Transforming Engineering Problems through Graph Representations.

Dr. Offer Shai
Department of Mechanics Materials and Systems
Tel Aviv University, Ramat Aviv 69978, Israel
Tel: 972-3-6406710, Fax: 972-3-6407617
email: shai@eng.tau.ac.il

Abstract

The paper introduces a general approach for solving engineering problems by transforming them to problems in other engineering fields, through general discrete mathematical models called graph representations. The idea of the method is first to raise the problem to an abstract mathematical level of graph representations. At that abstract level, either the solution is found through the tools of graph theory, such as the graph duality principle, or the problem is transformed further to another engineering domain, where the problem is associated with a known solution.

The paper demonstrates a number of applications of the approach, among them: deriving new concepts in engineering; establishing new engineering designs; analyzing complicated engineering systems, and a generic treatment of both analysis and design.

The paper draws the correlation of the suggested approach to known AI topics: representation change and classification problem solving method.

Keywords: knowledge transformation, graph representations, graph theory, duality, engineering design, engineering system analysis, classification problem solving.

1 Introduction

The current paper introduces a general approach for solving engineering problems by transforming engineering knowledge. The approach employs the so-called graph representations (Shai, 2001b) to reason with various engineering problems and systems. So far, the approach has yielded a number of practical and theoretical applications that were described in previous publications, as follows: analysis of integrated engineering systems (Shai and Rubin, 2003), systematic design of engineering systems (Shai, 2003), finding relations between different engineering fields (Shai, 2001a; Shai, 2002a), establishing new ways of collaboration between engineers from different fields (Shai and Reich, 2003), deriving known and new theorems and methods in different engineering domains (Shai, 2001b), establishing relations between different known methods in engineering (Shai 2001c) and checking validity of engineering systems (Shai and Preiss, 1999).

The method applies in a similar way to both analysis and design problems. The former deals with predicting the behavior of an engineering system, while the latter deals with synthesis of new engineering systems to produce some required behavior. Till now, graph representations were employed in mechanical linkages, trusses,

skeletal structures, dynamical systems, gear trains, electronic circuits, hydraulic systems and other engineering domains.
Throughout the paper the reader is provided with comprehensive examples emphasizing the abilities of the approach. The main examples cover the following types problems: checking the validity of an engineering system, analyzing integrated engineering system, designing a new mechanical system.
The structure of the paper is as follows:
Section 2 generally describes the strategy used by the approach to solve engineering problems and how it can be interpreted in AI. It is shown that changing the representation as a strategy for solving problems coincides with the ideas formulated by Simon (Simon, 1969), Amarel (Amarel, 1968) and Korf (Korf, 1980). Furthermore, a correlation of the presented approach to the Clancy's heuristic classification model was drawn. It is explained that due to its mathematical basis, the approach enables to implement systematically these two major AI topics in engineering. In the sake of clarity, the mathematical details are left for the sections that follow.
Section 3 defines and describes in detail the entity called a graph representation. Graph representations are shown to be mathematical abstractions capable of reflecting the structure and properties of engineering systems.
Section 4 shows the process of knowledge transformation between engineering fields through the graph representations. Practical examples demonstrate the abilities of the approach to yield innovative engineering solutions.
Section 5 systematizes the abilities described so far into a general methodology of engineering problem solving. The methodology is divided into four main steps performing the interplay between engineering domains and graph representations.

1. Transforming engineering systems as a strategy for solving problems.

Many works dealing with representing problems have been reported in the AI literature (Newell, 1965). One of the central issues discussed in these works addresses changing the representation of a problem. Amarel (Amarel, 1968), for example, demonstrated the importance of this issue through a series of representation transformations upon the known problem of missionaries and cannibals, so to make the solution of this problem trivial. He showed that there exists a relationship between different ways of formulating a problem and the efficiency of its solution. Simon (Simon, 1969), has also pointed to an essential role that can be played by the transformation process in problem solving. In his book, "The Sciences of the Artificial" (Simon 1969), in the section dealing with design, he provides an example demonstrating this idea on a game. This example is brought here to provide a reference to the approach devised in the paper.

