28 ✓

Investigation of Design Heuristics for Pruning the Number of Solutions

Y. C. Liu[1], A. Chakrabarti[2]

[1]Department of Industrial Design Department, University of Chang Gung University, Tao-Yuan, Taiwan
(ycl30@mail.cgu.edu.tw)

[2] Centre for Product Design and Manufacturing, Indian Institute of Science, Bangalore
(ac123@cpdm.iisc.ernet.in)

Abstract – **It is important to generate a broad range of concepts so that better or optimal concepts will be not be overlooked. Therefore, a systematic search of solution space has been developed with the use of computational tools. However, the problem of using systematic search is that too many alternatives are produced to be seriously considered by the designers.**

This paper addresses the issue of managing a solution space using domain knowledge as its basis in both theoretical and practical manners. Based on gaining an understanding of solutions in a particular domain, design heuristics are derived, by investigating the solution's functional, spatial, and physical attributes. This is to decide on similarity and feasibility of solutions in the solution space so that unnecessary exploration of these can be avoided at the earliest moment. The proposed methods lead to manage the number of solutions to be explored by the designer.

Keywords – Computer-aided conceptual design, synthesis, design heuristics, mechanism, similarity

I. INTRODUCTION

The issue of computer-aided conceptual design has been explored in many domains, such as in dynamic systems [1], mechatronics [2], mechanical designs [3,4], and structural design [5,6]. For such systems, one advantage is that concepts can be systematically or exhaustively generated, therefore some concepts that the designers would have overlooked can be reminded from the solutions generated by the systems. These concepts are generated by using a set of building blocks; each concept is composed of different types and numbers of building blocks. The idea is to offer these concepts to designers for exploration, modification and evaluation, before selecting promising ones for further development.

However, a problem of generating a large number of solutions is that designers in fact are unable to manually explore in a meaningful way. This raises a significant issue: how to reduce the number of solutions at the earliest possible moment without compromising the richness of the solution space.

There are two main approaches to manage the number of solutions in a solution space: to use feasibility, or similarity.

In the feasibility-based approach, designers choose a number of possible constraints with which to discard infeasible solutions. These can be based on domain-neutral knowledge, such as to limit the number of building blocks used for synthesis. Examples of applying domain-neutral knowledge for discarding infeasible solutions are such as [7] formulated (1) no more than n repeated basic elements are allowed in a solution chain and (2) a certain basic element is constrained to be at a specific place in the solution chain. Similar rules are seen that limit the size of the solution space by specifying an expected number by means of a set of rules. Constraints are defined based on the management of a large solution space rather than on engineering knowledge. Therefore, valuable solutions might well be missed when applying these heuristics.

In the similarity-based approach, constraints can be developed based on domain-based knowledge. A number of rules of similarity of solutions are used to partition the solution space into groups of solutions. Designers only need to investigate one or a few solutions in each group in order to develop an overview of the complete solution space rather than having to investigate each individual solution. The concept of similarity has been used extensively for developing databases of designs or building blocks such as, catalogue of elements [8], and catalogue of solutions [9], or for developing methods for generating designs such as analogical reasoning [10], for morphological chart [11]. However, its use for managing a solution space is few.

In this paper, we do not discuss the feasibility-based part of the above work, since that is along the lines of work taken earlier and elsewhere. The focus of this paper is the similarity-based approach, using domain knowledge, for managing (pruning or grouping) solutions generated by a compositional synthesis process. This paper develops a set of design heuristics to prune solutions, based on the developed approach, i.e., the functional synthesis approach. These design heuristics are rules to group concepts at the higher level (e.g. at the functional level) of solution abstraction, based on various degree of similarity between solutions. Various levels of similarity are proposed based on the same working principle, temporary duration of action, spatial arrangement, and motion among solutions. The effectiveness of each design heuristic is also discussed.

The rest of this paper is structured as: Section II describes the functional synthesis approach; Section III proposes various heuristics based on similarity; Section IV shows the method; and Section V is the discussion; and finally, concludes this paper in Section VI

.

II. THE FUNCTIONAL SYNTHESIS APPROACH

The functional synthesis approach (see [12-16] for further description) is developed based on the following two principles:

Principle 1: Concepts are generated by combining a set of functional building blocks (called basic elements). A wide variety of existing designs (such as gearbox designs with various numbers and types of gears) do seem to have common elements. Therefore, if these elements can be distilled, and rules are extracted to combine in a reasonable way, various designs could be generated.

