeral 0 especially indicates an input shaft.

In Figure 2(a), two local designs respectively comprise shafts 1-7, 12, 13 and shafts 8-10, 14. Figure 2(b) is a new

assembled design, where shaft 15 is a new added intermediate transmission shaft. Shafts 1-10 are output shafts and others are transmission shafts. There is not an input shaft. Shafts 11, 14, and 15 are connective shafts. Three red circles are connective gears.

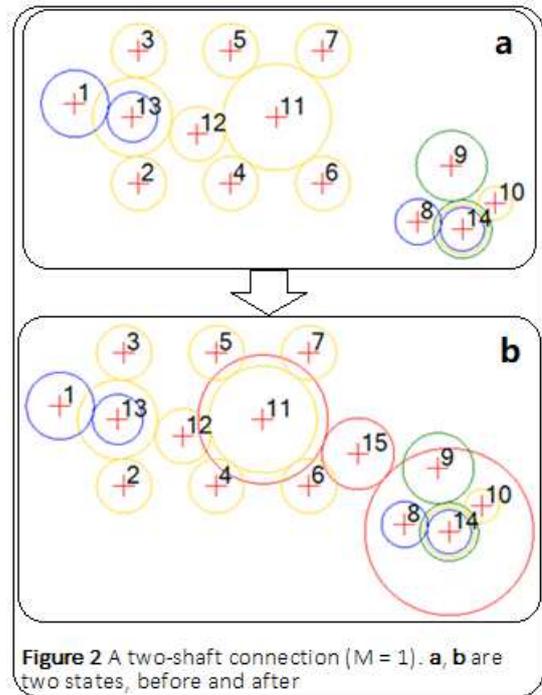


Figure 2 A two-shaft connection ( $M = 1$ ). a, b are two states, before and after

### 2.2 An N-shaft connection by one transmission shaft ( $N \geq 3$ )

An N-shaft connection by one transmission shaft is one in which N local designs are assembled into a new local design by adding one transmission shaft. In Figure 3, three shafts 12-14, respectively from three local designs, connect together by the new shaft 15, where shaft 0 is an input shaft, shafts 1-11 are output shafts, and shafts 12-15 are transmission shafts.

## 3 Automatic generation of a local design

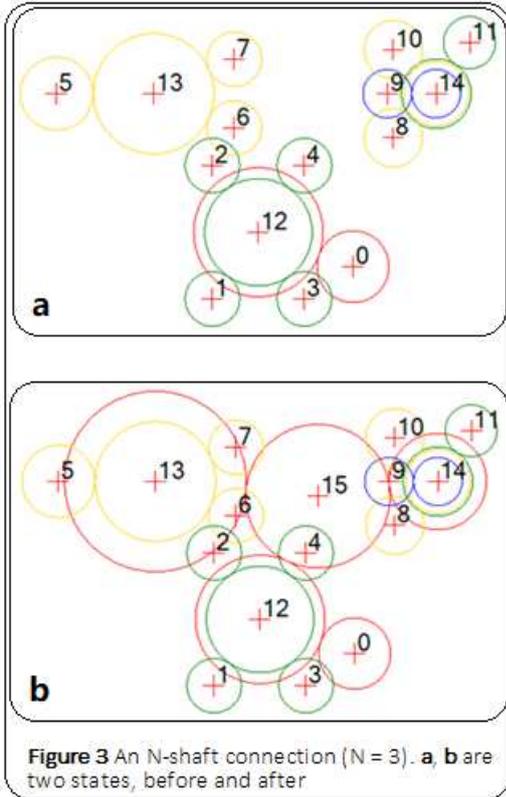
It is necessary that a gearing system is represented in a data structure to automate the design. It is abstracted as a tree, which is called a gearing tree.

**Definition 3.1** Assume that  $S_0$ ,  $S_1$ , and  $S_2$ , respectively, denote the sets of input shafts, output shafts and transmission shafts. A gearing tree is inductively defined as follows:

- (1) Every  $t \in S_0 \cup S_1$  is a gearing tree.
- (2) If  $t_i (i = 0, 1, \dots, L)$  is a gearing tree and the root of  $t_0$  belongs to  $S_0$ ,  $t(t_1, t_2, \dots, t_L)$  ( $t \in S_0 \cup S_2$ ) and  $t_0(t_1, t_2, \dots, t_{L-1})$  ( $t \in S_1 \cup S_2$ ) is a gearing tree when the kinematic and dynamic requirements are met.

A shaft, a node of a gearing tree, is defined in Table 1.

From Definition 3.1, an input shaft is always a root, an output shaft is always a leaf, and a transmission shaft that needs to be specially designed cannot be a leaf. By transmission shaft nodes, power diverts from a parent to its children, and conversely, load reacts from children to their parent. The design object is to construct a gearing tree in which the root is an input shaft, and the leaves contain all output shafts.



**Table 1** Definition of a gearing shaft

**Algorithm 1** Defining a shaft

```

class Shaft
{
    Define a Shaft pointer variable to point to the father of the Shaft
    Define a shaft variable Shaft that contains a shaft and 5 bushings
    Define a gear variable Gear [1...K], where Gear[i] ( $i = 1, \dots, K$ )
        corresponds to columns I, II, ..., K
    Define a bearing variable Bearing
    Define a construction function Shaft (p, s, g, b), where p is a Shaft
        pointer, s is a Shaft, g is a Gear, and b is a Bearing
    Define a function to judge whether the Shaft is an output shaft
    Define a function to judge whether the Shaft is an input shaft
    Define a function to check the strength
    Define a function to check the speed
    Define a function to check the interference
}
    
```

**3.1 A data structure of a two-shaft connection by M-transmission shafts ( $M \geq 0$ )**

A two-shaft connection by M-transmission shafts ( $M \geq 0$ )

has been discussed in Section 2.1. The connective method can be denoted as follows:

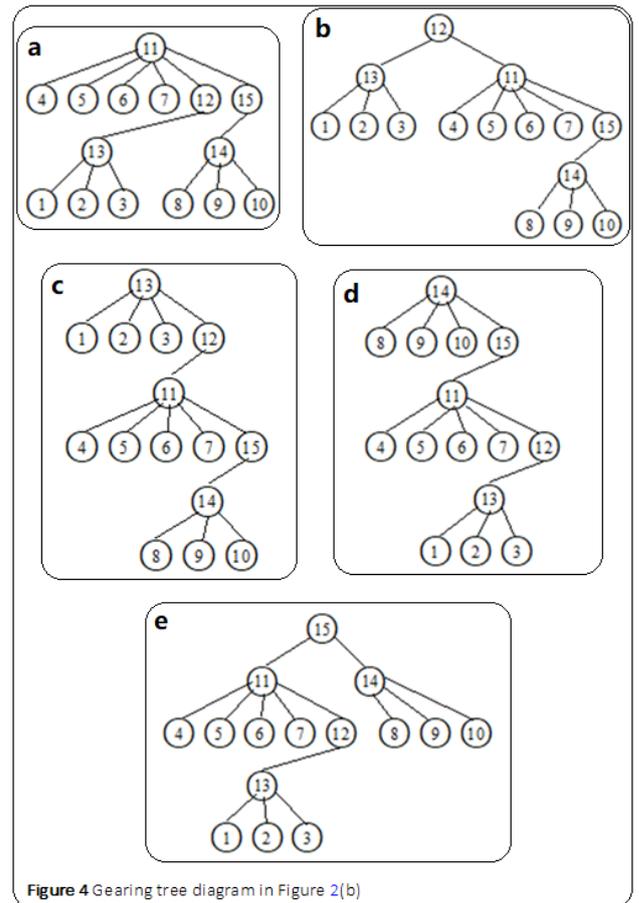
$$F(M): \begin{cases} (t_0, t_1) \rightarrow t_0(t_2, \dots, t_{M+1}, t_1) \\ (t_1, t_2) \rightarrow t_0(\dots, t_1, \dots, t_{M+2}, \dots, t_2, \dots) \\ (t_1, t_2) \rightarrow t_k(\dots, t_1, \dots, t_{M+2}, \dots, t_2, \dots) \end{cases} \quad (1)$$

where  $k \in [1, M + 2]$ ,  $F(M)$  is the connection method and  $t_i$  ( $i = 0, 1, \dots, M + 2$ ) is a gearing tree. If  $t_0$  is contained, two gearing trees are assembled into one new gearing tree. Otherwise, new trees have  $K_t$  types because every transmission shaft may be a root:

$$K_t = \sum_{i=1}^2 TNodes(t_i) + M \quad (2)$$

where  $TNodes(t_i)$  is the number of transmission shafts in gearing tree  $t_i$ .

**Example 3.1** Consider the data structure in Figure 2(b). From (2), new gearing trees are able to have five types, as shown in Figure 4.



**3.2 An N-gearing tree connection by one transmission shaft ( $N \geq 3$ )**

An N-gearing tree connection by one transmission shaft ( $N \geq 3$ ) can be denoted as follows:

$$G(N): \begin{cases} (t_0, t_1, \dots, t_{N-1}) \rightarrow t_0 (\dots t_N (t_1, \dots, t_{N-1}, \dots)) \\ (t_1, t_2, \dots, t_N) \rightarrow t_0 (\dots t_{N+1}, t_1, \dots, t_N, \dots) \\ (t_1, t_2, \dots, t_N) \rightarrow \begin{cases} t_{N+1} (t_1, t_2, \dots, t_N) \\ t_k (\dots t_{N+1}, t_1, \dots, t_N, \dots) \end{cases} \end{cases} \quad (3)$$

where  $G(N)$  is the connection method,  $t_i$  ( $i = 0, 1, \dots, N + 1$ ) is a gearing tree and the root is an input shaft when and only when  $i = 0$ . If gearing tree  $t_0$  is contained, new gearing tree has only one type, otherwise, has  $K_t$  types because every transmission shaft may be a root:

$$K_t = \sum_{i=1}^N TNodes(t_i) + 1 \quad (4)$$

where  $TNodes(t_i)$  is the number of transmission shafts in gearing tree  $t_i$ .

**Example 3.2** Consider the data structure in Figure 3(b). Because an input shaft 0 is contained, new gearing tree has only one, as shown in Figure 5.

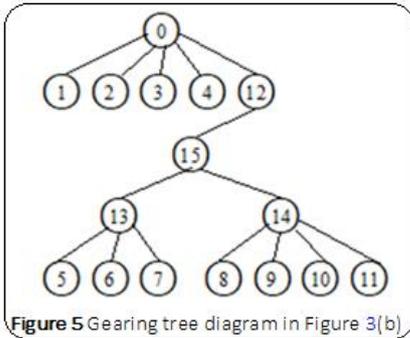


Figure 5 Gearing tree diagram in Figure 3(b)

Every  $F(i)$  ( $i = 1, 2, \dots$ ) or  $G(i)$  ( $i = 1, 2, \dots$ ) responds a local layout. Connective gears can be obtained by available gear traversal. Thus, new transmission shafts can be automatically generated. Of course, the traversals of the teeth and modules of gears and the determinations of the positions of new connection shafts generate a lot of data. To increase efficiency, some data have to be removed. In other hand, the connective methods also lead to as a rise in the transmitting component loads and a decline in the gear ratio tolerances as the conventional.