Example 1. The game of number scrabble.

Number scrabble is played with nine cards, valued from one through nine (**Figure 1**). The cards are placed in a row, face up, between the two players. The players select and draw, alternately, one of the cards that remain in the center. The aim of the game is for a player to make up a "book", that is, a set of exactly three cards whose spots add to 15, before his opponent can do so. The first player who makes a book wins; if all nine cards have been drawn without either player making a book, the game is a draw.

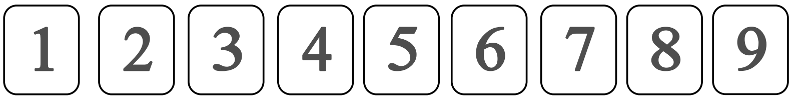


Figure 1. Number scrabble original card arrangement: nine cards numbered one through nine placed in the row, face up.

Developing a winning strategy for this game seems superficially to be a difficult problem. Simon suggested to transform the game by placing the cards in a different fashion depicted in **Figure 2**. In the new order, the cards form a "magic square", its members summing to 15 in each column, row and diagonal. The "book" now becomes a set of a three cards which form column, row or diagonal in the magic square, thus turning the number scrabble into the tic-tac-toe, the winning strategy for which is well known and established (Nilsson, 1971).

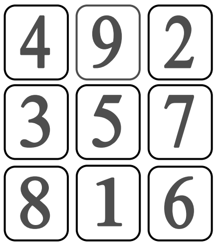


Figure 2. Number scrabble transformed arrangement: the cards of Figure 1 in rearranged position, allowing to transform the number scrabble game into tic-tac-toe.

Example 1 demonstrates the idea behind a general strategy of problem solving. In Simon's own words: "This view can be extended to all of problem solving - solving a problem simply means representing it so as to make the solution transparent". Figure 3 schematically depicts such a process of problem solving.

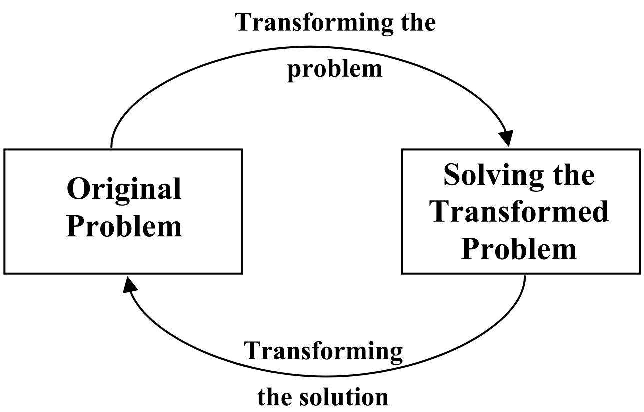


Figure 3. Diagram outlining Simons' problem solving strategy: the original problem (number scrabble) is transformed into some other problem (tic-tac-toe) for which the solution is known or transparent. Then, the solution of the transformed problem is transformed into the terminology of the original problem thus yielding the solution of the original problem

The current paper adopts the approach of representation change and applies it for solving real engineering problems, while introducing an approach that systemizes the transformation process. In Simon's example, a certain amount of creativity was required to arrive at the idea of turning the number scrabble into a magic square and tic-tac-toe. In order to enable systematic performance of this process with application to various practical engineering problems the suggested approach uses general discrete mathematical representations, called graph representations (Shai, 2001b). These representations provide a mathematical basis for systematic finding and carrying out of a proper transformation of the engineering problem. Specifically, a graph representation is used to create a well-established isomorphic mapping of the behavior of an engineering system. Accordingly, the problems related with specific engineering domain can be mapped into problems in the graph representation. Then, the problem can be transformed to other engineering domains related to the graph representations, thus forming a new – transformed engineering problem (Figure 4).