Principle 2: Concepts are generated to fulfill a small subset of function requirements, and then these concepts are modified and detailed so as to meet all the functional requirements. Infeasible ideas failing to meet the rest of the requirements are discarded.

Basic elements used in this approach are classified based on (1) the input and output kind (force or motion), (2) the directional relationships between the input and output (e.g. the directional relationships is parallel to and non-intersecting to each other), and (3) working principle. Some basic elements are listed in Fig. 1. For instance, Tierod1 basic element represents all sorts of connectors which transmit and channel the input translation to the output translation, with the condition that the channel between input and output is parallel to the direction of input and output. Likewise, Tierod2 basic element represents connectors which transverse input translation, with the condition that the channel between input and output is orthogonal to the direction of input and output.

Basic Elements	**Tierod1**	**Tierod2**	**Crank1**	**Crank2**	**Wedge**
I-O kind	F-F	F-F	T-F	F-T	F-F
Basic Elements	**Screw**	**Lever1**	**Lever2**	**Cam1**	**Shaft**
I-O kind	T-F	F-F	F-F	T-F	T-T

F: force (translation), T: torque (rotation), I/O: input/output

Fig 1 The entire set of elements used by the approach in synthesizing concepts

A. The rules for the combination of basic elements

The rules for the combination of basic elements are that the interconnection between two elements must meet kind connections, i.e. force-to-force, and torque-to-torque. For example, the combination of (Tierod1 Tierod1) is true because the interconnection between these two elements meet a force-to-force connection. Another example, the combination of (Crank2 Crank1) is false because the interconnection is a force-to-torque which fail to connect.

Three levels of solution abstraction are generated in this approach. These are topological, spatial, and generic physical. Fig. 2 illustrates the representation of a pair of spur gears at these three solution abstraction levels.

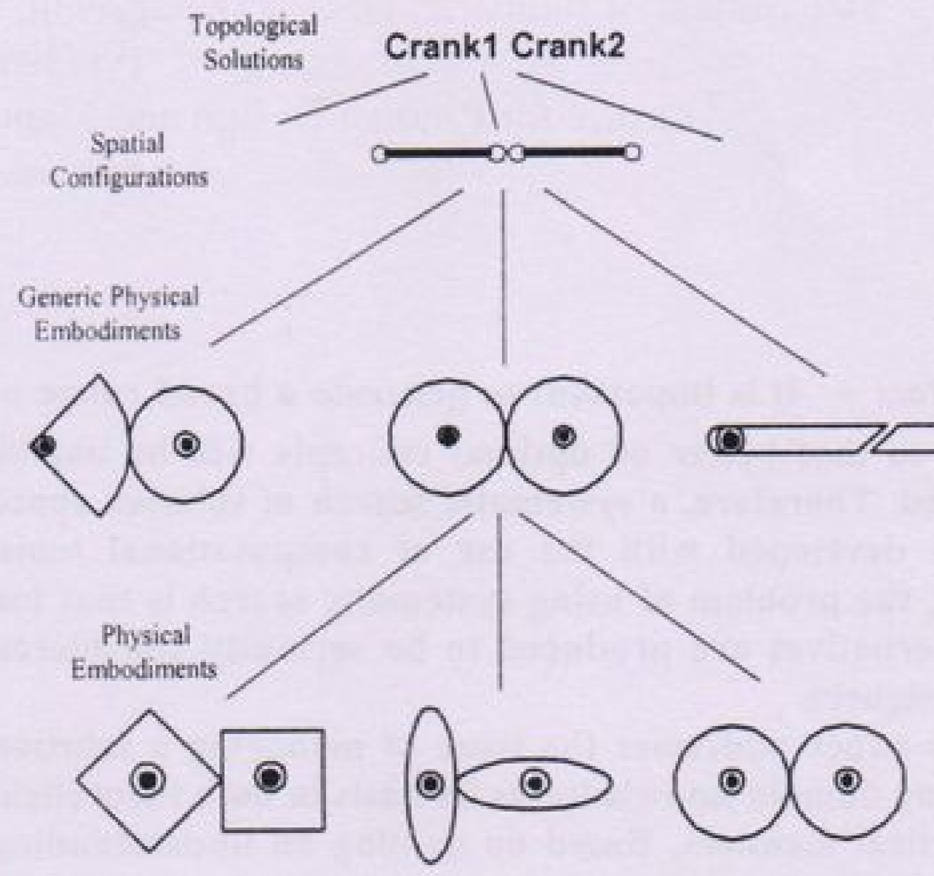


Fig 2 Four solution abstractions for a pair of spur gears.