#### 4 Automation of the design process

A gearing system design is an iterative process from local to global. We describe the process using a gearing forest automaton while the state data are denoted with a gearing forest that is defined as follows.

**Definition 4.1** Assume that  $t_i$  ( $i = 1, 2, \dots, L$ ) is a

gearing tree. A gearing forest is a set of  $t_1, t_2, \dots, t_L$  if there is no interference in the set. It is denoted by

$$(t_1, t_2, \dots, t_L).$$

A gearing forest is a data structure that reflects what degree the design has proceeded to.

**Definition 4.2** A gearing forest automaton is a 5-tuple:

$$A = (Q, \Sigma, \delta, Q_0, Q_K) \quad (5)$$

where  $K$  is the number of output shafts,  $Q$  is a set of  $K + 1$  states,  $\Sigma = \{F(0), F(1), \dots, F(M), G(3), G(4), \dots, G(N)\}$  (Section 3) is the input alphabet,  $\delta = Q \times \Sigma \rightarrow 2^{Q^*}$  is a set of transition functions,  $Q_0 = \{(t_0, t_1, \dots, t_K)\} \in Q$  ( $t_0$  is an input shaft and  $t_i$  ( $i = 1, 2, \dots, K$ ) is the output shaft) is the start state,  $Q_K \in Q$  is the accept state, and  $Q_i \in Q$  ( $i = 0, 1, \dots, K$ ) is a set of gearing forests that all have  $K + 1 - i$  gearing trees.

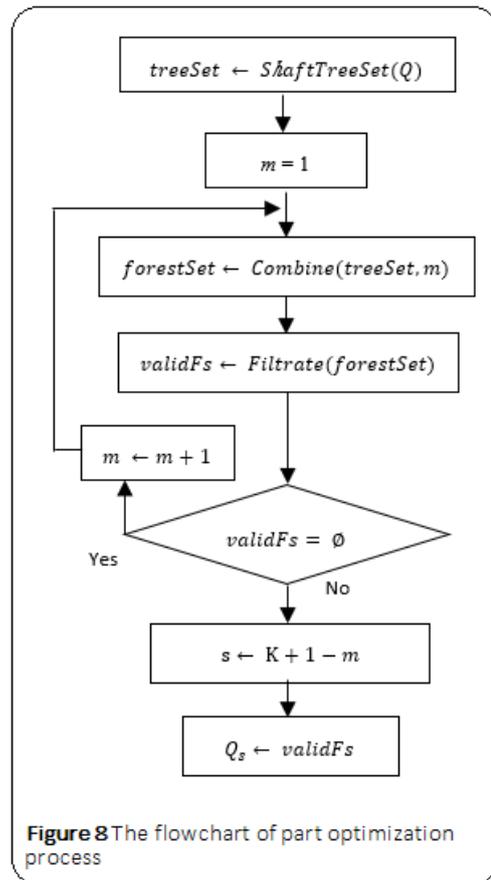
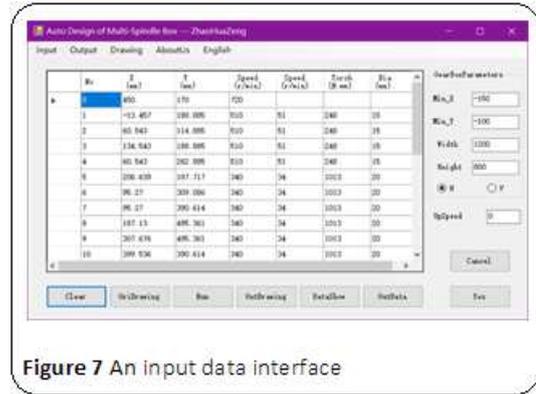
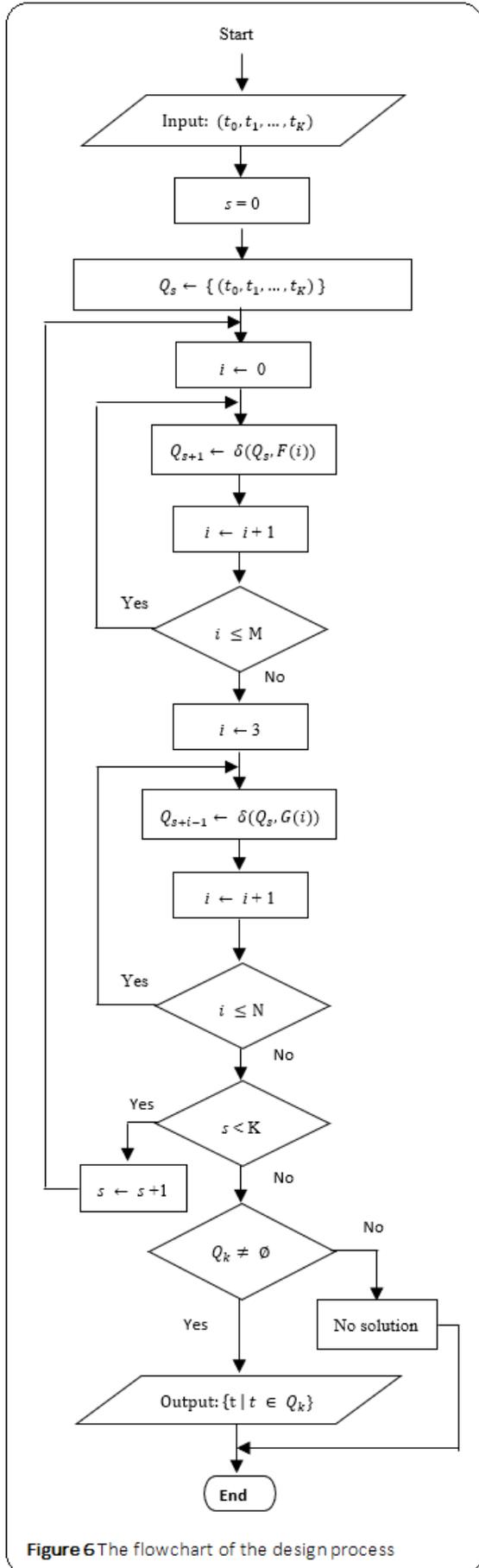
From Definition 4.2, the gearing forests in a state have the same number of gearing trees. Moreover, the higher the state is, the fewer the gearing tree number of a forest. Thus, connections proceed toward the successful direction when a gearing forest automaton runs. In the accept state, every gearing forest has only one gearing tree that is a design scheme.

#### 4.1 Implementation of an automatic process

A program has been developed based on a gearing forest automaton, and its source code is more than 100,000 lines. The flowchart is shown in Figure 6. As long as the basic parameters of the input shaft and output shafts are inputted, the program automatically generates the design schemes, produces the gearing system drawings, displays the gearing system data, and saves all data to an output file, such as an SQL database or an Excel spreadsheet. An input data interface is shown as Figure 7.

#### 4.2 Optimization

When there are many output shafts, design process is a fussy work needing a lot of calculations as a gearing system design is a non-deterministic polynomial hard [32]. In order to reduce time consuming of computation, after the automaton runs in a state, the newly generated gearing forests along with the existing gearing forests are optimized into a higher available state. The number of the gearing trees in a gearing forest, which stands for the height of a state, is considered as an optimal objective. A minimum number of the transmission shafts is as another optimal objective. Assume that  $\{t_1, t_2, \dots, t_L\}$  is a gearing forest that is composed of the gearing trees in the existing



and newly generated gearing forests. The optimization problem is written as

$$\arg \min \left( \sum_{i=1}^L C_i \cdot TNodes(t_i), \sum_{i=1}^L C_i \right),$$

$$U_{i \neq j, c_i \neq 0, c_j = 0}^L Interfere(t_i, t_j) = false, i, j = 0, 1, \dots, L,$$

$$\sum_{i=1}^L C_i \cdot PutNodes(t_i) = sumOutputNodes + 1,$$

$$C_i \in \{0, 1\},$$

(6)

where  $TNodes(t_i)$  is a function that computes the

number of the transmission nodes in the gearing tree  $t_i$ ,  $Interfere(t_i, t_j)$  is a function that judges whether the gearing tree  $t_i$  interferes with the gearing tree  $t_j$ ,  $PutNodes(t_i)$  is a function that computes the sum of the input nodes and the output nodes in the gearing tree  $t_i$ ,  $sumOutputNodes$  is the number of the output shafts. Note that a gearing tree at most has one input node. The constraint condition,

$$\bigcup_{i \neq j, C_i \neq 0, C_j = 0}^L Interfere(t_i, t_j) = false, i, j = 0, 1, \dots, L,$$

denotes that there is no interference in the gearing trees, so that  $\bigcup_{i=1}^L C_i \cdot t_i (C_i = 1)$  is a gearing forest. The constraint condition  $\sum_{i=1}^L C_i \cdot PutNodes(t_i) = sumOutputNodes + 1$  ensures the gearing forest belongs to a state of the gearing forest automaton.

The optimization procedure is outlined below. Assume that  $M$  is an  $L \times L$  matrix that denotes interferences between two trees. If a gearing tree  $t_i$  interferes with a Gearing tree  $t_j$ ,  $M[i, j] = 1, i \neq j, i, j = 1, 2, \dots, L$ , otherwise,  $M[i, j] = 0$ . The diagonal element  $M[i, i] = PutNodes(t_i), i = 1, 2, \dots, L$ . If  $(t_{i_1}, t_{i_2}, \dots, t_{i_M})$  is a minimum combination of  $(t_1, t_2, \dots, t_L)$ , and satisfies

$$\begin{aligned} C_i &= 1, i = i_1, i_2, \dots, i_M, \\ M[i, j] &= 0, i \neq j, i, j = i_1, i_2, \dots, i_M, \\ \sum_{i=1}^M M[i, i] &= sumOutputNodes + 1, \end{aligned}$$

$(t_{i_1}, t_{i_2}, \dots, t_{i_M})$  is an optimal solution. Note that  $M$  is a large sparse matrix and is very suitable to be describe by a generics, a data structure.

It is possible that the best design schemes are omitted although the probability is low. But the optimization saves an enormous amount of time and can perform a complicated gearing system design. It is apparently a qualitative leap, compared with a traditional design in which there is only a design scheme.

The flowchart of the optimization process is achieved when  $s \leftarrow s + 1$  of Figure 6 is replaced by Figure 8.  $ShaftTreeSet(Q)$  is a function that extracts all shaft trees of all states.  $Combine(treeSet, m)$  gets all combinations of  $m$  elements.  $Filtrate(forestSet)$  reserves the combinations that have  $sumOutputNodes + 1$  output shafts and constitute shaft forests.