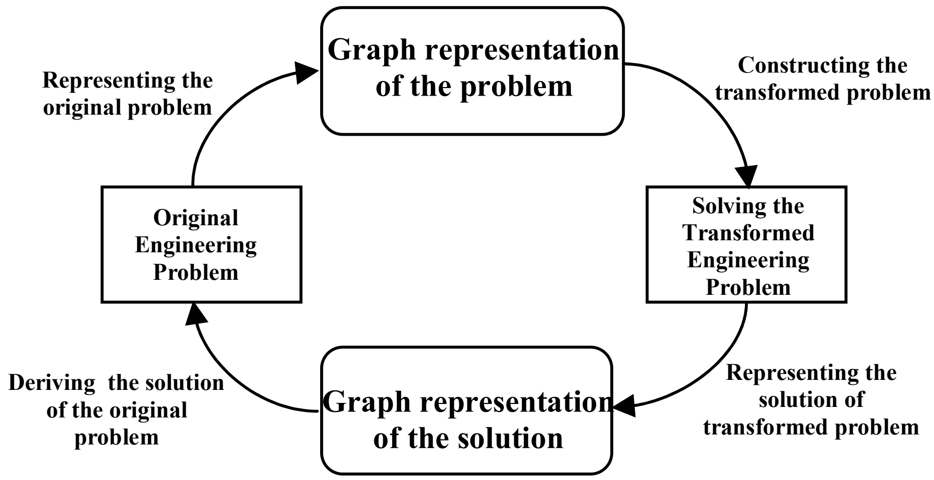


Figure 4. Suggested problem solving strategy: the transformations depicted in Figure 3 are augmented by intermittent level of graph representations, enabling to systematize the transformation process.

If the solution for the transformed problem is known, then the same link is used to transfer it back to the original engineering domain, thus yielding a solution to the original problem.

Bringing the mathematical meta-level of graph representations into the process of problem solving through transformations also enables to relate the suggested method to heuristic classification methodology for solution of problems (Clancy, 1985). In general, the method of Clancy can be seen as comprised of three main steps: abstraction – the step at which the problem data is transformed into abstract terminology, heuristic match – the step at which the problem is "matched" with specific abstract solution, and refinement – the step where the obtained solution is fitted to the requirements of the original problem. Same steps can be traced in the method depicted in Figure 4. As the level of graph representations is an abstract mathematical level, the problem formulation in the terminology of graph representations correlates to the data abstraction of the problem. When the problem is transformed to another engineering domain, and the solution from that domain is transformed back to the graph representation, the result presents an abstract solution to the abstract formulation of the problem. It should be noted that this "matching" of the solution is performed by means of strongly established mathematical tools, such as graph duality principle (section 3.2) and the construction rules of the graph representations. Finally, the transformation of the solution from graph representations to the original engineering domain, can be seen as the refinement of the solution, since the final product of this step is a solution that based on the of mathematical foundation of this method satisfies the requirements of the original problem.

Example 2 shows an application of the suggested approach for solving an engineering problem. For convenience, analogous terms from Example 1 are provided in the brackets besides the corresponding terms along the explanation of the example.

Example 2. Determining the Stability of a Truss through graph duality transformations.

Current example demonstrates usage of the so-called graph duality transformation, which is one of the powerful transformations employed by the current approach. This transformation will be thoroughly described in section 3.2. Duality transformation is based on the well-known duality principle from graph theory (Swamy and Thurasiraman, 1981), which defines a mathematical correspondence between the properties of two distinct graphs. On the basis of this duality, a duality relation between the domain of trusses and the domain of mechanisms was established (Shai, 2001a). The essence of this relation is that for each truss there exists a corresponding dual mechanism, whose mechanical properties match to those of the truss. For instance, the author has proved that a truss is stable if and only if its corresponding dual mechanism is mobile.

Figure 5a, shows a complicated truss system [*row of cards*], for which there is a requirement to determine whether it is stable or not [*make up a "book"*]. Finding a solution to this problem without performing calculations is not easy even for experts in mechanical engineering. Thus, the current approach suggests transforming this problem to a problem in the dual mechanism [*magic square*] (Shai, 2002a).

The problem of whether the truss is stable [*make up a "book"*] is transformed into the problem whether the dual mechanism is mobile [*win in tic-tac-toe*], a known problem in machine theory [*artificial intelligence and computer science*] for which there exist a number of methods [*winning strategies for tic-tac-toe*].

