Path Planning for Product Function Transformation based on Kruskal Algorithm

Jinpu Zhang[1], Guozhong Cao[2], Qingjin Peng[3], Runhua Tan[4] and Huangao Zhang[5]

[1]Hebei University of Technology, Jinpu_zhang@163.com
[2]Hebei University of Technology, Caoguozhong@hebut.edu.cn
[3]University of Manitoba, Qingjin.Peng@umanitoba.ca
[4]Hebei University of Technology, Rhtanhebut@163.com
[5]Hebei University of Technology, Zhgzwy@hebut.edu.cn

Corresponding author: Guozhong Cao, Caoguozhong@hebut.edu.cn

Abstract. Technical systems with only one unique function cannot meet the diverse needs of users. The transformable system can perform different functions through the structure change in the application. A key step of designing a transformable system is to search an optimal path of the transformation process between different functions of the system. Through analyzing the system composition, the diversity needs can be implemented in multiple environments and actions. Different use environments and active objects form different states of a system. The distance between different use environments or active objects can be measured based on similarity of their characteristic parameters. In order to save resources and simplify operations, a minimum spanning tree of an undirected graph is formed using the Kruskal algorithm to search the optimal transformation path. A transformable wheelchair is designed using the proposed method in a case study.

Keywords: Product design, Funtion transformation, Path planning, Kruskal algorithm.
DOI: https://doi.org/10.14733/cadaps.2022.741-754

1 INTRODUCTION

A technical system with multi-functions can meet diversified user needs in different environments [29]. An effective method is required to design a multi-functional system. A technical system can be divided into two types, one uses the fixed structure, the other applies the transformable configuration. The latter is a resource-saving solution compared to the former one. Different states are decided by configurations of this kind of the system in performing a certain type of functions. Transformation refers to a process in which the technical system changes its morphology in order to perform a certain function or improved function [23]. Osprey aircraft, Satellite antenna, and Swiss army knife are some examples of transformable systems. The first step of development of a transformable technical system is to plan different function states and paths of the state transformation.

Functions of a technical system are designed based on user needs [19]. Functions are performed through interactions of system elements and environments [10]. Diversity needs may come from the existing or new users. The needs include different functions in the same environment, same function working in different environments, or different functions working in different environments, as shown in Figure 1.

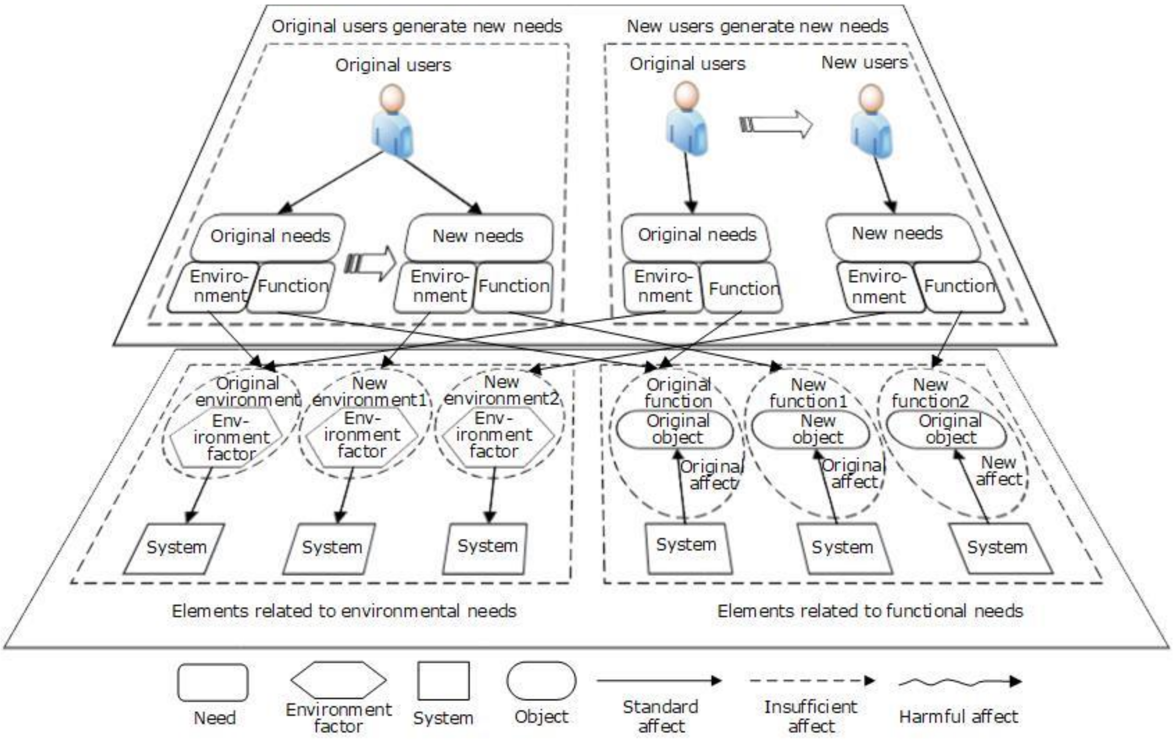

Figure 1: Product needs and system domains.