### 5.3 Examples and discussions

A gearing system has three typical structures:

1. Multi-output shafts are on a circle.

2. Multi-output shafts are in a straight line.
3. Multi-transmission shafts are on a circle.

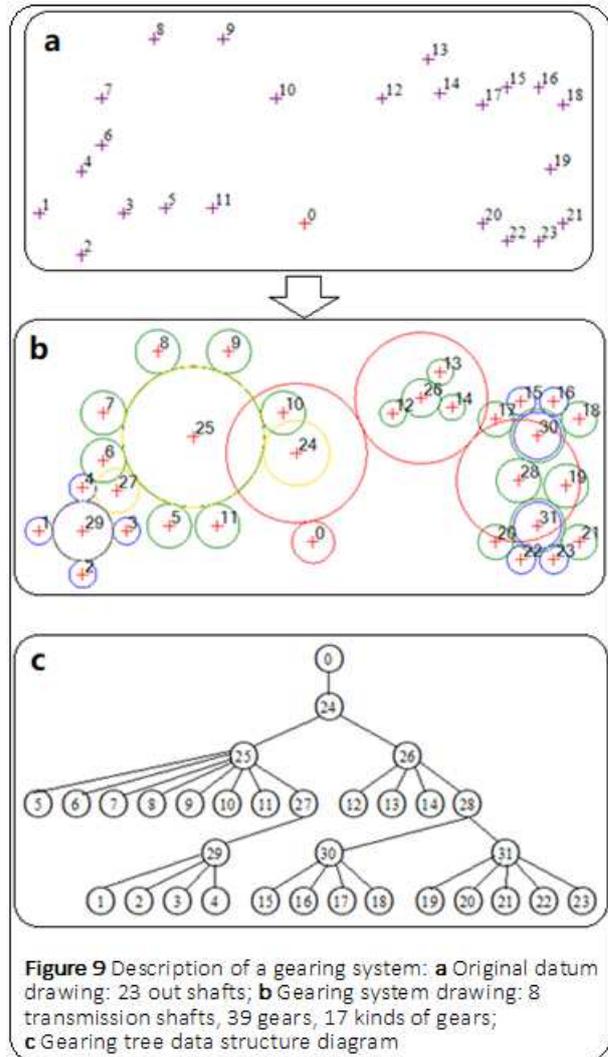
**Example 5.1** This example is a classic one from a design handbook [1]. There are 23 output shafts, and the original datum drawing is shown in Figure 9(a).

(1) Construction of the gearing forest automaton

Assume that 0 represents an input node and  $i (i = 1, 2, \dots, 23)$  represents output node. A gearing forest automaton is defined as follows:

$$A = (Q, \Sigma, \delta, Q_0, Q_{23}).$$

Automaton  $A$  has 24 states denoted by  $Q_0, Q_1, \dots, Q_{23}$ . A gearing forest has  $24 - i$  gearing trees in the  $Q_i (i = 0, 1, \dots, 23)$  state, where  $Q_0$  is the start state and  $Q_{23}$  is the accept state.



**Figure 9** Description of a gearing system: **a** Original datum drawing: 23 out shafts; **b** Gearing system drawing: 8 transmission shafts, 39 gears, 17 kinds of gears; **c** Gearing tree data structure diagram

(2) Running of the gearing forest automaton

The start state  $Q_0 = \{ (0, 1, 2, \dots, 23) \}$  is composed of an input node and all the output nodes, as shown in

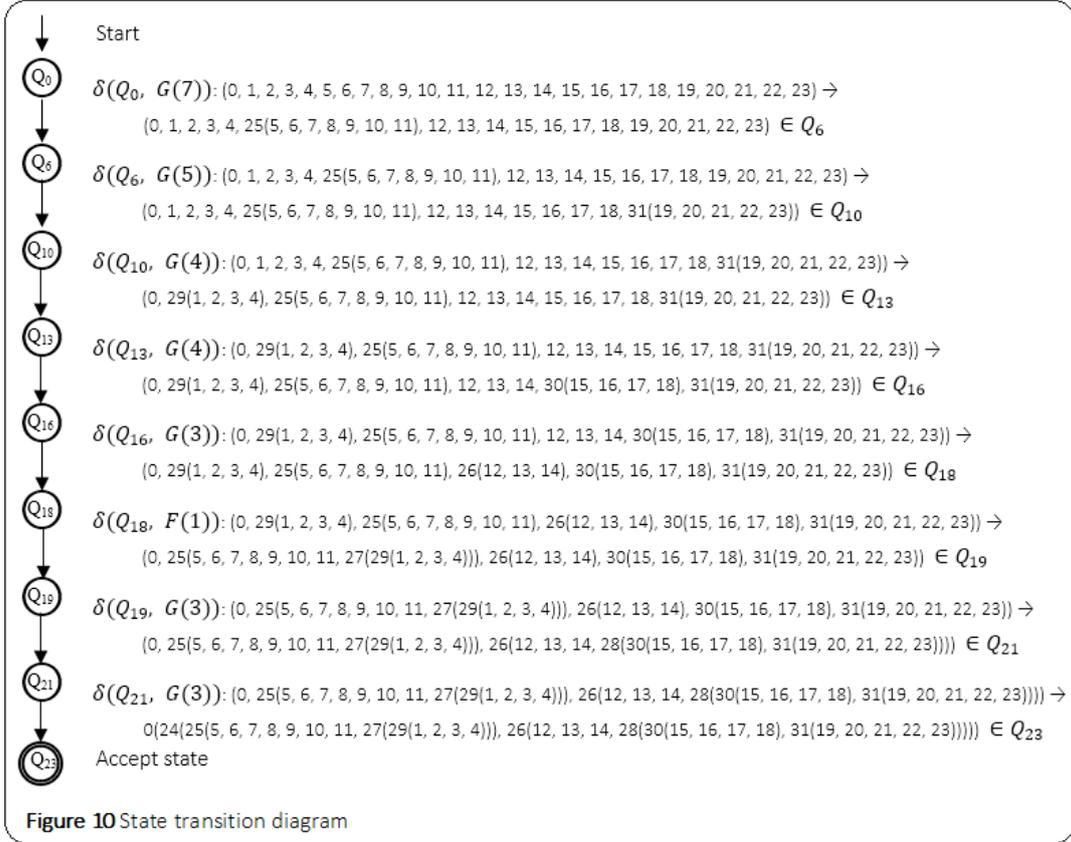


Figure 10 State transition diagram

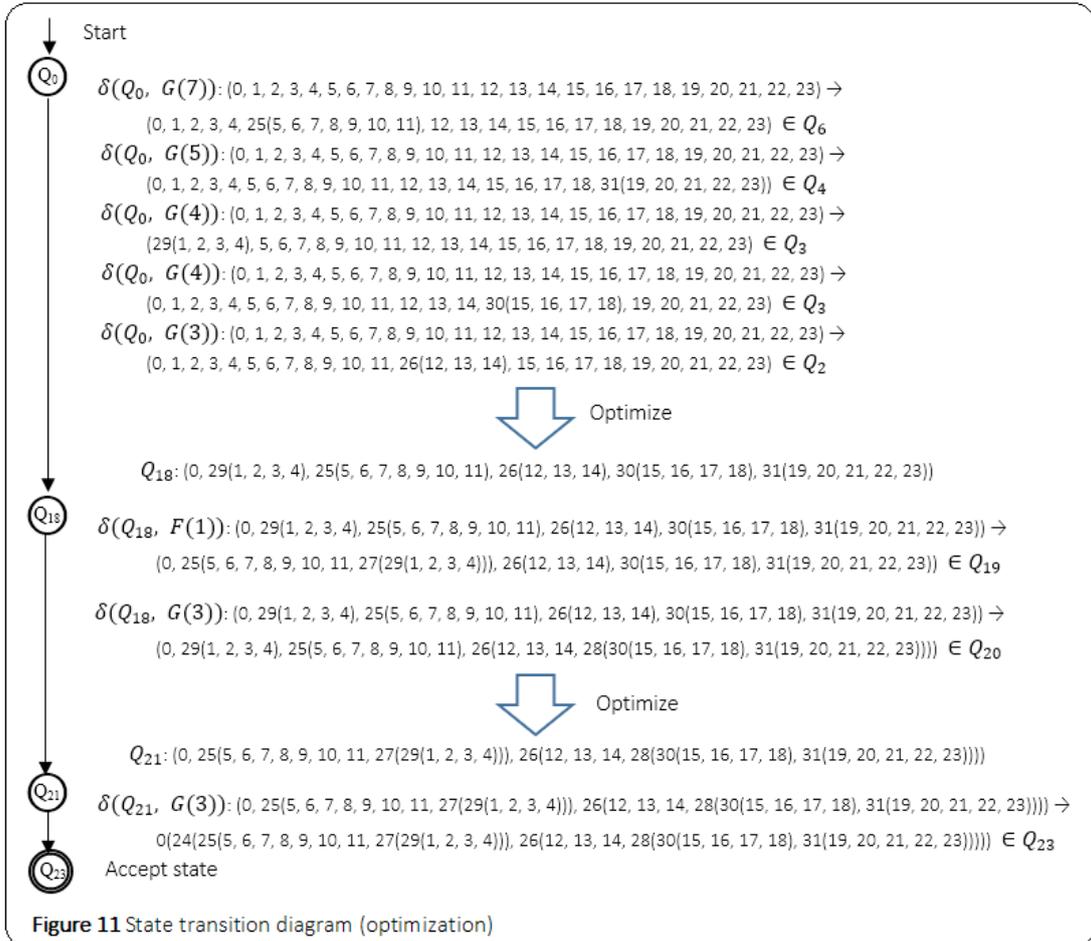


Figure 11 State transition diagram (optimization)

**Table 2 Gearing system data (Figure 9(b))**

A gear is denoted by (teeth, module, modification), or by (teeth, module) (modification = 0)

No	Gear ( I )	Gear( II )	Gear(III)	Gear(IV)	(X, Y)	Speed	Diameter	Torch
0				(24,3, -0.071)	(450,170)	720	20	13283
1	(23,2)				(-13.457,188.885)	510	15	248
2	(23,2)				(60.543,114.885)	510	15	248
3	(23,2)				(134.543,188.885)	510	15	248
4	(23,2)				(60.543,262.885)	510	15	248
5		(25,3)			(206.639,197.717)	340	20	1013
6		(25,3)			(95.27,309.086)	340	20	1013
7		(25,3)			(95.27,390.614)	340	20	1013
8		(25,3)			(187.13,495.361)	340	20	1013
9		(25,3)			(307.676,495.361)	340	20	1013
10		(25,3)			(399.536,390.614)	340	20	1013
11		(25,3)			(288.167,197.717)	340	20	1013
12		(22,2)			(585,390)	340	20	1013
13		(22,2)			(665,460)	340	20	1013
14		(22,2)			(685,400)	340	20	1013
15	(24,2)				(802,410)	340	20	1013
16	(24,2)				(857,410)	340	20	1013
17		(20,3)			(759,380)	300	30	2716
18		(20,3)			(900,380)	300	30	2716
19		(25,3)			(878,266)	245	30	4067
20		(20,3)			(759,170)	300	30	2716
21		(20,3)			(900,170)	300	30	2716
22	(24,2)				(802,140)	340	20	1013
23	(24,2)				(857,140)	340	20	1013
24			(37,3,0.132)	(80,3, -0.381)	(422.065,322.103)	216	20	44276
25		(80,3)	(80,3,0.319)		(247.403,349.85)	100	25	27818
26		(33,2, -0.255)		(75,3, -0.19)	(632.903,415.968)	230	15	29447
27			(25,3)		(118.423,257.804)	320	15	1602
28		(25,3)		(70,3, -0.175)	(797.099,275)	247	25	23229
29	(52,2, -0.5)		(35,3)		(60.543,188.885)	228	15	2243
30	(41,2, -0.342)	(31,3, -0.197)			(829.5,351.859)	199	15	11881
31	(41,2, -0.342)	(31,3, -0.197)			(829.5,198.141)	199	15	16924

Figure 9(a). Figure 10 is a state transition diagram for creating the design scheme in Figure 9(b), and Figure 11 is an optimal state transition diagram.