Figure 5a shows the truss and Figure 5b shows the transformed engineering system, which for this case is the mechanism dual to the truss. In the mechanism of Figure 5b, links 1 and 9 are co-linear thus, on the basis of the basic mechanism theory, it can be straightforwardly concluded and mathematically proved that the mechanism in Figure 5b is stuck (Norton, 1992). The latter conclusion being the solution of the transformed problem can be mapped into the solution of the original problem. Since the mechanism dual to the original truss is locked it follows that the truss is not stable. **Figure 6** shows, in a schematic form, the outline of the process performed above.

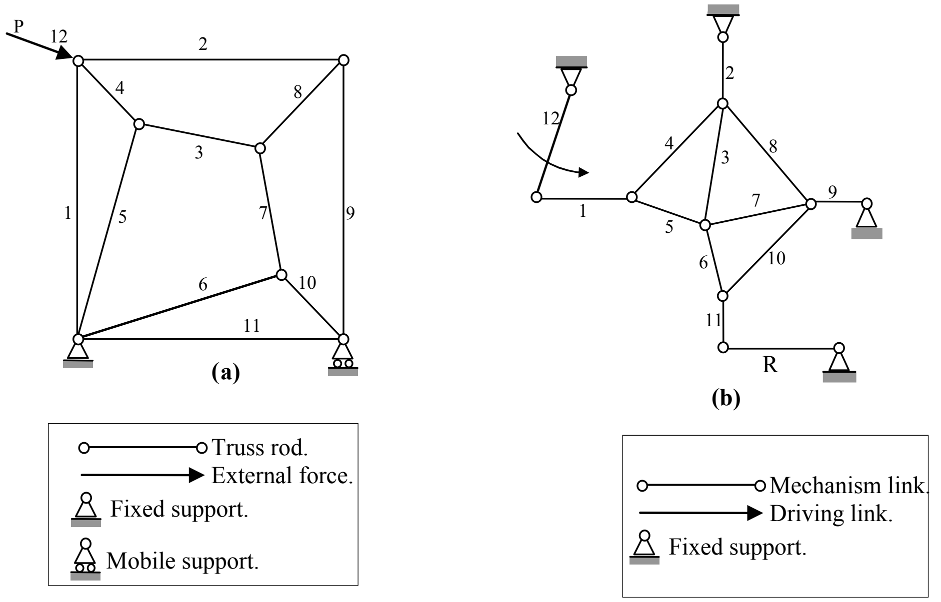


Figure 5 : Transforming the truss to its corresponding dual mechanism: (a) The schematic presentation of a plane truss, acted upon by an external force - 12, (b) The mechanism dual to the truss – each link in the mechanism corresponds to an element in the truss (rod, external force, reaction), as is highlighted by means of the matching numbering.

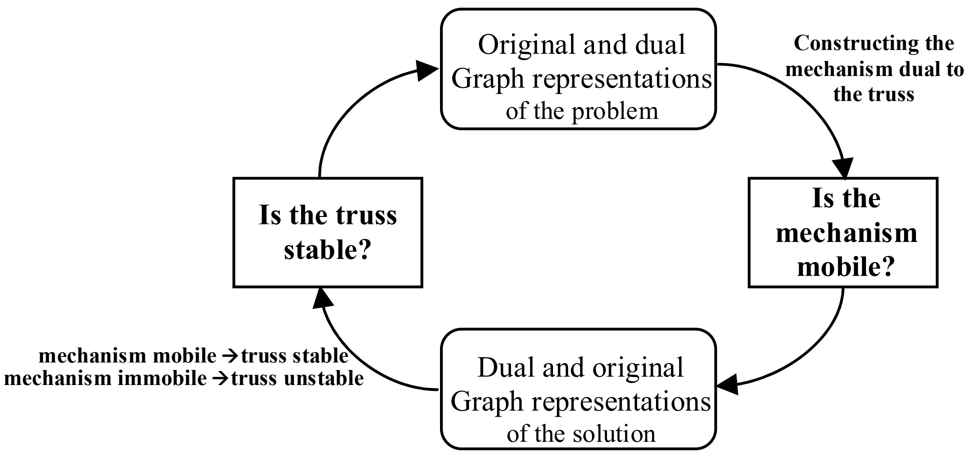


Figure 6. Applying the method for checking the stability of trusses: the problem of whether the truss is stable is transferred first to the level of graph representations, where it turns into the problem of graph representation validity. Then through the duality transformation the problem is transformed into the problem of whether certain mechanism is mobile. The answer is transferred back to the graph representations and then to the domain of trusses, thus yielding the solution to the original problem.