B. Combinatorial synthesis

This is a process to synthesize an exhaustive set of solutions which satisfy the input-output motion requirement. These solutions are combinations of basic elements, interconnected by motion connections. All possible topological solutions with a different number, type, and compositional order of basic elements are considered by using kind synthesis.

Each resulting solution is a causal chain of operators (i.e., basic elements) connecting the given input kind to the desired output kind. For instance, if the solution is (Wedge Wedge Crank2), it transfers an input force using a Wedge basic element, the second Wedge basic element taking the force and transferring it into an output force, and finally it is transformed by a Crank2 basic element into an output torque. The number of generated solutions is determined by (1) the input and output kind requirement derived from the characteristics of the problem, (2) the number and variety of basic elements and (3) the maximum number of basic elements that can be used for the construction of each solution.

For example, if a design problem is given to transform an input translation to an output translation with a database of Tierod1 and Tierod2 elements, with the condition that all possible solutions have at most two basic elements in each solution, six solutions can be generated; these are (Tierod1), (Tierod2), (Tierod1 Tierod2), (Tierod2 Tierod1), (Tierod1 Tierod1), and (Tierod2 Tierod2). The procedure for the combinatorial synthesis is as follows:

- In the database library, each basic element containts the knowledge of its specific input-output motion.
- Select the basic elements from the library, and decide the maximum allowable number of basic elements in each solution.

- Specify the input and output requirement in terms of the input and output motion.
- Search for all possible combinations of the basic elements which satisfy the interconnectivity rule and the requirement.

Take a design problem as an example in which the input to output is a force to a torque, used a database of 10 basic elements: Tierod1, Tierod2, Wedge, Shaft, Crank1, Crank2, Lever1, Lever2, Cam1, and Screw with that all solutions use up to 3 of these elements. The total solution number is 41

1. Theoretical Investigation of Kind Synthesis

Suppose a single-input-single-output (SISO) design problem is expressed as the following transformation:

$$a \to b$$

Where a is the perceived input kind and b is the desired output kind. Let this problem be solved by using the maximum allowable basic elements to be n. This is equivalent to forming chains of basic elements, whose length should be in the range between 1 and n, including 1 and n. The number of basic elements in the database is k.

For exhaustive search, the number of possible solutions depends on the characteristics of these k elements. This is because inside each solution chain, the consecutive elements of the list are connected end to end, and must meet the inter-connectivity rule. Additionally, the beginning of the first element and the end of the last element of this solution chain must have the input kind a, and the output kind b respectively. The number of solutions is between a theoretical lowest and highest bound. This is expressed in the following equation:

$$0 \leq \text{number} \leq (\mathrm{k}^{n} + k^{n-1} + k^{n-2} + k^{n-3} + \ldots + k) \quad (1)$$

Considering the situation where the number of solutions is equal to the highest bound $(\mathrm{k}^{n} + k^{n-1} + k^{n-2} + k^{n-3} + \ldots + k)$, any two of these k basic elements in any consecutive order meet the inter-connectivity rule, and any elements can be placed as the first or last element in each solution chain. Considering the situation that the solution number is equal to the lowest bound, no two of these k basic elements in any order meet the rule, nor can any elements be placed as the first or last element in each solution chain.

Theoretically, if we can make the highest bound smaller (i.e. to decrease the highest bound), this will help to cope with the issue of 'too many solutions'. The following heuristics considering the similarity and infeasibility of solutions, based on the specialized circumstances which are relevant to characteristics of the selected basic elements for synthesis, will permit the replacement of the theoretical highest bound in "(1)" with a smaller value.