**(3) Design scheme analysis**

The design scheme of Figure 9(b) is created by the program, which has 8 transmission shafts, 39 gears and 17 kinds of gears. The number of transmission shafts, the most important optimization criterion [1], [2], is reduced by 27% than the design handbook, from 11 to 8. The data

are shown in Table 2. There are five connections of case 1:

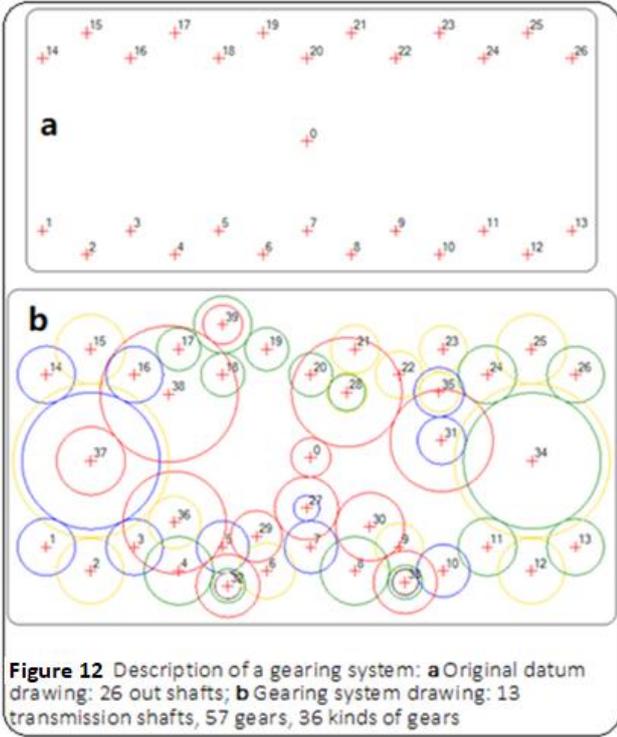
- (i)  $G(7)$ : 25(5, 6, 7, 8, 9, 10, 11).
- (ii)  $G(5)$ : 31(19, 20, 21, 22, 23).
- (iii)  $G(4)$ : 29(1, 2, 3, 4).
- (iv)  $G(4)$ : 30(15, 16, 17, 18).
- (v)  $G(3)$ : 26(12, 13, 14).

In addition, there are two connections of case 3:

- (i)  $G(3)$ : 26(12, 13, 14, 28(30(15, 16, 17, 18), 31(19, 20, 21, 22, 23))). Note that shaft 26 is the root of this tree, and

the trees whose roots are shafts 28, 30 and 31, are not used in the design.

(ii)  $G(3)$ : 0(24(25(5, 6, 7, 8, 9, 10, 11, 27(29(1, 2, 3, 4))), 26(12, 13, 14, 28(30(15, 16, 17, 18), 31(19, 20, 21, 22, 23))))). Note that shaft 24 is a new added transmission shaft.



**Example 5.2** The example is one of case 2, from an engine works. There are 26 output shafts.

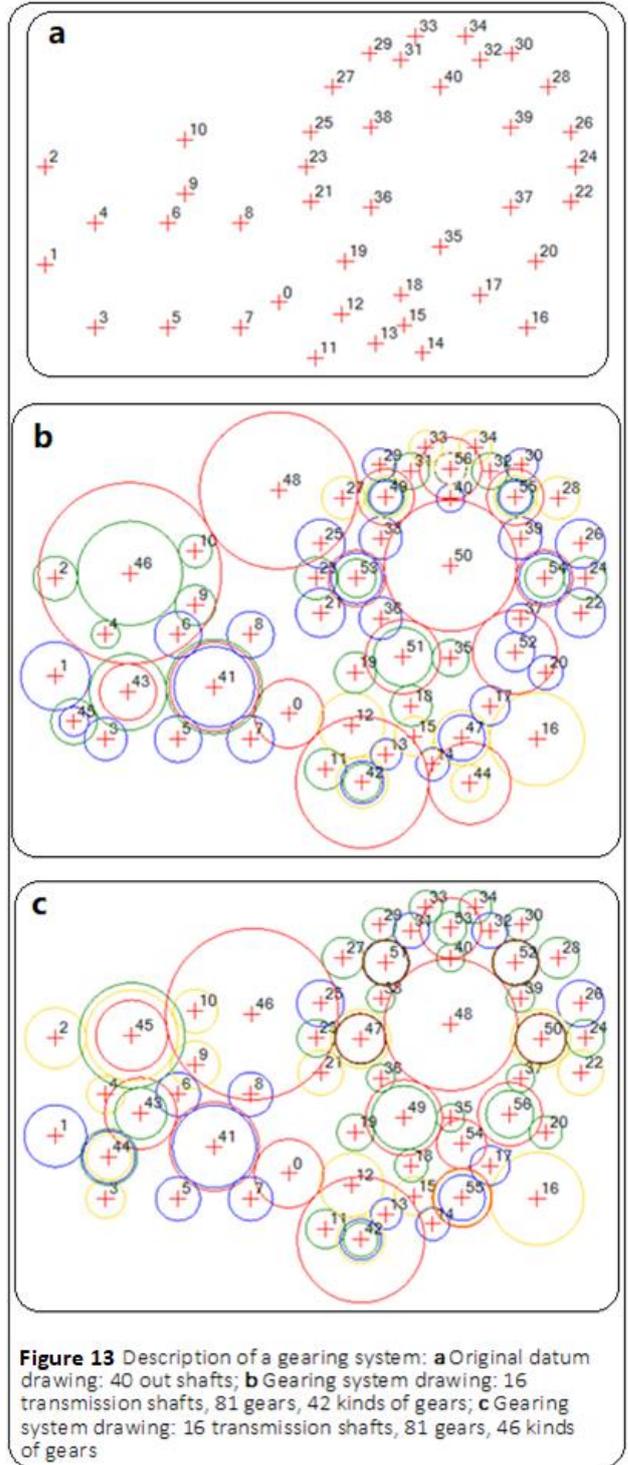
Figure 12 (a) is the original datum drawing, and Figure 12 (b) is a design scheme that is created by the automaton. The transmission shafts are reduced by 50% than the engineer’s scheme that is generated by using the BOXCAD, a current software tool of designing a gearing system, from 26 to 13.

There are four lines:

- (i) 1, 3, 5, 7, 9, 11, 13.
- (ii) 2, 4, 6, 8, 10, 12.
- (iii) 14, 16, 18, 20, 22, 24, 26.
- (iv) 15, 17, 19, 21, 23, 25.

**Example 5.3** The example is from the rear bridge of a vehicle in a plant, and is a very complicated one. There are 40 output shafts, and the original datum drawing is shown in Figure 13(a).

Figure 13(b) and Figure 13(c) show two gearing system design schemes that are automatically created by the program. Compared with engineer’s design scheme, the transmission shafts are all reduced by 43% from 28 to 16.

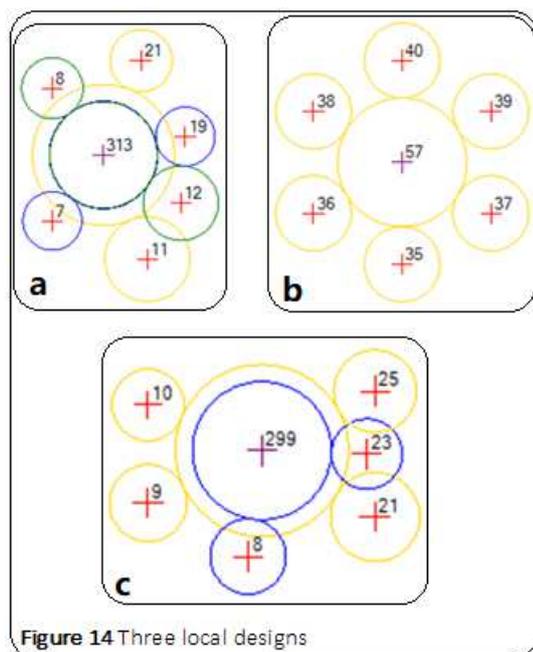


In Figure 13(b), there is a connection in which 7 trees of multiple nodes are assembled into a new tree by a new added shaft 50:

49(27, 29, 31, 50(51(18, 19, 35), 52(20, 37), 53(21, 23, 25, 36, 38), 54(22, 24, 26, 39), 55(28, 30, 32), 56(33, 34, 40))),

which is hardly imaginable if an artificial design is used.

It is a basic design principle that one transmission shaft drives as more output shafts as possible [1], [2]. Three local designs, shown in Figure 14, are optimal because one transmission shaft drives 6 output shafts that are the most. However, they are not optimal if we look at the whole picture as they don't appear in the optimized solutions in Figure 13. Analogously, although shaft 50 in Figure 13(b) drives the more shafts than shaft 48 in Figure 13(c), two solutions use the equal transmission shafts, even the equal gears. Therefore, the design principle needs to be investigated.



## 6 Conclusions

The paper proposed a gearing forest automaton as an automatic design tool to automatically create the design schemes for a gearing system. The principal aim of the presented work is to transform a conventional design way into a new one in which multiple local designs are automatically assembled together without artificial intervention until the solutions appear. For automation, it is crucial that a design process is described by data structure.

A program for realizing the automaton has been developed, and is validated. Compared with the traditional methods, the program design not only significantly decreases the designer's workload and avoids design errors as desired but also, more importantly, greatly improves the design quality. In addition, a traditional design guideline that one shaft directly connects as more shafts as possible, is challenged.

### Availability of data and materials

The datasets used or analyzed during the current study are available from the corresponding author on reasonable request.

### Competing interests

The authors declare no competing financial interests.

### Funding

No Funding.

### Acknowledgements

Not applicable.

### Authors' contributions

HZ proposed the modeling and wrote the manuscript; ZY validated the model; QZ did the decision support and data processing. All authors read and approved the final manuscript.

### Authors' information

Huazeng Zhao, born in 1961, is currently a professor at Shandong College of Information Technology, China. He received his Master degree from Jilin University of technology, China, in 1988. His research interests include mechanical drive, Data mining and engineering design automation.

Zuxiao Yang, born in 1960, is currently a professor at Weifang University, China. He received his bachelor degree from Shandong University of Technology, China, in 1982. His research interest is a focus on applications of advanced techniques in manufacturing.

QiuShi Zhao, born in 1995, is a senior student at Virginia Tech, USA. She is interested in decision support and data processing.

### Author Details

<sup>1</sup>Department of Information, Shandong College of Information Technology, Weifang 261061, China. <sup>2</sup>School of Beihai, Weifang University, Weifang, China. <sup>3</sup>Pamplin College, Virginia Tech, Blacksburg, VA24060, USA.