2 Structure of the graph representation

A graph representation is an isomorphic graph-theoretical abstraction of an engineering system, which can be used for design (Shai, 2003; Shai and Reich 2003), analysis (Shai, 2001b; Shai, 2001c; Shai and Rubin, 2003), and other forms of reasoning upon the system. Different types of graph representations are characterized by four main parts, as follows:

1) **Embedded Knowledge** – This primary component contains mathematical knowledge underlying the representation, including mathematical laws, theorems and methods. Some of the embedded knowledge varies from one type of graph representation to another, while some, such as the theorems and laws of graph theory, constitute a constant core of the representations.
2) **Relations to other Graph Representations** – Mathematical relations in graph theory yield relations between different types of graph representations.
3) **Represented Engineering Domains** - Each Graph Representation can be applied to represent a number of engineering domains. Choosing a proper Graph Representation for an engineering system being treated has a crucial impact on the effectiveness of the subsequent process.
4) **Rules for construction of the Graph Representation** – The algorithmic steps for constructing the graph representation corresponding to an engineering system. This part can be seen as a "dictionary" for changing the terminology of the engineering system to the terminology of the graph representation and vice versa.

Figure 7 schematically outlines the structure of the graph representation and its four components.

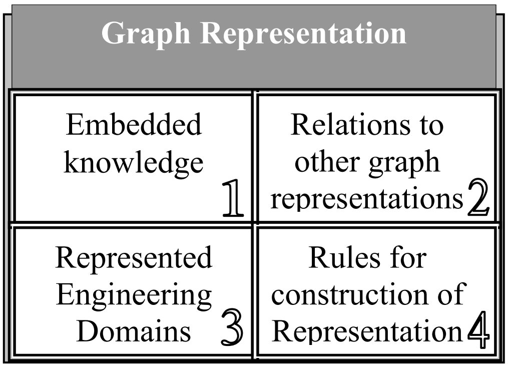


Figure 7. Structure of graph representation: the four fundamental components of the graph representation.

A graph representation can be seen as a discrete mathematical model of the system it is applied to represent. Being such, the four components underlying the representation can be formalized in terms of discrete mathematics, such as matrices, trees and predicate calculus. Consequently, the layout of graph representation makes it convenient for computerization and the engineering problems can be transformed to problems in graph theory for which there exist efficient algorithms (Even 1979).

Example 3 demonstrates a graph representation of a determinate truss, while outlining the four components described above.

Example 3. Flow Graph Representation of determinate trusses.

Figure 8 shows an example of a truss and a Flow Graph Representation representing it.