Environment refers to related elements outside a system, object refers to a target of the system. Changes in user needs can lead to changes of the system and object. Changes of environment and object will make the technical system difficulty working effectively. The system needs to be redesigned to adapt to the change. Therefore, this paper proposes a method to find related environments and objects in design for transformable systems. Based on similarity of related factors, distances of different working states are decided using the Kruskal algorithm to form a search tree with the shortest distance for the optimal transformation path. A transformable wheelchair is designed to verify the proposed method.

2 RELATED REASERCH

Some exiting design methods can be used to design transformable systems. Modular design emphasizes that a product is formed by functional independent modules. The addition, separation, modification, relocation and replacement of modules can realize partial changes of a product [13-14]. Modularization strategies and design methods of the product life cycle have been proposed [9, 11, 16-17, 22]. Modularization can be applied in transformable products to reduce design redundancy. Reconfigurable design forms different design schemes by rearranging and combining product modules [6, 8, 12]. Reconfigurable systems have flexibility in changing the product configuration. Adaptive products provide different functions by changing the form of mechanical systems. These design methods provide guidelines for design of product functions and structures [27, 32]. These methods also provide guides in the design process of transformable systems.

A transformation design method was proposed by Singh et al [27] to consider the transformation principle of a product in the transformation of different states. When there is a single-functional system suitable for design as a transformable system, advantages and disadvantages of two types of systems are discussed [2, 5, 28]. Singh et al. [25] proposed a design framework of the plot-goal-need-capability. T card was designed to record the transformation principle. The card supports the definition and implementation of each principle and operator. The color and shape of the card are different according to relevance of the principle. The combination of mind maps and transformation principles can spread design thinking to develop different transformation products [24]. Transformation principles are used in different fields in the different frequency. After studying a large number of products, the priority of principles can be summarized [28, 30]. Camburn et al. [6] Proposed a design method to improve the quality, novelty and flexibility of the solution. Son et al. [26] studied positive effects of the transformation on the environmental protection. However, there is a lack of a method to plan the transformation path between multiple states of a system.

A graph consisting of points and lines without loops is called a tree. A graph may contain many trees. A tree that can traverse all points of a graph is called a spanning tree. The spanning tree with the minimum weight sum of edges is called the minimum spanning tree [31]. A transformable system can be represented by a graph, in which points represent states and lines represent distances between the states. If each state can be transformed, it will make the system very complex. Therefore, it is very important to find the minimum spanning tree of a transformable system. The minimum spanning tree is one of the most studied problems in the combinatorial optimization. Fast greedy algorithms can be applied in solving this problem, including the Prim algorithm, Kruskal algorithm and Boruvka algorithm [1, 3]. The Kruskal algorithm is one of the faster methods to find the minimum spanning tree in a graph [8, 18]. Kruskal [15] proposed the Kruskal algorithm to solve the problem of travel agents. The basic idea is to choose the edge with the least weight without making up the loop. The Kruskal algorithm is widely used in the circuit system design, campus path planning, etc [4, 20-21]. The minimum spanning tree of a transformable system can be found by the Kruskal algorithm as the optimal transformation path.

3 PROPOSED METHOD

3.1 Distance Between Environment and Object

Figure 2 shows a transformable system with four states, where nodes S1, S2, S3 and S4 represent four states of the system.

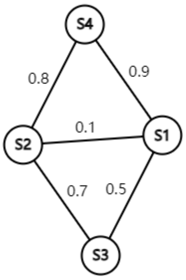

Figure 2: A transformable system with four states.

Edges or connections of nodes represent distance between the nodes. Before planning the minimum spanning tree, we should first find the distance between each state. Application environments and objects of a technical system can be analyzed according to user needs. The environments and objects are combined according to application scenarios to obtain potential states of the system as shown in Figure 3.

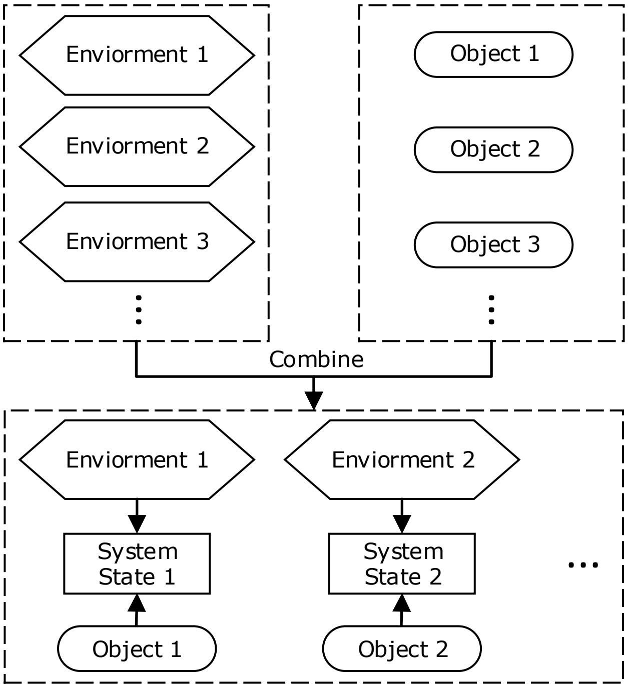

Figure 3: Combination of environments and objects.