2. Detailing topological solutions

Having generated solutions in the topological level, the next step is to generate spatial variations for them. Each resulting spatial configuration is an abstract representation of 3D physical embodiments which would be oriented in space at a certain instant. The procedure for generating spatial variations is that each basic element has different spatial orientations and I-O directions, considering combinations of all possibilities of each basic element in a topological solution, its alternative spatial configurations, which meet the inter-connectivity requirements, are generated.

C. Embodying spatial configurations

After generating spatial configurations of a topological solution, the next step is to embody these spatial configurations. Variations in terms of generic shapes and interfaces are generated for each spatial configuration, with the condition that internal elements in between must meet the inter-connectivity requirements. This contains the consideration of all possible generic physical shapes for each spatial orientation in a spatial configuration, and ensuring appropriate interfaces between connecting objects.

III. TYPES OF SIMILARITY

Three types of similarity are defined below:

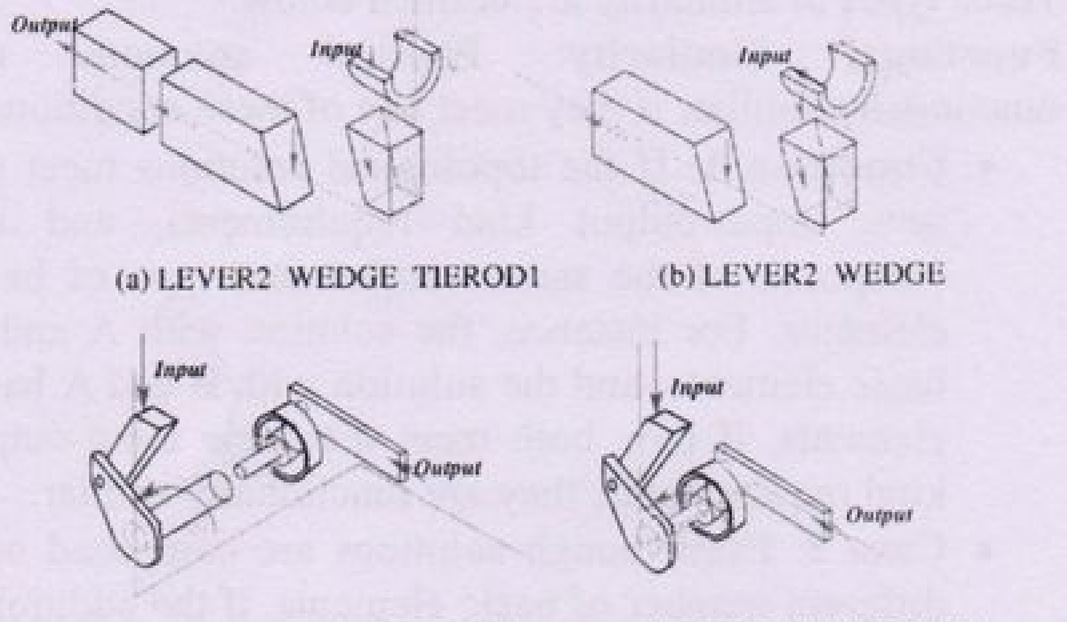
Functional Similarity: Physical solutions are functionally similar, if they meet one of these conditions:

- **Condition 1**: If the topological solutions meet the same input-output kind requirements, and are composed of the same number and type of basic elements. For instance, the solution with A and B basic elements, and the solution with B and A basic elements, if they both meet the same input-output kind requirements, they are functionally similar.
- **Case 2**: Even though solutions are composed of a different number of basic elements, if the additional basic elements have their working principles only for channeling or transmitting the input kind requirement, they are functionally similar. For instance, the solution with A, B, and B basic elements, and the solution with A and B basic elements are functionally similar, if they both meet the same input-output kind requirements, and B basic element is used only for channeling or transmitting the input kind requirement. However, the solution A and B, and the solution A are not functionally similar.

For instance, as shown in Fig. 3, solutions (a) and (b), and (c) and (d) are respectively functionally similar. In (a) and (b), they are mechanisms that convert and enlarge the input (the downward translation) into the output (going left and inside) by means of a lever and a wedge principle. In (c) and (d), they are used to convert a reciprocating translation (going up and down) into a translational output with motion going left and right, by means of a crank and a cam-and-follower principle. Moreover, (a), and (c) are not functionally similar, even though they both are used to enlarge and convert the direction of input, because they are not based on the same physical principle.