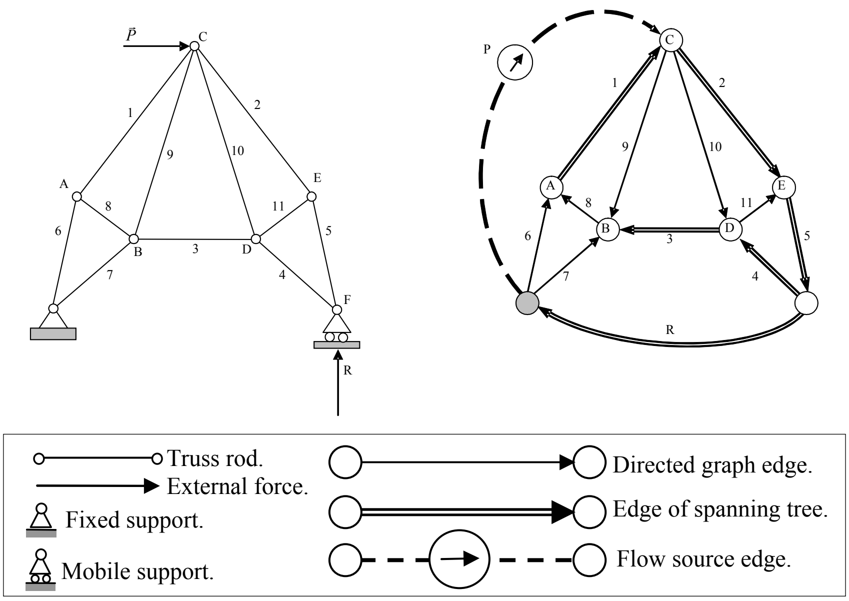


Figure 8. (a) Schematic presentation of a determinate truss and (b) Corresponding graph representation – a graph where each edge corresponds to an element of the truss and each vertex to its pinned joint.

The four components of the Flow Graph Representation of a determinate truss are as follows:

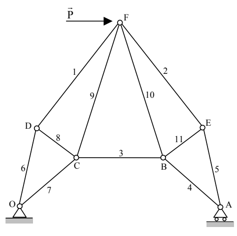
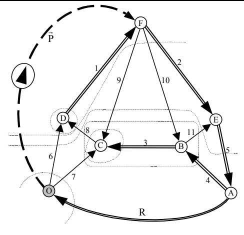
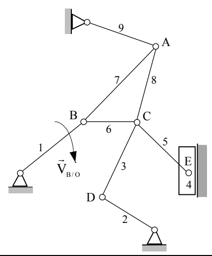
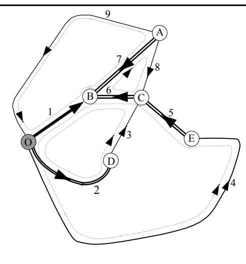
1. **Embedded knowledge:**
   a. Theorems and methods from graph theory: cutsets, spanning trees, degrees of the vertices, etc (Swamy and Thulasiraman, 1981). In the graph of Figure 8b, the edges belonging to the spanning tree are designated by double lines. Each edge in the spanning tree defines a cutset – a set of edges whose removal from the graph will turn it into a disconnected graph. Each edge not in the spanning tree defines a circuit – a set of edges forming a closed path.
   b. Flows - each edge in the graph is associated a vector variable called 'flow'. The flows in the graph satisfy the 'flow law' saying that the sum of the flows in each cutset of the graph is equal to zero (Shai, 2001a).
   c. Methods for checking graph rigidity (Shai, 2001b; Recski, 1989; Lovasz and Yemini, 1982).
2. **Relations to other representations:**
   a. This representation possesses a duality relation to the Potential Graph Representation (Shai, 2001a) as is explained in detail in section 3.2 .
3. **Represented engineering domains:**
   a. Determinate structures (Shai 2001a) – such as the current example – a determinate truss.
   b. Static systems (Shai, 2001b).
   c. Electric circuits (Shai, 2001b).
4. **Rules for constructing the representation (Shai 2001a):**
   a. Create a vertex in the graph for each pinned joint in the truss. In Figure 8, the corresponding vertices and the pinned joints are designated with matching letters – from ‘A’ to ‘F’.

b. Create an edge in the graph for every rod; its end vertices correspond to those joints that connect the rod to the truss. In Figure 8 the corresponding edges and rods are designated with matching numbers from ‘1’ to ‘11’. The direction of each truss edge is arbitrarily assigned. The force in the rod is represented by the flow through the corresponding edge. Specifically, meaning of the flow in edge ‘e’ is the force applied to the head vertex (joint) by the corresponding rod, which is, of course, equal to the force that the tail vertex (joint) applies to the rod. Accordingly, the flow law satisfied by the flows in the graph corresponds to the force equilibrium satisfied by the forces acting in the truss.

c. An extra vertex called the “reference vertex”, is now added (the vertex filled with grey color in Figure 8). This might be thought of as the ‘earth’.

d. The following edges for the external forces and reactions are added:
   i. For each externally applied force a “flow source edge” is added, from the reference vertex to the vertex corresponding to the joint upon which it acts. In Figure 8, both the flow source and the external force are designated with the label ‘P’.
   ii. For each reaction a “reaction edge” is added from the vertex corresponding to the joint at which the reaction acts, to the reference vertex - one for each relevant principal direction. Thus, in a plane truss, for each mobile support there will be one corresponding edge, and for a pinned support there will be two corresponding edges. In Figure 8, the reaction edge and the reaction are designated with ‘R’.