Because the system is related to environment and object, the distance between them is used to represent the distance between states. Environments can be represented as follows.

$$E_i = \begin{pmatrix} e_{i1} & f_{i1} & v_{i1} \\ \vdots & \vdots & \vdots \\ e_{ij} & f_{ij} & v_{ij} \\ \vdots & \vdots & \vdots \end{pmatrix} \tag{3.1}$$

where E_i is the i_{th} environment in which the technical system works, e_{ij} is the j_{th} environmental factor of E_i, f_{ij} is the feature of e_{ij}, and v_{ij} is the feature value range. Similarly, the object is expressed as follows.

$$O_j = \begin{pmatrix} o_{j1} & f_{j1} & v_{j1} \\ \vdots & \vdots & \vdots \\ o_{jj} & f_{jj} & v_{jj} \\ \vdots & \vdots & \vdots \end{pmatrix} \tag{3.2}$$

A similar environment with the original system environment will have a close distance between them, which is easy for their transformation. The environment distance has a relation with environment similarity as follows.

$$D(E_i,\ E_j) = 1 - Sim(E_i,\ E_j) \tag{3.3}$$

Where $Sim(E_i, E_j)$ is the similarity between environment E_i and E_j. Similarly, the object distance has a relation with object similarity as follows.

$$D(O_i,\ O_j) = 1 - Sim(O_i,\ O_j) \tag{3.4}$$

As the environment and object are related to the system, the distance of different forms of a system is decided as follows.

$$D(S_i,\ S_j) = \frac{D(E_i,\ E_j) + D(O_i,\ O_j)}{2} \tag{3.5}$$

Environmental factors include different features of a system. Some features of environmental factors will directly affect the system. The similarity of these features represents the similarity of environmental factors. Features include physical, chemical and geometric features, etc [33], as shown in Table 1.

Feature type	Feature name	Feature dimension
Physical	Force	N
	Density	g/cm^3
	• • •	• • •
Chemical	Affinity	L/mol
	Reaction rate	mol/(L·s)
	• • •	• • •
Geometric	Volume	cm^3
	Length	cm
	• • •	• • •
• • •	• • •	• • •

Table 1: Classification of features.

If similarity between different features is 0, and r_{ij} is the similarity between features f_i and f_j, value v_i of feature f_i is in [a, b], and value v_j of feature f_j is in [c, d], the feature similarity can be decided by Equation (3.6).

$$r_{ij} = Sim(f_{ii},\ f_{ij}) \tag{3.6}$$

When $f_{ii} \neq f_{ij},\ r_{ij} = 0.$
When $f_{ii} = f_{ij},$

$(1) v_{ii} \supseteq v_{ij},\ r_{ij} = \dfrac{d-c}{b-a}.$

$(2) v_{ii} \subseteq v_{ij},\ r_{ij} = \dfrac{b-a}{d-c}.$

$(3) v_{ii} \cap v_{ij} \neq 0\ and\ a < c,\ r_{ij} = \dfrac{b-c}{d-a}.$

$(4) v_{ii} \cap v_{ij} \neq 0\ and\ a > c,\ r_{ij} = \dfrac{d-a}{b-c}.$

$(5) v_{ii} \cap v_{ij} = 0,\ r_{ij} = 0.$

where $Sim(f_{ii}, f_{ij})$ is the similarity between feature f_{ii} and f_{ij}. The similarity between two environments can be decided as follows.

$$Sim(E_i, E_j) = \frac{\sum r_{ij}}{N} \tag{3.7}$$

where N is the sum of feature numbers in two environments. Similarly, the similarity of different objects can be decided by Equation (3.8).

$$Sim(O_i, O_j) = \frac{\sum r_{ij}}{N} \tag{3.8}$$

The distance between two system states can then be decided by Equations (3.1)-(3.8).

3.2 Optimal Transformation Between Different States

A graph G(S, D) can be constructed using the system state as node S, and the distance between states as edge D. The minimum spanning tree can be found by the Kruskal algorithm as shown in Figure 4 [31].

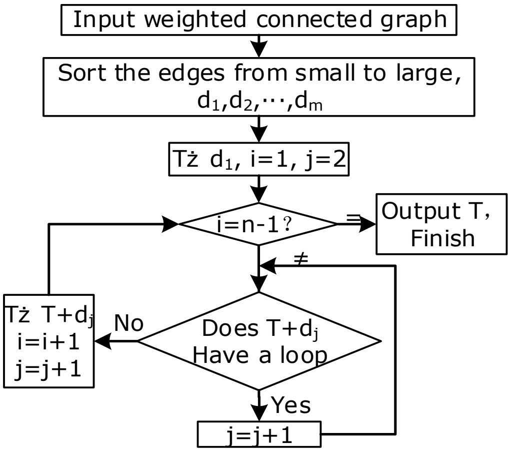

Figure 4: Flowchart of the Kruskal algorithm.

Steps of the Kruskal algorithm are as follows.