- **Spatial Similarity**: Physical solutions are spatially similar, if they are functionally similar, and they have the characteristics which meet the same functional requirements in the aspect of (1) direction, and (2) position. For instance, as shown in Fig. 3, solutions (a) and (b), and (c) and (d) are respectively functionally similar, and they respectively meet the same requirements in the aspect of (1) direction, and (2) position. An example of solutions that are functionally similar but not spatially similar is shown in Fig. 4.

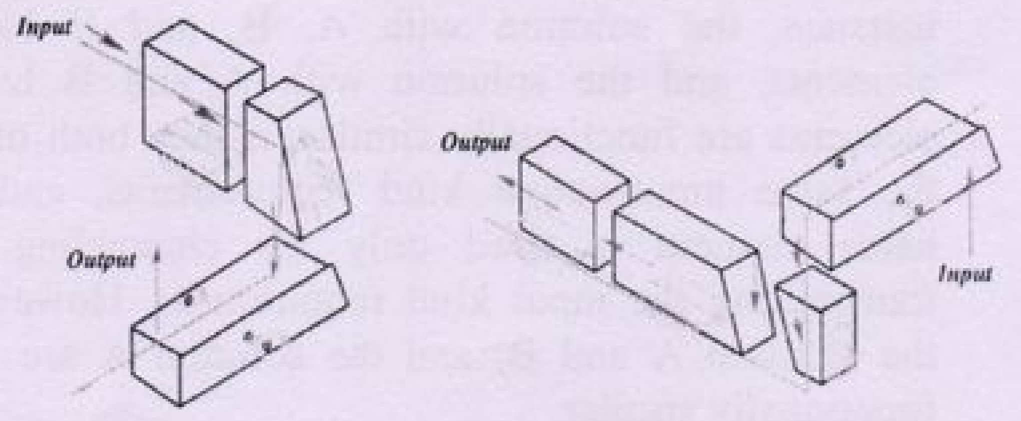
Physical Similarity: Physical solutions are physically similar, if they are spatially similar, and have the characteristics that meet the same functional requirements with respect to (1) action of duration, and (2) magnitude. For instance, in Fig. 3, solution (a) and (b), and (c) and (d) are spatially as well as physically similar respectively. An example of solutions that are spatially similar but not physically similar is shown in Fig. 1. Solutions shown in the third level in Fig. 1 (generic physical embodiment) are regarded as spatially similar but not physically similar. However, solutions shown in the fourth level (physical embodiment) are regarded as physically similar. Grouping solutions, which are physically similar, prevents missing any valuable concepts.



(a) LEVER2 WEDGE TIEROD1 (b) LEVER2 WEDGE

(c) CAM2 SHAFT SHAFT CAM1 (d) CAM2 SHAFT CAM1

Fig. 3. Examples of similar solutions



(a) TIEROD1 WEDGE LEVER1 (b) LEVER1 WEDGE TIEROD1

Fig. 4. An example to demonstrate that solutions can be functionally similar though not spatially similar

IV. INVESTIGATION ON HEURISTICS

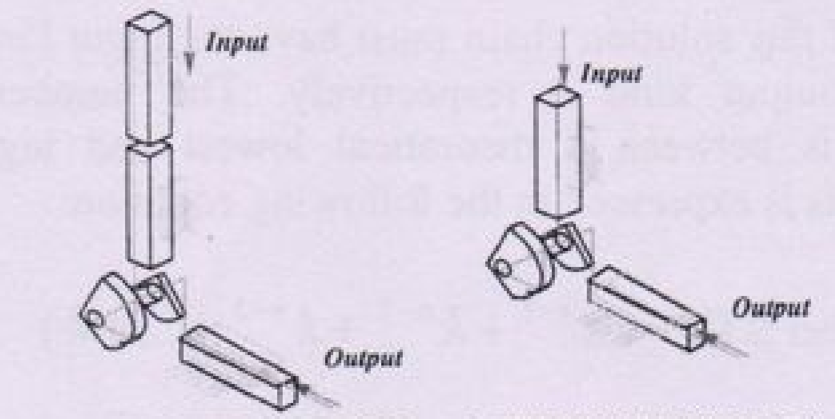
Three heuristics are discussed:

- **Heuristic 1**: Group similar (functionally and physically similar) solutions.
- **Heuristic 2**: Basic elements, whose classification is based on the same physical principle, can be grouped together at the earliest possible level (e.g., topological level).
- **Heuristic 3**: Group solutions which are functionally but not physically similar into clusters.