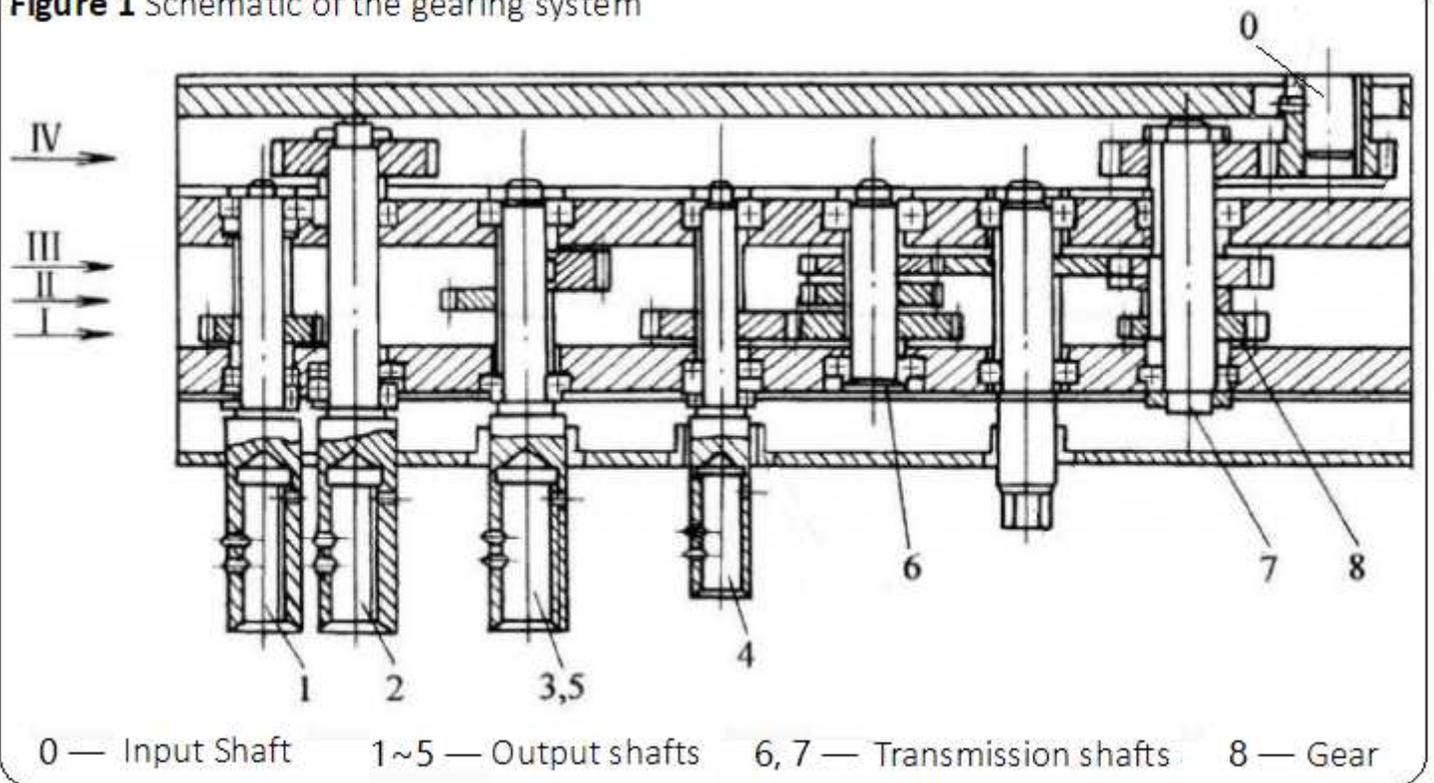
### References

- [1] Dalian Combined Machine Tool Institute. Combined Machine Tool Design, China Machine Press, 1978.
- [2] Y J Sun, C G Guo, J P Wei. Design of Machine Manufacturing Equipment. Beijing Institute of Technology Press, 2017.
- [3] W C Yu. General program on coordinates in gearing system of multi-spindle head. Journal of Fudan University (Natural Science), 1973, 3: 50-61.
- [4] Section of Mechanical Manufacture of Xi'an Jiaotong University, China. General program for coordinates in gearing system of multi-spindle head of modular machine tool. Journal of Xi'an Jiaotong University, China, 1977, 1: 39-55, 1977.
- [5] L Z Zhou, B S Tong. Interactive design of gearing systems in multi-spindle heads of combined machine tool. Journal of Engineering Graphics, 1984, Z1: 31-40.
- [6] Y Fukuda, T Watanabe and, M Yamamoto. Developmental research of CAD/CAM system for multi tool spindle head. Research reports Kumano technical college, 1993, 10: 19-33.
- [7] F Hüseyin, M Dülger. Computer-aided design of multispeed gearboxes. Computer-Aided Design, 1993, 25(11): 720-726.
- [8] M Ogrnjancovic. Decisions in gear train transmission design. Research in Engineering Design. 8: 178-187.
- [9] M A Nasser, F R Gomma, M A Asy, et al. Computer Aided Design of Multi-Stage Gearboxes. International Journal of Advanced Engineering and Global Technology, 2014, 2(12): 148-157.
- [10] H Maeng, J Park, H Park. Design of Special-purpose Machine Tool Based on a Multi-spindle Head. Journal of the Korean Society of Manufacturing Technology Engineers, 2015, 24(6): 675-681.

- [11] Y B Li, S X Weng. Computer aided design on multi-spindle head of modular machine tool—design and check of transmitting and components. *Journal of Shanghai Jiaotong University*, 1984, 18(4): 89-98.
- [12] Z Q Cao, Z Y Sun, J N Zhong, et al. Calculation of coordinates for the multi spindle headstock of the modular machine tools by the general parameter method. *Journal of Hunan University*, 1984, 11(4): 13-24, 1984.
- [13] S S Vadnere, A A Deshmukh, S R Chauhan, et al. Design and analysis of a multi spindle drill gearbox - Torque divider. *International Journal on Recent Technologies in Mechanical and Electrical Engineering*, 2014, 1(4): 29-33.
- [14] P F Mcgoldrick, D Bruce. An interactive program for the computer-aided design of gearing system in multi-spindle drilling head. *Computer-Aided Design*, 1982, 14(2): 182-189.
- [15] G H Shi. Study on intelligent design of modular machine tool spindle box. *Fujian Agriculture and Forestry University*, 2006.
- [16] J Han. Research of intelligence system for spindle box of modular machine. *Agricultural University of Hebei*, 2012.
- [17] H Zhang. Research and development of automation design system for special machine tool spindle box. *Xi'an technological university*, 2019.
- [18] A Armillotta. Tolerance analysis of gear trains by static analogy. *Mechanism and Machine Theory*, 2019, 135: 65-80.
- [19] S Prayoonrat, D Walton. Practical approach to optimum gear train design. *Computer Aided Design*, 1988, 20(2): 83-92.
- [20] C W Wu, Y H Wu, C B Ma, et al. Optimization design of gear train based on particle swarm optimization algorithm. *Applied Mechanics and Materials*, 2013, 373-375: 1072-1075.
- [21] C Gologlu, M Zeyveli. A genetic approach to automate preliminary design of gear drives. *Computers & Industrial Engineering*, 2009, 57(3): 1043-1051.
- [22] K Holder, S Rudolph, R Stetter, et al. Automated requirements-driven design synthesis of gearboxes with graph-based design languages using state of the art tools. *Forschung im Ingenieurwesen*, 2019, 83(3): 655–668.
- [23] T H Chong, I Bae, G J Park, et al. A new and generalized methodology to design multi-stage gear drives by integrating the dimensional and the configuration design process. *Mechanism & Machine Theory*, 2002, 37(3): 295-310.
- [24] B Aberšek, J Flašker, J Balič. Expert system for designing and manufacturing of a gear box. *Expert Systems with Applications*, 1996, 11(3): 397-405.
- [25] V V Lozin. Gearing optimization. *Optimization and Engineering*, 2008, 9: 201-211.
- [26] N Marjanovic, B Isailovic, V Marjanovic, et al. A practical approach to the optimization of gear trains with spur gears. *Mechanism & Machine Theory*, 2012, 53: 1-16, 2012.
- [27] W C Yu. Analysis for gearing system of multi-output shafts by the graph theory method. *Journal of Fudan University (Natural Science)*, 1977, 2: 67-72.
- [28] Joint Research Group from Shanghai Jiaotong University, Fudan University, Shanghai Electrical Mechanical Design Institute, China. Auto design on transmission system of output shaft box of modular machine. *Journal of Mechanical Industry automation Design*, 1977, 1: 31-35.
- [29] D D Sun. A computer-aided design system of a gearing system of a multi-spindle head of a drilling modular machine tools. *Chinese Manufacturing Automation*, 1983, 2: 1-10, 48.
- [30] Y L Dong, M R Jiang. Review of Chinese modular machine tools spindle box CAD system. *Modular Machine Tool & Automatic Manufacturing Technique*, 1986, 11: 2-7.
- [31] Y L Dong. The drilling modular machine tools spindle box CAD system pass appraisal. *Modular Machine Tool*, 1984, 2: 2-2.
- [32] Y P Lu, J Z Cha. Principle of Expansion Method for Layout Design. *Chinese journal of computers*, 2001, 24(1): 1077-1084.
- [33] Y P Lu, J Z Cha, C W Tony. Expansion approach for packing design of planar gear transmission. *Chinese journal of Mechanical Engineering*, 2001, 37(12): 16-20.
- [34] Y P Lu, J Z Cha, J Y Li. Cluster Analysis for Gear Space Arrangement of Complex Gear Transmission. *Journal of Computer-Aided Design & Computer Graphics*, 2001, 13(6): 505-508.
- [35] W R Laxon, K H Malby, D H Mountford, et al. Automatic design of structural steelwork. *Computer-Aided Design*, 1980, 12(1): 35-42.
- [36] A S Kumar, J Y H Fuh, T S Kow. An automated design and assembly of interference-free modular fixture setup. *Computer-Aided Design*, 2000, 32(10):583-596.
- [37] D Potočnik, B Dolšak, M Ulbin. An automated design system for compound washer dies. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 2013, 35: 293–304.
- [38] K S Kul'ga, P V Vinogradov. Automated design of model sets in the manufacture of large structural castings for the basic components of mechatronic machine tools. *Russian Engineering Research*, 2016, 36:476–483.
- [39] M Delavar, G T Bitsuamlak, J K Dickinson, et al. Automated BIM-based process for wind engineering design collaboration. *Building Simulation*, 2020, 13: 457–474.
- [40] N Inoue, N Shimotai, T Uesugi. Cellular automaton generating topological structures. *Proceedings of SPIE - The International Society for Optical Engineering*, 1994, 2361.
- [41] E Kita, T Toyoda. Structural design using cellular automata. *Struct Multidisciplinary Optimization*, 2000, 19(1): 64-73.
- [42] S Setoodeh, Z Gürdal, L T Watson. Design of variable-stiffness composite layers using cellular automata. *Computer Methods in Applied Mechanics and Engineering*, 2006, 195(9-12): 836-851.
- [43] C M Herr, R C Ford. Cellular automata in architectural design: from generic systems to specific design tools. *Automation in Construction*, 2016, 72(1): 39-45.
- [44] D Agarwal, T T Robinson, C G Armstrong, et al. Enhancing CAD-based shape optimisation by automatically updating. *Structural and Multidisciplinary Optimization*, 2019, 59(5): 1639–1654.