1) Sort edges from small to large, d_1, d_2, …, d_m.
2) $T \leftarrow d_1$, i=1, j=2.
3) If i=n-1, output T, and the algorithm ends, otherwise, go to 4).
4) If d_j is added to T, a loop is formed, then j=j+1, go to 3). Otherwise go to 5).
5) $T \leftarrow T+d_j$, i=i+1, j=j+1, go to 3).

An optimal transformation path can be found to traverse all states with the minimum distance. The technical system is evaluated by the idealization level as follows [33].

$$I = \frac{\sum U}{\omega_1 \sum H + \omega_2 \sum C} \tag{3.9}$$

where *I* is the degree of idealization. A large *I* indicates the high value of a technical system. *U* is the benefit of the technical system. In this paper, it is expressed by the number of environments and objects that the system can adapt. *H* is the harmful function of the system. In this paper, it is expressed by the failure index. Key state has a high failure index because it is related to more than one other state and used frequently. Therefore, the failure index is expressed by the proportion of key states. *C* is cost of the technical system based on the normalized design complexity which is expressed by the distance between states. ω_1 and ω_2 represent the weight of *H* and *C*, which is determined by designers according to different situations.

In Figure 1, the change of environments and objects may cause their relationship change for the system from a standard affect to a harmful affect. Altshuller proposed 76 standard solutions in TRIZ [33] to be specially used to solve the deficiency or harmful affect between system factors. The application process of the standard solution can assist in the configuration design for each state of the system.

4 CASE STUDY

The wheelchair is considered as a technical system in the case study. Users are adults with mobility difficulties. The main environment is outdoor in normal weather conditions. A wheelchair model is shown in Figure 5.

Figure 5: 3D model of the wheelchair.

In Figure 5, the environment element associated with the system is a smooth road, which is characterized by the height of obstacle. The object element associated with the system is adult, which is characterized by the horizontal displacement, supporting capacity, supporting area and displacement of thigh. Environments of the wheelchair are represented in Equation (4.1).

$$E_1 = \textit{Smooth road} \quad \textit{Height of obstacle} \quad cm \quad [0-3] \tag{4.1}$$

Objects of the wheelchair are as follows.

$$O_1 = \begin{pmatrix} \textit{Adult} & \textit{Horizontal displacement} & m & [0-\infty] \\ \textit{Adult} & \textit{Supporting capacity} & kg & [0-90] \\ \textit{Adult} & \textit{Supporting area} & m^2 & [0.1-0.3] \\ \textit{Adult} & \textit{Displacement of thigh} & cm & [0-5] \end{pmatrix} \tag{4.2}$$

Potential environments and objects can be obtained according to diverse user needs. Features and values of potential environments are listed in Table 2.

Features	E_1	E_2	E_3	E_4	E_5	E_6	E_7
Height of obstacle (cm)	[0-3]	[0-30]	[0-3]	[0-3]	[0-1]	[0-1]	[0-1]
Rainfall (mm/h)				[2-10]			
Flow rate (L/min)							[0-6]
UV index			[5-15]				
Wind velocity (m/s)				[0-20]			

Table 2: Features and values of different environments.

In Table 2, environment conditions are normal (E_1), step room (E_2), outdoor with strong sunshine (E_3), outdoor with wind and rain (E_4), indoor (E_5), toilet (E_6), and shower room (E_7). Features and values of potential objects are shown in Table 3.

Features	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}
Distance of horizontal movement (m)	[0-∞]	[0-∞]						[0-∞]	[0-∞]	[0-∞]	[0-∞]	[0-∞]
Weight to support (kg)	[0-90]	[0-90]	[0-90]	[0-90]	[0-90]	[0-90]	[0-90]	[0-60]	[0-90]	[0-90]	[0-90]	[0-90]
Supporting area (m^2)	[0.1-0.3]	[0.1-0.3]	[0.1-0.3]	[0.11-0.31]	[0.09-0.29]	[0.05-0.15]	[0.2-0.5]	[0.05-0.2]	[0.1-0.3]	[0.15-0.35]	[0.2-0.5]	[0.1-0.5]
Length of thigh displacement (cm)	[0-5]	[0-5]	[0-5]	[0-5]	[0-5]		[0-5]	[0-5]	[0-5]	[0-5]	[0-5]	[0-5]
Distance of vertical displacement (m)		[0-∞]										
Distance of tableware displacement (cm)				[0-1]								
Distance of axillary displacement (cm)						[0-2]						
Distance of shank displacement (cm)							[0-5]			[0-2]	[0-5]	
Distance of back displacement (cm)							[0-5]				[0-5]	
Distance of head displacement (cm)							[0-5]				[0-5]	
Distance of goods displacement (cm)												[0-5]

Table 3: Features and values of different objects.

Objects in Table 3 include adults travel (O_1), adults up and down stairs (O_2), adults sitting (O_3), adults for meal (O_4), adults in toilet (O_5), adults in shower (O_6), adults in rest (O_7), children with mobility difficulties (O_8), adults with upper limb injuries (O_9), adults with lower extremity injuries (O_{10}), comatose adults (O_{11}), and goods (O_{12}). Through combining environments and objects, a total of 21 potential states are obtained in Table 4.