A. Heuristics 1: Group Similar Solutions

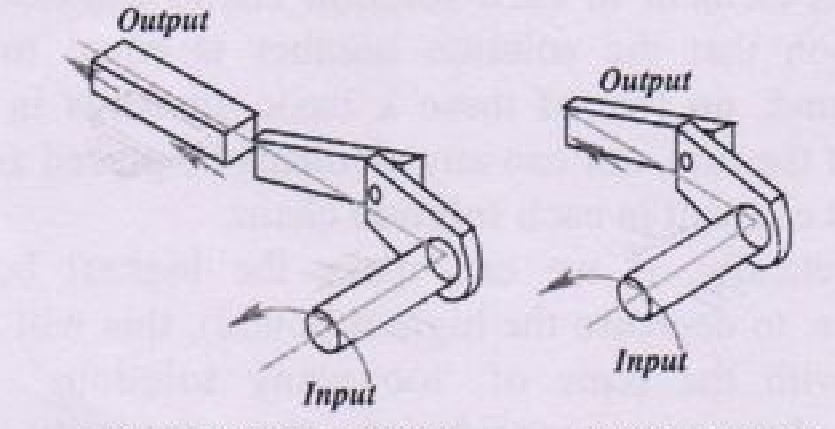
There are two cases:

- **Case 1**: Among the solution chains, all those having two of the same basic elements connected consecutively is functionally and physically similar to those having one such element, and therefore these two solutions can be grouped together. For example, in Fig. 1(a) and (b), the solution chain (Tierod1 → Tierod1 → Crank2 → Shaft → Crank1 → Tierod1) is similar to the solution chain (Tierod1 → Crank2 → Shaft → Crank1 → Tierod1). This is true for the basic elements that transmit or channel translation or rotation, i.e. Tierod1, Tierod2, and Shaft.
- **Case 2**: Solutions that can be generated by others. Three examples are shown in Fig 6. For example, the solution chain (Shaft → Cam1) is obvious to one who has seen (Shaft → Cam1 → Tierod1), as shown in (a) and (b).



(a) TIEROD1 TIEROD1 CRANK2 SHAFT CRANK1 TIEROD1 (b) TIEROD1 CRANK2 SHAFT CRANK1 TIEROD1

Fig. 5. This simplified example demonstrates those having two of the same basic elements connected consecutively have the same embodiment as those having one such element.



(a) SHAFT CAM1 TIEROD1 (b) SHAFT CAM1

Fig. 6. The simplified example of Case 2 that solutions can be grouped

Using the two cases of Heuristic 1, all the solution in Figure 4.4. which contain two basic elements and one basic element can be grouped and discarded, because each is similar to an equivalent solution with three basic elements.

This heuristic can be extended to the generic situation so that for each kind synthesis with the maximum allowable number of elements to be n, all those solutions having less than n basic elements in the solution chains can be grouped and discarded. For the theoretical investigation, applying Heuristic 1, the theoretical highest bound will be decreased to k^n, and "(Equation 1)" will become:

$$0 \leq number \leq k^n \tag{2}$$

Heuristic 1 is developed based on a judgment of whether or not solution chains are similar. Heuristic 1 cannot be applied unless at least one appropriate basic element (i.e. Tierod1, Tierod2, or Shaft) for channeling or transmitting input is selected from the database elements for kind synthesis.