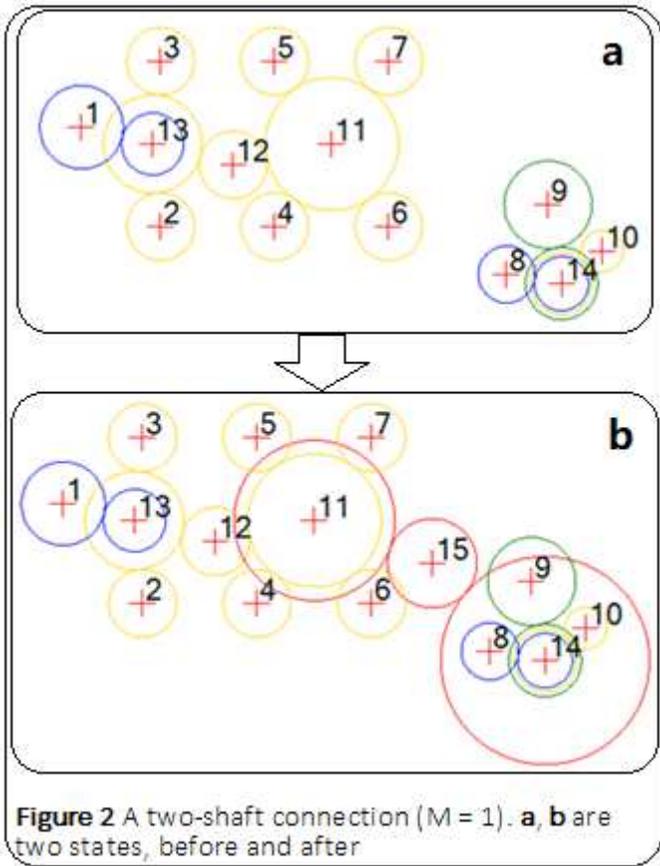
# Figures

**Figure 1** Schematic of the gearing system



**Figure 1**

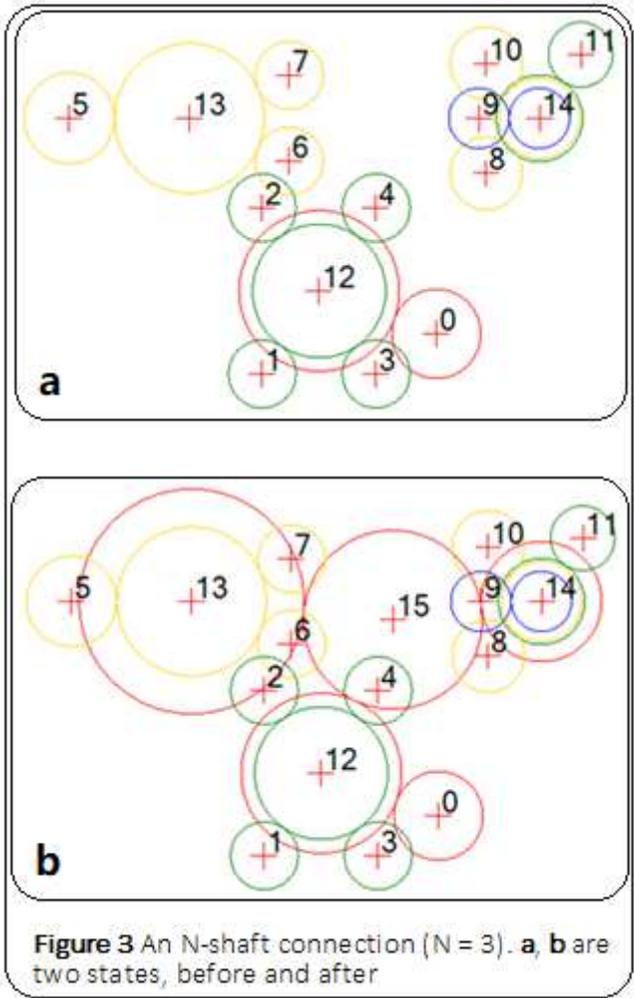
(caption included in graphic)



**Figure 2** A two-shaft connection ( $M = 1$ ). **a**, **b** are two states, before and after

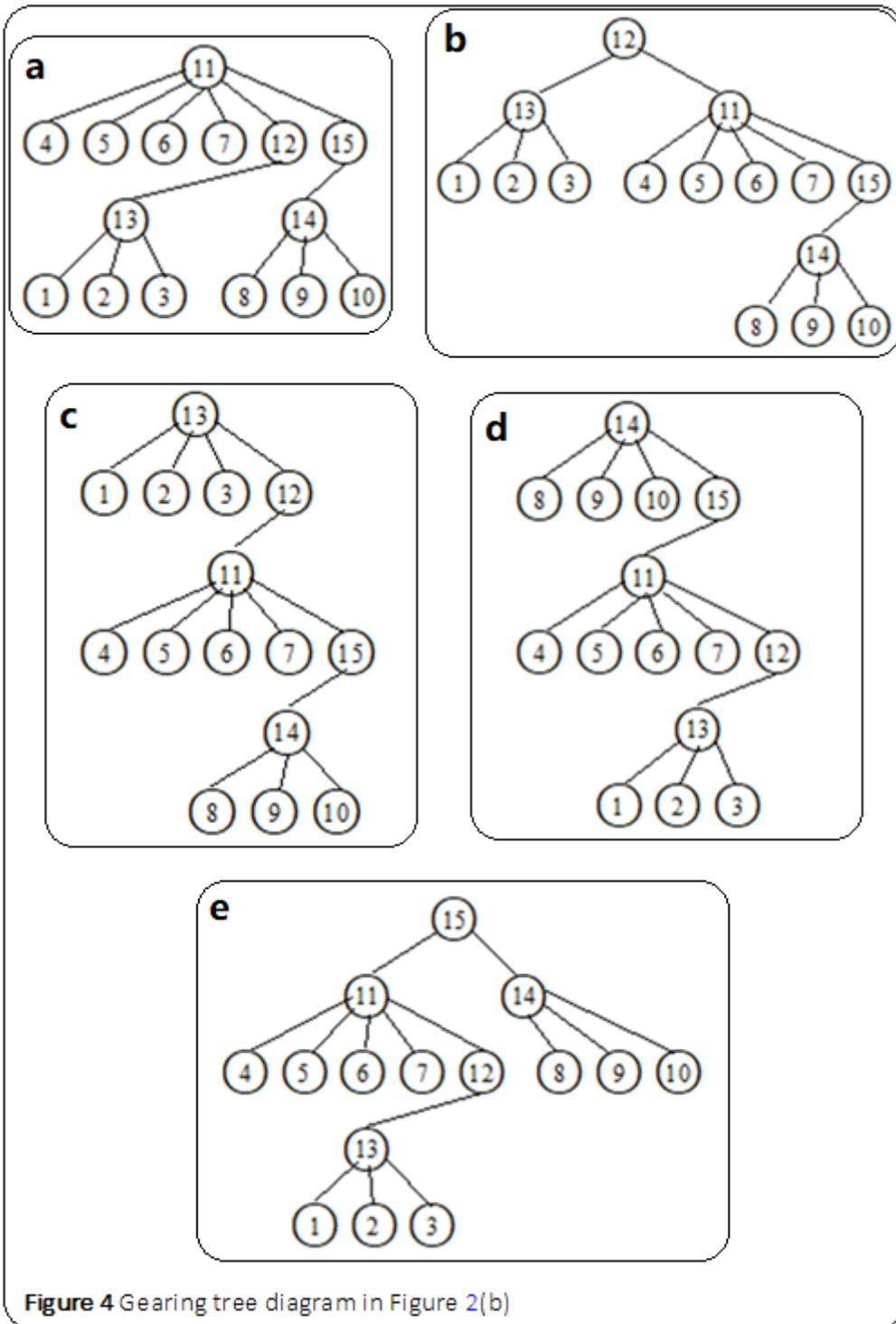
**Figure 2**

(caption included in graphic)



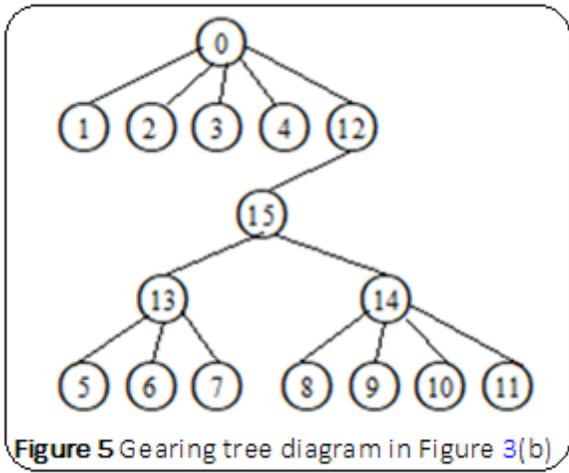
**Figure 3**

(caption included in graphic)



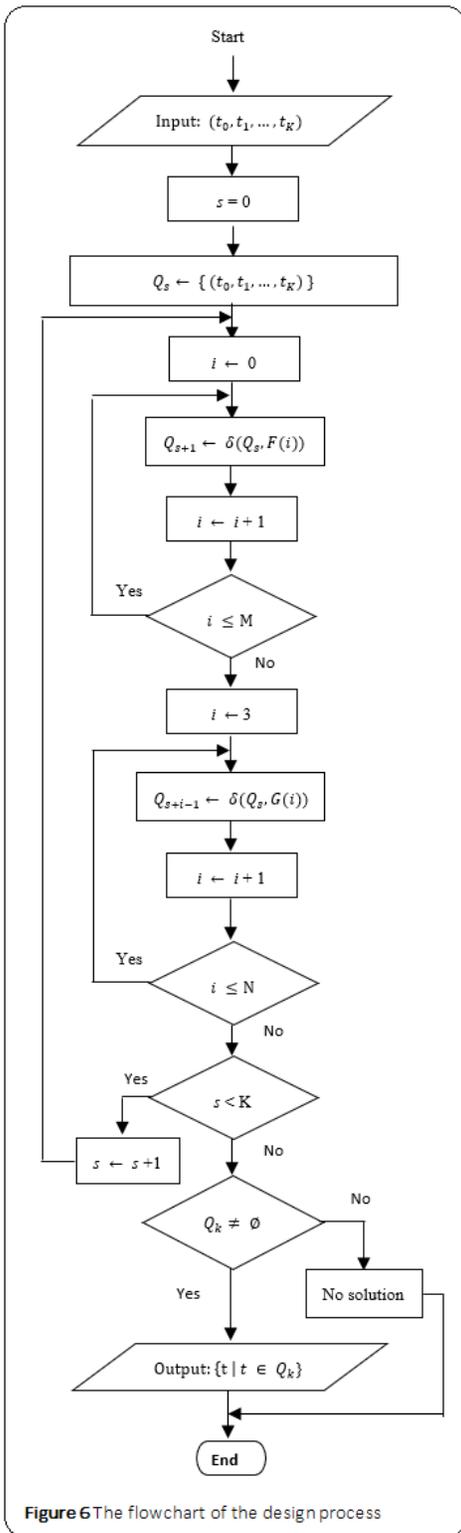
**Figure 4**

(caption included in graphic)