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}
E_1	S1							S2	S3	S4	S5	S6
E_2		S7										
E_3								S8	S9	S10	S11	S12
E_4								S13	S14	S15	S16	S17
E_5			S18	S19								
E_6					S20							
E_7						S21						

Table 4: Combination of potential environments and objects of the wheelchair.

Distances between states are calculated by Equations (3.1)-(3.8) as shown in Table 5. Table 5 shows the adjacency matrix of 21 states of the wheelchair.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21
S1	0	0.12	0.00	0.14	0.27	0.24	0.55	0.37	0.25	0.39	0.52	0.49	0.49	0.37	0.52	0.65	0.61	0.46	0.55	0.47	0.76
S2		0	0.12	0.22	0.31	0.32	0.61	0.25	0.37	0.47	0.56	0.57	0.38	0.49	0.59	0.69	0.69	0.58	0.64	0.57	0.74
S3			0	0.14	0.27	0.25	0.55	0.37	0.25	0.39	0.52	0.50	0.49	0.38	0.52	0.65	0.63	0.46	0.55	0.47	0.76
S4				0	0.23	0.29	0.59	0.47	0.39	0.25	0.48	0.54	0.59	0.52	0.38	0.60	0.67	0.58	0.61	0.58	0.79
S5					0	0.28	0.75	0.56	0.52	0.48	0.25	0.58	0.69	0.65	0.60	0.38	0.71	0.68	0.69	0.77	0.81
S6						0	0.65	0.57	0.50	0.54	0.58	0.25	0.69	0.63	0.67	0.71	0.38	0.59	0.63	0.59	0.73
S7							0	0.64	0.58	0.62	0.77	0.68	0.65	0.59	0.63	0.79	0.69	0.69	0.75	0.70	0.81
S8								0	0.12	0.22	0.31	0.32	0.38	0.49	0.59	0.69	0.69	0.66	0.72	0.65	0.83
S9									0	0.14	0.27	0.25	0.49	0.38	0.52	0.65	0.63	0.54	0.63	0.55	0.84
S10										0	0.23	0.15	0.59	0.52	0.38	0.60	0.67	0.66	0.69	0.66	0.88
S11											0	0.33	0.69	0.65	0.60	0.38	0.71	0.72	0.77	0.85	0.90
S12												0	0.69	0.63	0.67	0.71	0.38	0.67	0.71	0.67	0.82
S13													0	0.12	0.22	0.31	0.32	0.70	0.76	0.70	0.84
S14														0	0.14	0.27	0.25	0.59	0.70	0.60	0.84
S15															0	0.23	0.29	0.70	0.74	0.70	0.89
S16																0	0.33	0.80	0.82	0.89	0.91
S17																	0	0.71	0.75	0.72	0.83
S18																		0	0.14	0.03	0.73
S19																			0	0.15	0.76
S20																				0	0.72
S21																					0

Table 5: Distances between different states of the wheelchair.

A undirected graph is drawn to represent the initial transformation path by taking states as points and distances as lines, as shown in Figure 6.

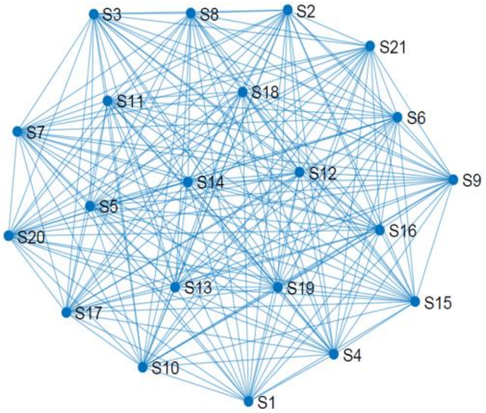

Figure 6: The initial transformation path of wheelchair states.

If all the connected states in Figure 6 can be transformed each other, the system will be very complex and difficult to design. Edges in Figure 6 are sorted by numerical values, d_1=0.01, d_2=0.03, d_3=0.115, ..., d_{210}=0.91. Let d_1 be the initial tree T, and edges from d_2 to d_{210} are added to T in the order from small to large. When there is a loop in T after adding d_i, d_i is deleted, d_{i+1}, is continuously added until all the edges have been tested, then T is the minimum spanning tree. This process can be completed by running a computer program such as codes in MATLAB. The minimum spanning tree in Figure 6 can be searched using the Kruskal algorithm for the optimal transformation path as shown in Figure 7.

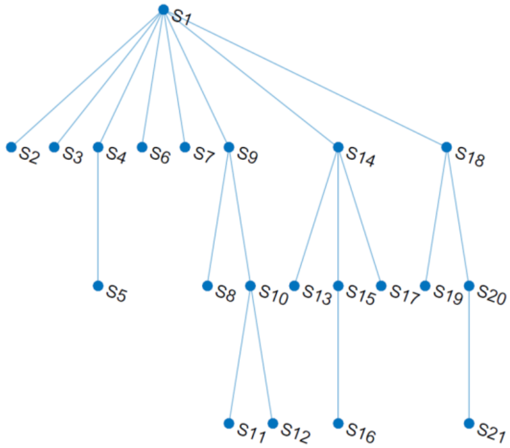

Figure 7: The optimal transformation path of wheelchair states.