B. Heuristic 2: Group Basic Elements

Those basic elements, whose classification is based on the same physical principle but represent differently in the spatial level, can be grouped together. For instance, Tierod1 and Tierod2 are used to transmit or channel the input to output. These two basic elements are based on the same physical principle, with different spatial relationships between the input-output direction in the possible embodiment. This is also true for Lever1 and Lever2. These basic elements at the topological solution level can be treated as the same without considering spatial relationships. If we use a Tierod element to represent both Tierod1 and Tierod2 basic elements, any solution chains with (Tierod1 → Ba), and (Tierod2 → Ba) would be represented in terms of (Tierod → Ba), where Ba stands for the rest of basic elements in the solution chains. The solution chain (Tierod → Tierod) represents the possible solution chains (Tierod1 → Tierod2), (Tierod1 → Tierod1), (Tierod2 → Tierod2), and (Tierod2 → Tierod1). Note that Heuristic 2 can be used in conjunction with Heuristic 1: any solution chain (Tierod → Tierod → Ba) is equivalent to (Tierod → Ba). Among k basic elements, if they can be grouped into (k-m) basic elements, the highest bound of "(3)" will be

$$0 \leq number \leq (\ k-m\ )^n \tag{4}$$

The underlying reason of applying Heuristic 2 is to make sure that all the basic elements used for synthesis are independent at the present level. Applying Heuristic 2 will be valid for situations where these basic elements are "dependent".

C. Heuristic 3: Group Solutions into Clusters

Those solution chains which are functionally but not physically similar can be clustered together for easy search. An example is those solution chains (Lever → Tierod → Wedge), (Lever → Wedge → Tierod), (Wedge → Lever → Tierod), (Wedge → Tierod → Lever), (Tierod → Wedge → Lever), and (Tierod → Lever → Wedge) which share the same physical principles but are combined in a different order. Therefore, the whole solution space will be grouped into a few sets of clusters. Heuristic 3 is particularly useful for the situations in which a huge number of solution chains still exists after applying Heuristic 1 and 2. Designers are unable to look at each of these solutions within time limitations; thus they need to navigate this large solution space with a possible guide. In addition, applying Heuristic 3, designers will be prevented from being in a situation in which they spend more time on organizing a huge amount of solutions than on carrying out the actual search for solutions. Each cluster can be investigated and compared. If designers are interested in a particular cluster, they can look at all the solutions in that cluster. The upper bound of "(2)" with Heuristic 1, 2, and 3 will become (where i = k-m, and i > n),

For n = 2, $0 \leq number \leq (\frac{(i-1)\times i}{2} + i)$

For n = 3, $0 \leq number \leq (\frac{i\times(i-1)\times(i-2)}{6} + i\times(i-1) + i)$

And, for n =4,

$$0 \leq number \leq (\frac{i\times(i-1)\times(i-2)\times(i-3)}{24} + i\times(i-1)\times(i-2) + 2\times i\times(i-1) + i) \tag{3}$$

Note that a different combination of the same basic elements can have very different physical embodiments. Grouping solutions into clusters is by no means the same as grouping similar solutions. However, this will provide a good way to navigate and search ideas that are more "different". Instead of searching a large number of solutions, a suitable solution number is first selected so as to be able to manually navigate the solution space, and search for a feasible solution. This feasible solution might provide clues to search for an optimal solution.

V . DISCUSSIONS OF THE HEURISTICS

Although the three heuristics, shown in Section III, might not be effective in all situations, in general, they will be effective, and their effect will be large. While the perspective taken is primarily with respect to basic elements, it is noted that the underlying meaning of these heuristics and their attendant combinatorial reductions might also be valid for other synthesis algorithms. Three points are discussed:

- In general, generating a wide range of solutions will help enhance the quality of the promising solution because of a more comprehensive search of the solution space. However, in practice, designers are only capable of manually exploring a small part of the solution space. Therefore, developing heuristics or other techniques could aim at reducing the (large) number of solutions which are similar or infeasible, and make the solution space more manageable. In the long-term, with the development of comprehensive heuristics or techniques, the number of the resulting solutions can be smaller, and this will help designers to effectively explore the solution space.
- In general, the variety of concepts increases with the number of building blocks used for synthesis, and the chosen maximum allowable elements for each solution chain. In other words, the number of building blocks and the maximum allowable elements determine the size of the solution space. However, the number of similar and infeasible solutions also increases when increasing the number of building blocks and the maximum allowable element number, and this will increase the complexity and time taken to organize the solution space. As a guide, the number of allowable elements chosen for synthesis should be related to the number of basic elements. If, for a given number of basic elements, too many elements are allowed in each solution, we found that the additional solutions merely contain repeated elements. A way to solve these problems is required.
- More mechanical engineering heuristics can be collected from more specialized design situations; however, these can only be obtained from understanding what the potential embodiments of a topological solution might be. Implementing solutions to more design problems by using this methodology will help gradually collect experience to support dealing with the solution space in a more efficient way.