**Figure 5**

(caption included in graphic)



**Figure 6**

(caption included in graphic)

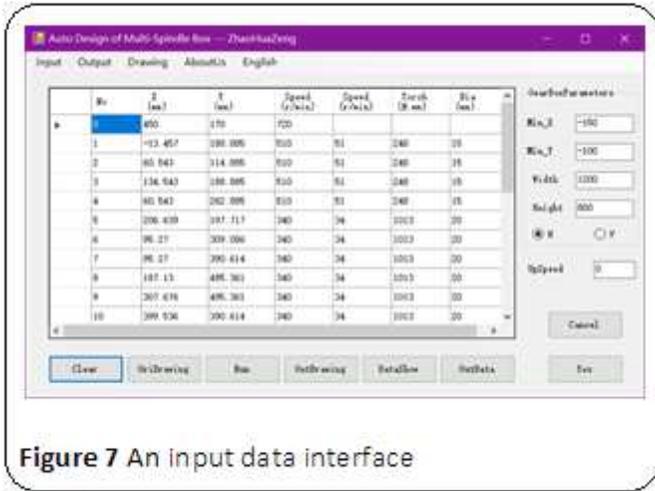


Figure 7 An input data interface

Figure 7

(caption included in graphic)

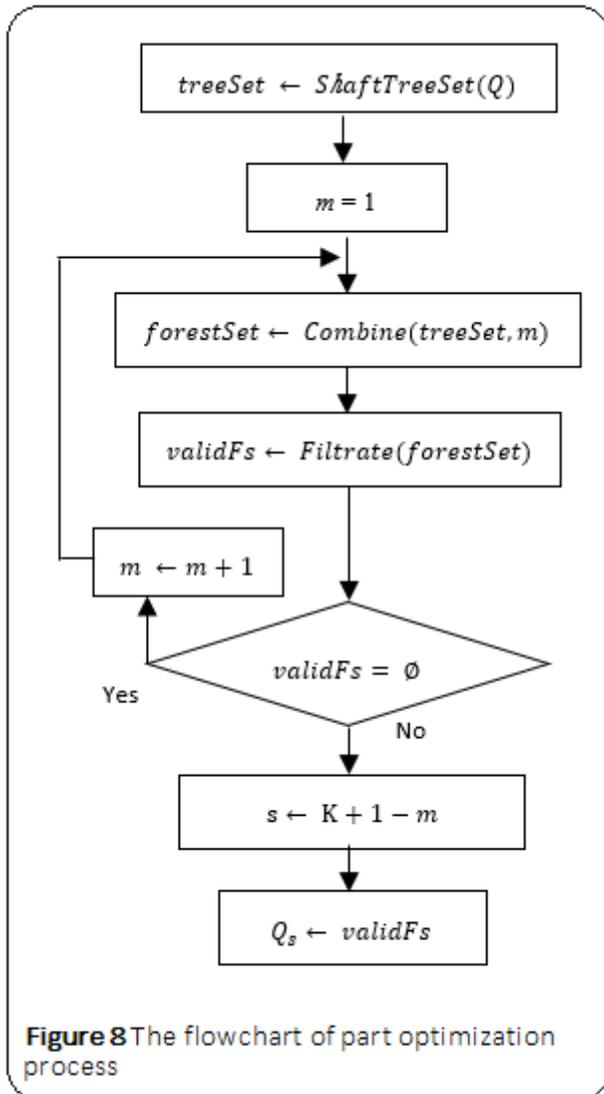
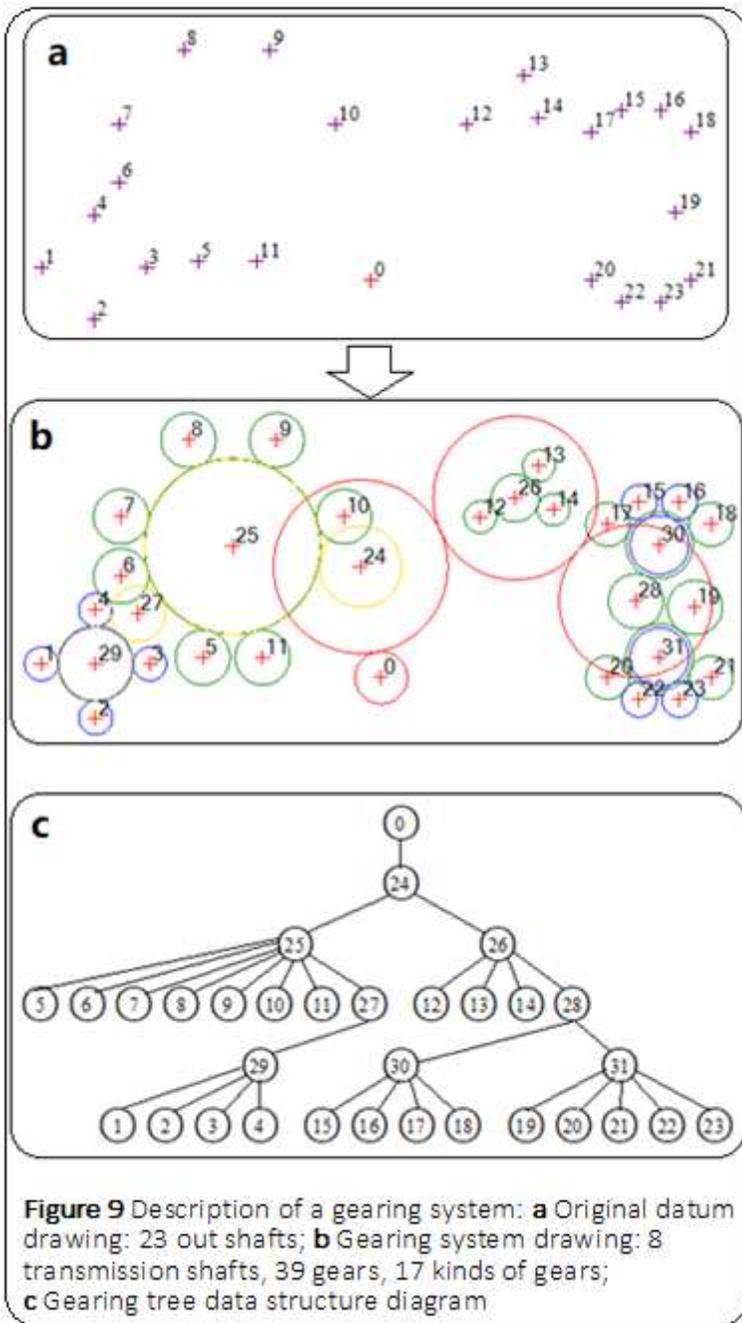


Figure 8 The flowchart of part optimization process

Figure 8

(caption included in graphic)



**Figure 9**

(caption included in graphic)

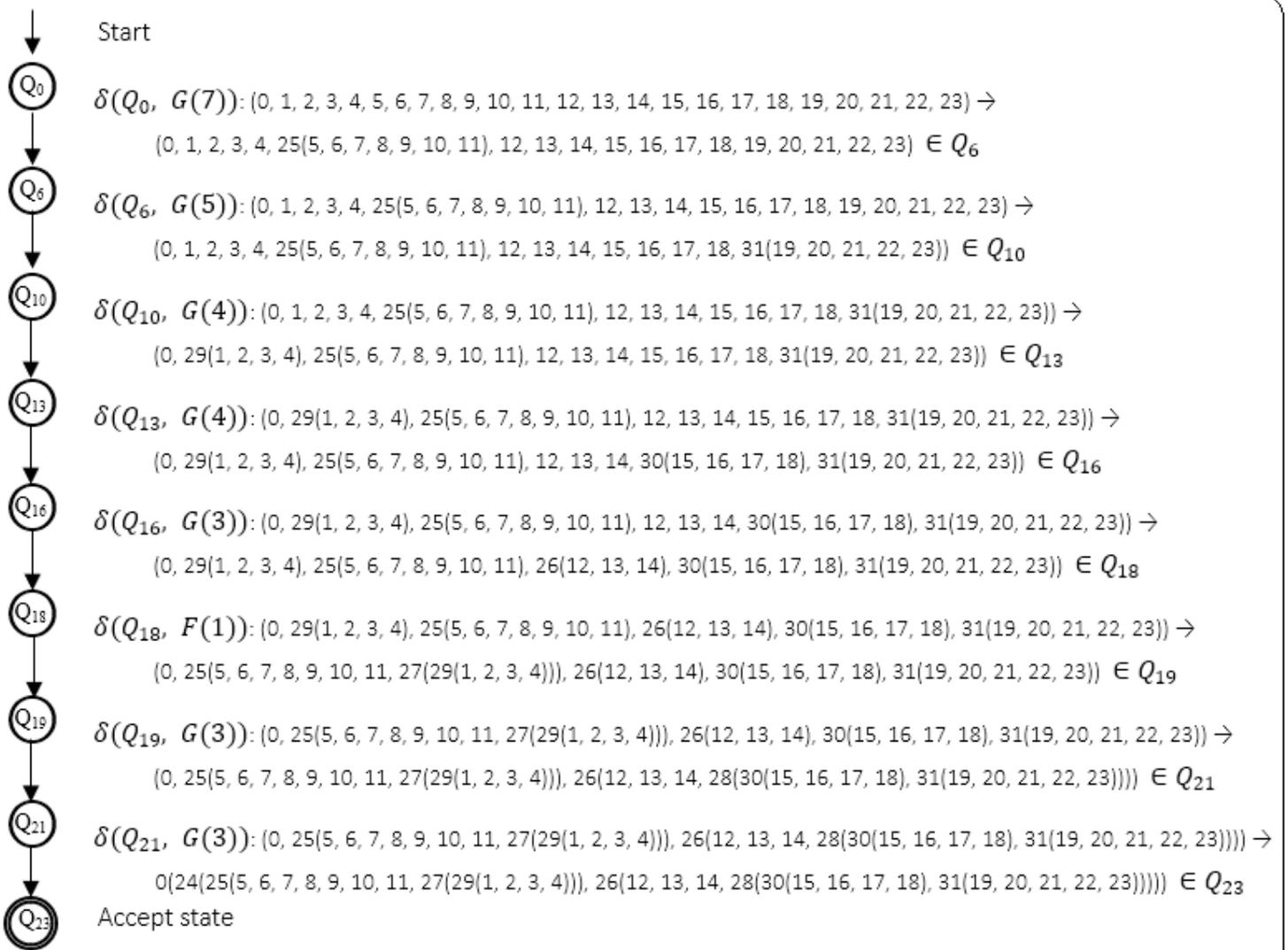


Figure 10 State transition diagram

Figure 10

(caption included in graphic)