The tree in Figure 7 is searched for the optimal transformation path, in which nodes represent 21 states of the wheelchair. Connected lines indicate that these states can be transformed each other. The environment of the wheelchair has been changed from E_1 to E_3. The environmental factor ultraviolet in E_3 has a harmful effect on adults. According to the application process of the standard solution, the 9th standard solution can be used, which is: if there is a harmful effect between two substances, a third substance can be introduced to eliminate the harmful effect [33]. The solution is to install a shield on the wheelchair to block ultraviolet rays, as shown in Figure 8.

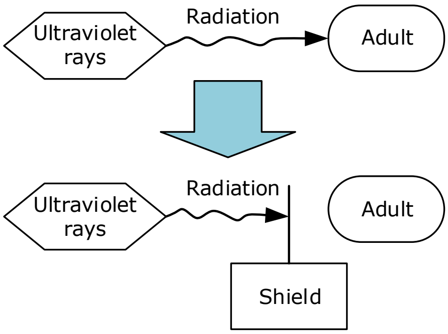

Figure 8: Application of a standard solution to eliminate harmful effects.

According to the application of standard solution in Figure 8. A 3D model of the wheelchair in environment E_3 is shown in Figure 9.

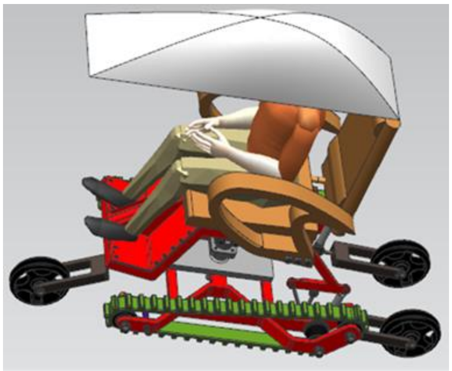

Figure 9: Wheelchair in environment E_3.

Following the above method, each state of the transformable wheelchair is designed using standard solutions shown in Figure 10.

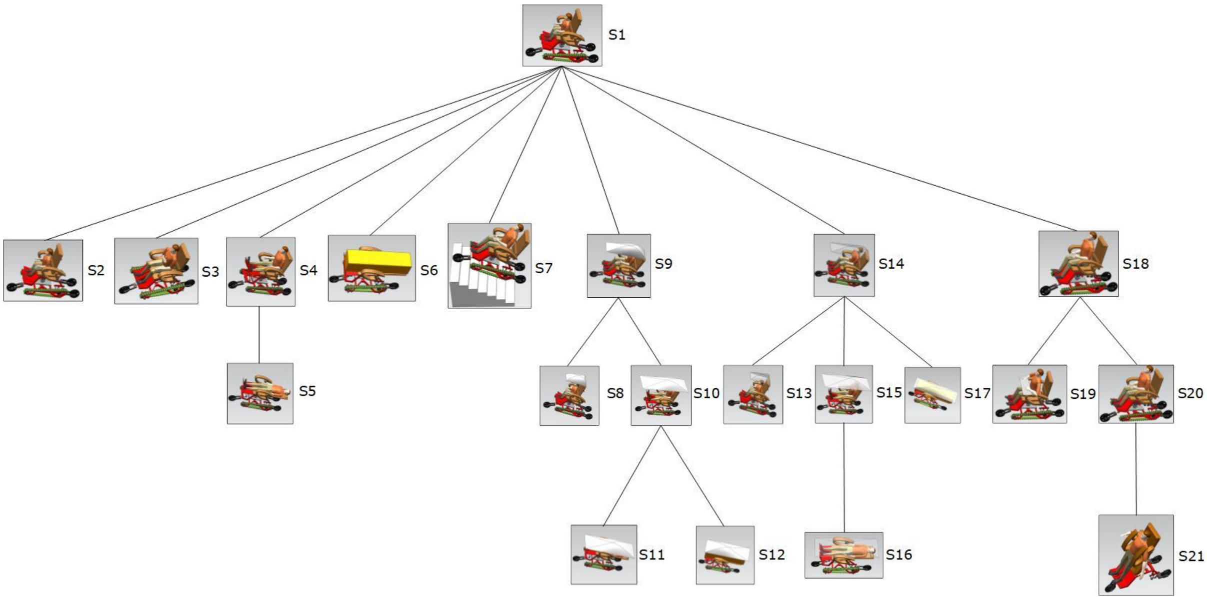

Figure 10: Wheelchair model on optimal transformation path.

States S8, S9, S10, S11 and S12 in Figure 10 are also obtained according to the 9th standard solution. As the Kruskal algorithm is a greedy algorithm, the solution obtained in each step will be better than the previous step. The optimal transformation path is therefore unique. The designer does not need to choose the optimal path from multiple paths.

The design result is a transformable wheelchair with 21 working states. If the transformation path is not planned, it will make the design complex and difficult to determine the key working state. This research mainly focuses on planning the optimal path. The idealization level of the system designed according to two paths in Figures 6 and 7 is evaluated to verify the effectiveness of the proposed method. Evaluation indexes of the system are used in the solution evaluation shown in Table 6.