VI. CONCLUSIONS

This paper proposed methods for screening a solution space theoretically as well as practically. The idea of using these is to discard infeasible solutions, and similar solutions, and put solutions into clusters so that designers can simply investigate one or a few representatives from each cluster to get an overview of the solution space, rather than having to explore the space exhaustively. The heuristics were developed to group solutions based on functional, spatial, and physical descriptions. These methods provide designers with a means to quickly prune the solution space and to effectively navigate the solution space, while attempting not to miss valuable solutions. Future research includes exploring additional heuristics and investigating the effectiveness and constraints further .

REFERENCES

[1] R. Bracewell, and J. Sharpe, Functional Descriptions Used in Computer Support for Qualitative Scheme Generation – Schemebuilder, AI EDAM, Vol. 10, pp. 333-345, 1996.

[2] T. Tomiyama, V. D'Amelio, J. Urbanic, and W. ElMaraghy, Complexity of Multi-Disciplinary Design, CIRP Annals Manufacturing Technology, Vol. 56, No. 1, 2007.

[3] A. Chakrabarti, P. Langdon, Y.C. Liu, and T.P. Bligh, Supporting Compositional Synthesis on Computers, Engineering Design Synthesis: Understanding, Approaches and Tools, pp. 179-197, Springer, London, 2002.

[4] Y.L. Tian, H.J. Zou, and W.Z. Guo, An Integrated Knowledge Representation Model for the Computer-Aided Conceptual Design of Mechanisms, Int J Adv Manuf Technol, Vol. 28, pp. 435-444, 2006.

[5] J. Jupp, and J. S. Gero, Towards computational analysis of style in architectural design, *Journal of the American Society for Information Science* **57**(11): 1537-1550, 2006.

[6] T. J. McCarthy, Knowledge Based Systems in Structural Steelwork Design: A Review, Engineering Computational Technology, pp. 53-74, Civil-Comp, Edingburgh, UK, 2002.

[7] A. Chakrabarti, M. X. Tang, M.X. Generating conceptual solutions on FuncSION: evolution of a functional synthesiser. Proceedings, 4th International Conference on Artificial Intelligence in Design, AID'96, Stanford, CA, Edited by J.S. Gero, F. Sudweeks, pp. 603-622, Kluwer Academic, 1996.

[8] K. Roth, Design models and design catalogs, Proceedings 6th International Conference on Engineering Design ICED'87, Boston, MA, pp.60-67, 1987.

[9] T. Murakami, and N. Nakajima, Mechanism concept retrieval using configuration space, Research in Engineering Design, 9, pp. 99-111, 1997.

[10] L. Qian, and J. S., Gero, J. S, Function-behaviour-structure paths and their role in analogy-based design, *AIEDAM* **10**:289-312, 1996.

[11] G. Pahl, and W. Beitz, *Engineering Design - A Systematic Approach*, Springer-Verlag, London, 2007.

[12] Chakrabarti A. and Bligh T. P. Functional Synthesis of Solution-Concepts in Mechanical Conceptual Design. Part I: Knowledge Representation, Research in Engineering Design, **6**(3), pp.127-141, 1994.

[13] Chakrabarti A. and Bligh T. P. Functional Synthesis of Solution-Concepts in Mechanical Conceptual Design. Part II: Kind Synthesis, Research in Engg. Design, **8**(1), pp-52-62, 1996.

[14] Chakrabarti A. and Bligh T. P. Functional Synthesis of Solution-Concepts in Mechanical Conceptual Design. Part III: Spatial Configuration, Research in Engineering Design, 8(2), pp.116-124, 1996.

[15] Chakrabarti A., and Bligh, T.P. An Approach to Functional Synthesis of Design Concepts: Theory, Application, and Emerging Research Issues, AI in Engineering Design, Analysis and Manufacturing, **10**(4), pp.313-331, 1996.

[16] Y. C. Liu, A. Chakrabarti, and T. P. Bligh, Towards an Ideal Approach for Concept Generation, Design Studies Journal, **24**(4), pp 341-355, 2003.