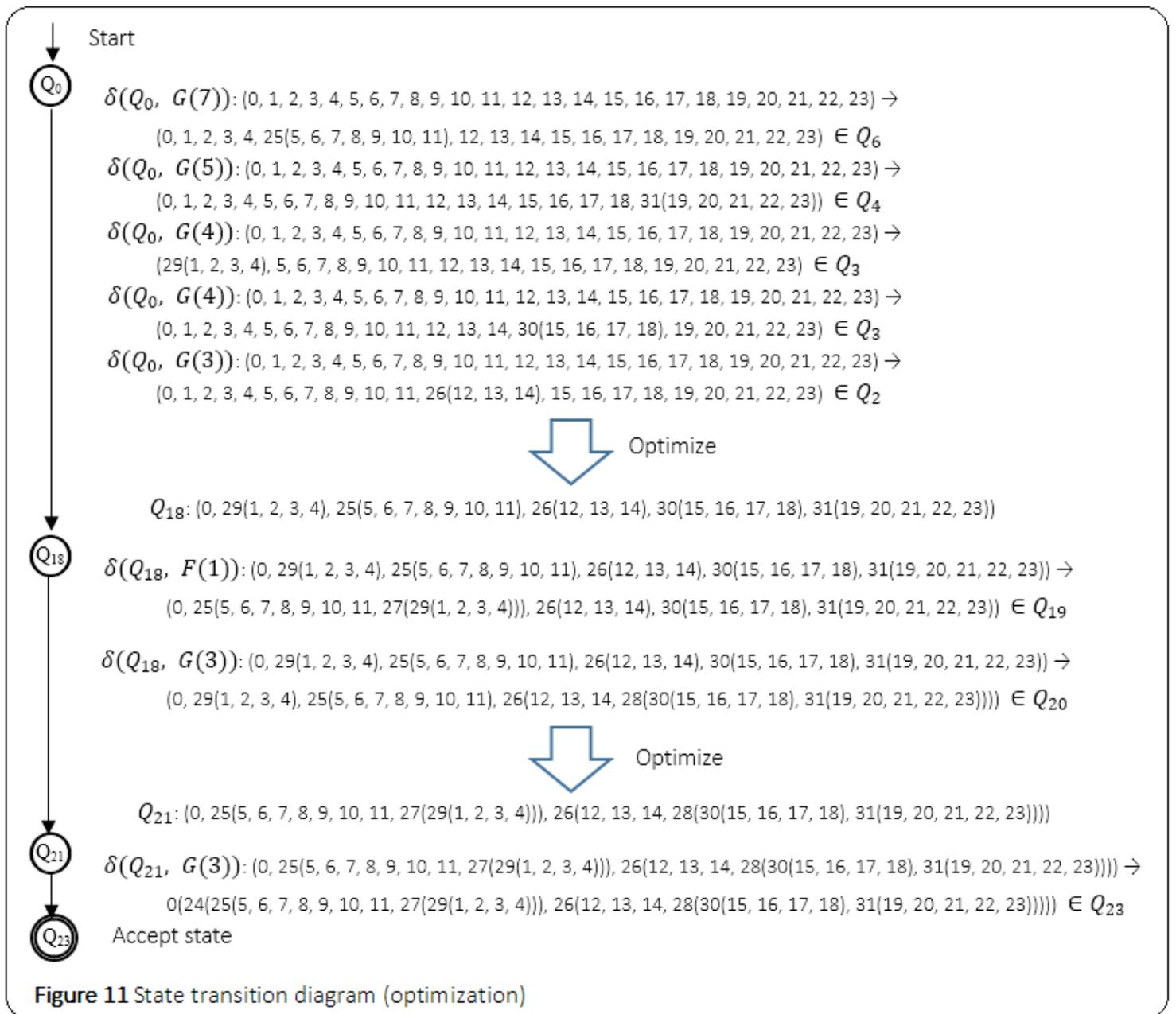


Figure 11 State transition diagram (optimization)

Figure 11

(caption included in graphic)

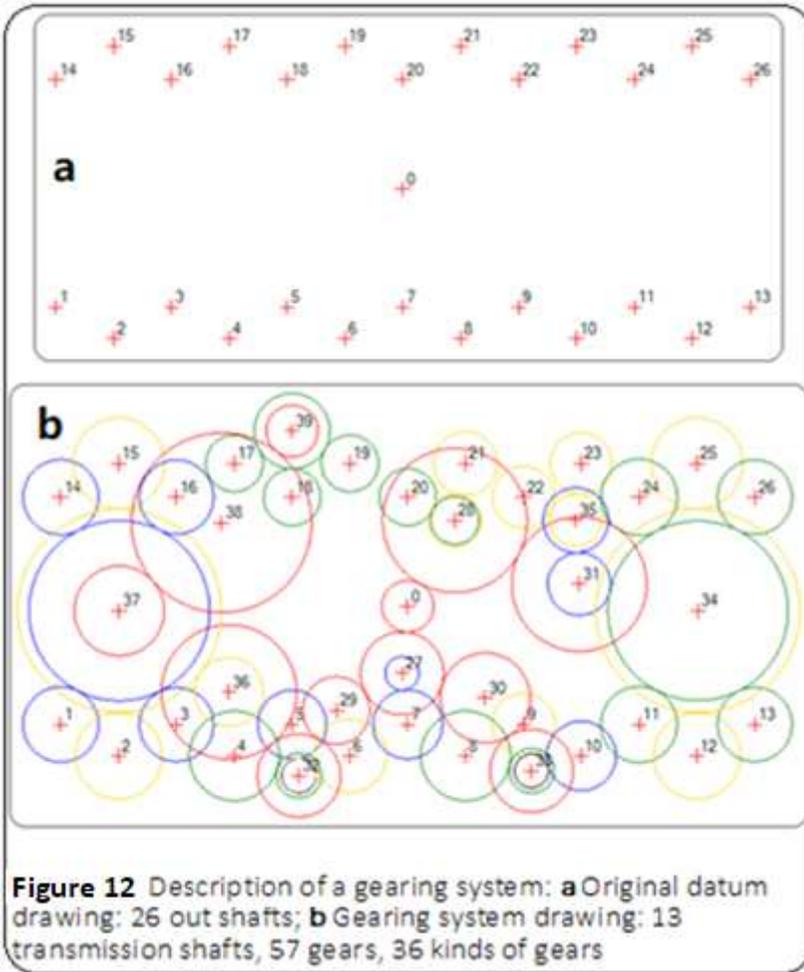
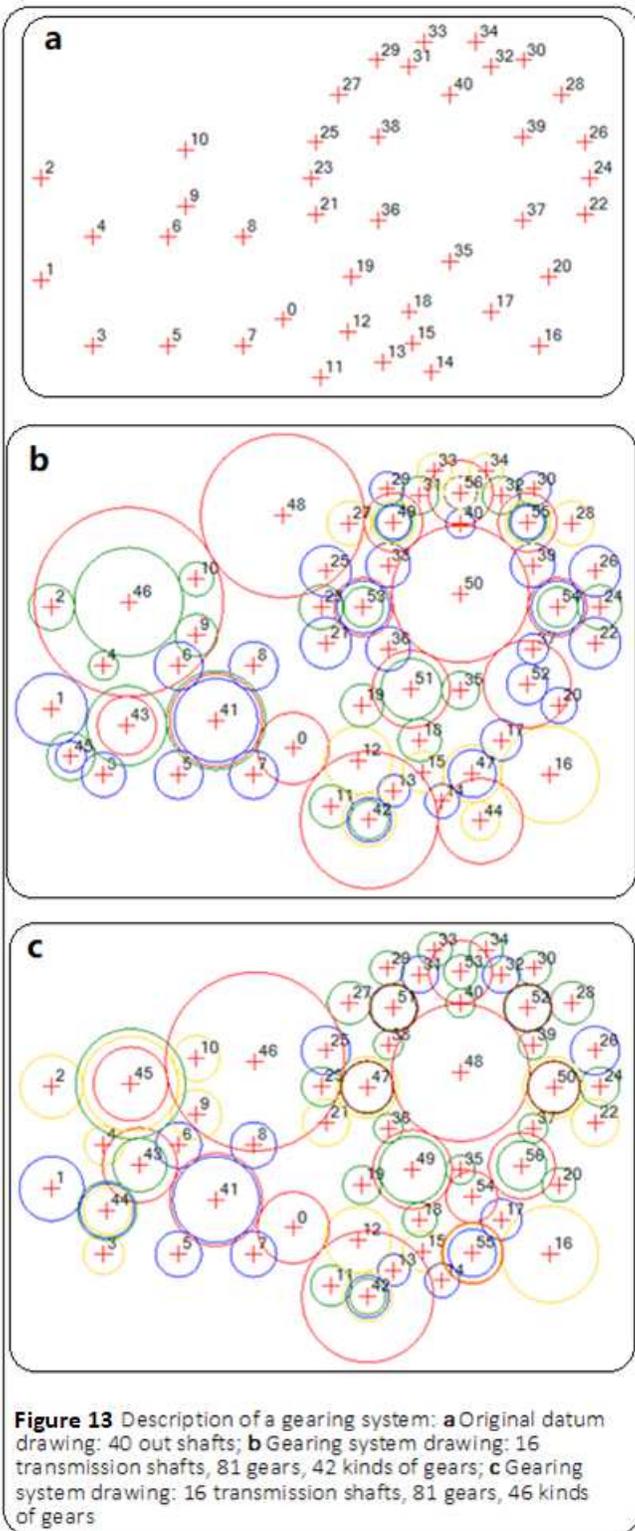


Figure 12

(caption included in graphic)



**Figure 13**

(caption included in graphic)

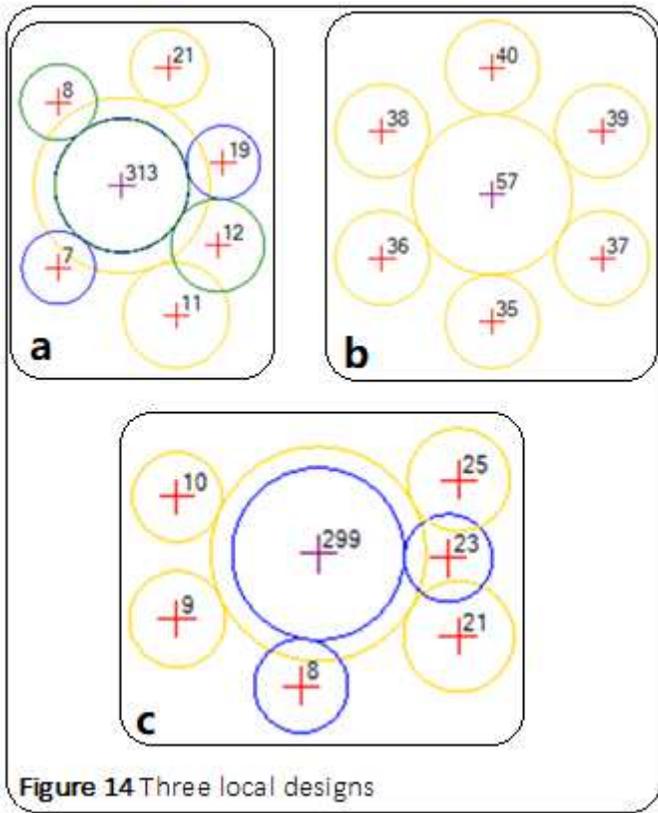


Figure 14 Three local designs

Figure 14

(caption included in graphic)