Evaluation indexes	Initial transformation path	Optimal transformation path	Weight
Number of environments	7	7	
Number of objects	12	12	
Failure index	1	0.38	ω_1=0.4
Normalized design complexity	0.54	0.02	ω_2=0.6

Table 6: The evaluation indexes of the wheelchair.

The idealization degree of the system is evaluated by Table 6 and Equation (3.9). The idealization degree of system I_B according to the initial transformation path is as follows.

$$I_B = \frac{\sum U}{\omega_1 \sum H + \omega_2 \sum C} = \frac{7+12}{0.4\times 1 + 0.6\times 0.54} = 26.24 \tag{4.3}$$

The idealization degree of system I_A according to the optimal transformation path is as follows.

$$I_A = \frac{\sum U}{\omega_1 \sum H + \omega_2 \sum C} = \frac{7+12}{0.4\times 0.38 + 0.6\times 0.02} = 115.85 \tag{4.4}$$

Obviously, $I_A > I_B$, the system based on the optimal transformation path is more effective.

5 CONCLUSIONS

A transformation product designed based on the optimal transformation path can save resources and simplify product operations. In this research, potential environments and objects are determined for diversity needs of product users. Distances between potential states of the system are obtained based on similarity of environments and objects. The optimal transformation path of states is planned to save resources and simplify transformation operations. A transformable wheelchair is designed to illustrate the effectiveness of the proposed method. Further work will apply the proposed method in design of different transformable products for the method improvement.

ACKNOWLEDGEMENTS

This research is sponsored by the National Innovation Method Work Special Project NO.2019IM020200.

Jinpu Zhang, http://orcid.org/0000-0002-6813-8466
Guozhong Cao, http://orcid.org/0000-0002-2939-5566
Qingjin Peng, http://orcid.org/0000-0002-9664-5326
Runhua Tan, http://orcid.org/0000-0002-6797-8199
Huangao Zhang, http://orcid.org/0000-0001-8945-0056

REFERENCES

[1] Bergantiños, G.; Lorenzo, L.; Lorenzo, S.: A generalization of obligation rules for minimum cost spanning tree problems, European Journal of Operational Research, 211(1), 2011, 122-129. https://doi.org/10.1016/j.ejor.2010.10.035

[2] Bortolini, M.; Galizia, F.-G.; Mora, C.: Reconfigurable manufacturing systems: Literature review and research trend, Journal of Manufacturing Systems, 49(1), 2018, 93-106. https://doi.org/10.1016/j.jmsy.2018.09.005

[3] Broutin, N.; Devroye, L.; McLeish, E.: Note on the Structure of Kruskal's Algorithm, Algorithmica, 56(1), 2010, 141-159. https://doi.org/10.1007/s00453-008-9164-4
[4] Çakır, E.; Ulukan, Z.: A Hybrid Kruskal's Algorithm Based on Intuitionistic Fuzzy with Hamacher Aggregation Operator for Road Planning, International Journal of Fuzzy Systems, 2021, 1-14. https://doi.org/10.1007/s40815-020-01007-4
[5] Camburn, B.; Guillemette, J.; Crawford, R.; Wood, KL.; Jensen, DJ.; Wood, JJ.: When to transform? Development of indicators for design context evaluation, International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, Quebec, Canada, 5(1), 2010, 249-266. https://doi.org/10.1115/DETC2010-28951
[6] Camburn, B.; Wood, K.; Crawford, R.; Robbens, J.; Jensen, D.; Patel, A.: Advances in transformational design: Correlating context evaluation to quality feasibility and novelty, International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, Illinois, USA, 7(1), 2012, 447-460. https://doi.org/10.1115/DETC2012-71416
[7] Carbonari, L.; Callegari, M.; Palmieri, G.; Palpacelli, M.-C.: A new class of reconfigurable parallel kinematic machines, Mechanism and Machine Theory, 79(1), 2014, 173-183. https://doi.org/10.1016/j.mechmachtheory.2014.04.011
[8] Devi, M.-M.; Geethanjali, M.: Hybrid of Genetic Algorithm and Minimum Spanning Tree method for optimal PMU placements, Measurement, 154, 2020, 107476. https://doi.org/10.1016/j.measurement.2020.107476
[9] Gadalla, M.; Xue, D.: An extended analytic network process method for optimal design of reconfigurable products considering dependency relations among descriptions of design/process candidates and evaluation criteria, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 234(24), 2020, 4763-4779. https://doi.org/10.1177/0954406220932606
[10] Gauss, L.; Lacerda, D.-P.; Sellitto, M.-A.: Module-based machinery design: a method to support the design of modular machine families for reconfigurable manufacturing systems, The International Journal of Advanced Manufacturing Technology, 102(9-12), 2019, 3911-3936. https://doi.org/10.1007/s00170-019-03358-1
[11] Gero, J.-S.: Design Prototypes: A Knowledge Representation Schema for Design, AI Magazine, 11(4), 1990, 26-36. https://doi.org/10.1609/aimag.v11i4.854
[12] Inoue, M.; Yamada, S.; Miyajima, S.; Ishii, K.; Hasebe, R.; Aoyama, K.; Yamada, T.; Bracke, S.: A modular design strategy considering sustainability and supplier selection, Journal of Advanced Mechanical Design, Systems, and Manufacturing, 14(2), 2020, JAMDSM0023. https://doi.org/10.1299/jamdsm.2020jamdsm0023
[13] Jung, S.; Simpson, T.-W.: New modularity indices for modularity assessment and clustering of product architecture, Journal of Engineering Design, 28(1), 2017, 1 – 22. https://doi.org/10.1080/09544828.2016.1252835
[14] Kashkoush, M.; Elmaraghy, H.: Designing modular product architecture for optimal overall product modularity, Journal of Engineering Design, 28(5), 2017, 293 – 316. https://doi.org/10.1080/09544828.2017.1307949
[15] Kruskal, J.-B.: On the shortest spanning subtree of a graph and the traveling salesman problem, Proceedings of the American Mathematical society, 7(1), 1956, 48-50. https://doi.org/10.2307/2033241
[16] Li, Q.: The optimal multi-period modular design with fairness concerns, International Journal of Production Economics, 206, 2018, 233-249. https://doi.org/10.1016/j.ijpe.2018.10.007
[17] Martinez, M.; Xue, D.-A.: A modular design approach for modeling and optimization of adaptable products considering the whole product utilization spans, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 232(7), 2018, 1146-1164. https://doi.org/10.1177/0954406217704007

[18] Masoumeh, Z.; Mohammad, R.; Mollakhalili, M.; Kamal, M.: A rapid learning automata-based approach for generalized minimum spanning tree problem, Journal of Combinatorial Optimization, 40(1), 2020, 636-659. https://doi.org/10.1007/s10878-020-00605-0

[19] Miles, L.-D.: Techniques of value analysis and engineering, Miles Value Foundation, Washington DC, 2015.

[20] Petrică, C.-P. The generalized minimum spanning tree problem: An overview of formulations, solution procedures and latest advances. 2020, 283(1):1-15. https://doi.org/10.1016/j.ejor.2019.05.017

[21] Sarkar, D.; De, A.; Chanda, C.-K.; Goswami, S.: Kruskal's maximal spanning tree algorithm for optimizing distribution network topology to improve voltage stability, Electric Power Components and Systems, 43(17), 2015, 1921-1930. https://doi.org/10.1080/15325008.2015.1062818

[22] Sheng, Z.; Li, Y.; Wu, L.; Xie, H.: Lifecycle-oriented product modular design of CNC machine tools, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2017, 231(11): 1981-1994. https://doi.org/10.1177/0954406215625679

[23] Singh, V.; Skiles, S.-M.; Krager, J.; Wood, K.-L.; Jensen, D.; Sierakowski, R.: Innovations in design through transformation: A fundamental study of transformation principles, Journal of Mechanical Design, 131(8), 2009, 081010. https://doi.org/10.1115/1.3125205

[24] Singh, V.; Walther, B.; Krager, J.; Putnam, N.; Koraishy, B.; Wood, K.; Jensen, D.: Design for transformation: theory, method and application, Proceedings of the ASME International Design Engineering Technical Conference & Computers and Information in Engineering Conference, Las Vegas, NV, USA, 2007, 447-459. https://doi.org/doi:10.1115/DETC2007-4876

[25] Skiles, S.-M.; Singh, V.; Krager, J.; Seepersad, C.-C.; Wood, K.-L.; Jensen, D.: Adapted Concept Generation and Computational Techniques for the Application of a Transformer Design Theory, Proceedings of the ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Philadelphia, Pennsylvania, USA, 2006, 951-965. https://doi.org/10.1115/detc2006-99584

[26] Son, J.-J.; Shu, L.-H.: The mechanical transformation and environmentally conscious behavior—ERRATUM, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 29(1), 2015, 129-129. https://doi:10.1017/S0890060414000341

[27] Tu, Q.; Deng, Y.-M.; Nee, A.-Y.-C; Lu, W.: Functional upgrading fundamental and performance upgrading priority ranking of adaptable-function products, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 233(6), 2019, 2135-2148. https://doi.org/10.1177/0954406218780518

[28] Weaver, J.-M.: Transformer Design: Empirical Studies of Transformation Principles, Facilitators, and Functions, MS.D.Thesis, University of Texas at Austin, Austin, TX, USA, 2008.

[29] Weaver, J.; Wood, K.; Crawford, R.; et al. Transformation design theory: a meta-analogical framework[J]. Journal of Computing and Information Science in Engineering, 10(3), 2010, 031012. https://doi.org/10.1115/1.3470028

[30] Weaver, J.; Wood, K.-L.; and Jensen, D.: Transformation Facilitators: A Quantitative Analysis of Reconfigurable Products and Their Characteristics, International Design Engineering Technical Conferences, Brooklyn, NY, 2008, 351-366. https://doi.org/10.1115/DETC2008-49891

[31] West, D.-B.: Introduction to graph theory, Prentice Hall, Upper Saddle River, NJ, 1996.

[32] Yang, P.; Deng, Y.-M.: Evaluating behavioral process similarity of mechanical products in redesign for functional adaptability, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 231(12), 2017, 2211-2222. https://doi.org/10.1177/0954406216631576

[33] Zhang, H.: Innovative Design-Systematic Innovation Based on TRIZ, Machinery Industry Press, Beijing, 2017.