Table 1 lists the main graph representations employed by the current approach, together with their basic properties and references to a more detailed description of all the four components of the representation.

Type of graph representation	General Description	Related engineering disciplines	References	Example of engineering system (schematic description)	The graph representation of the example engineering system
Flow Graph Representation (FGR).	Each edge in FGR is associated a vector called 'flow'. Flows in FGR satisfy the "flow law", stating that the sum of flows in each cut-set is equal to zero.	Determinate structures, static systems, electric circuits.	Shai 2001a, Shai 2001b, Shai 2002b.	Determinate truss: 	
Potential Graph Representation (PGR).	Each vertex in PGR is associated a vector, called 'potential' that satisfy the 'potential law', saying that the sum of potential differences in each circuit is equal to zero.	Mechanisms, electric circuits, conjugate structures.	Shai, 2001a, Shai 2001b.	Mechanism: 	
Resistance Graph Representati	Generalizes properties of FGR and PGR. Additionally, each edge	electronic circuits, indeterminate	Shai, 2001b.	Integrated system:	

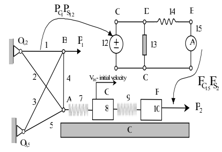
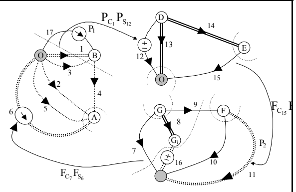
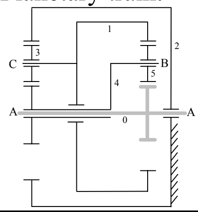
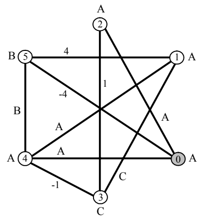
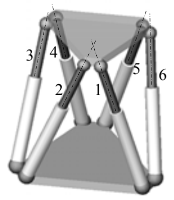
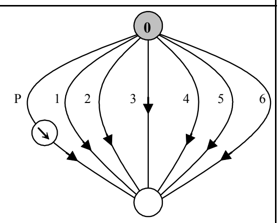
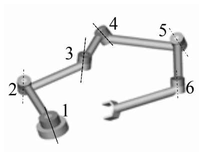
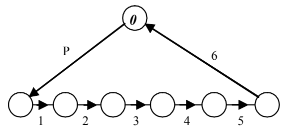
Type of graph representation	General Description	Related engineering disciplines	References	Example of engineering system (schematic description)	The graph representation of the example engineering system
on (RGR).	is associated with a terminal equation defining the relation between its flow and potential difference.	structures, hydraulic and dynamic systems, integrated systems.			
Line Graph Representation (LGR).	It corresponds vertices to the elements of the engineering systems and edges to the relations between them.	Gear and planetary trains.	Shai 2001b, Shai and Preiss 1999.	Planetary train: 	
Flow Line Graph Representation (FLGR)	Possesses the same embedded knowledge as FGR, augmented by the construction rules from LGR.	static pillar systems, Stewart platforms	Shai 2002b.	Stewart platform: 	
Potential Line Graph Representation (PLGR)	Possesses the same embedded knowledge as PGR, as augmented by the construction rules of LGR	Mechanisms, serial robots	Shai 2002b.	Serial robot: 	

Table 1. Typical graph representations and their properties.

The graph representations constitute a meta-level to which the knowledge underlying engineering systems can be brought or "ascended", as is visualized in Figure 9. The figure summarizes two types of relations associated with graph representations: the relations between the graph representations and themselves; the correspondence between the graph representations and engineering domains. The map enables one to see the possibility of laying out different knowledge transfer routes between the representations and the engineering domains.

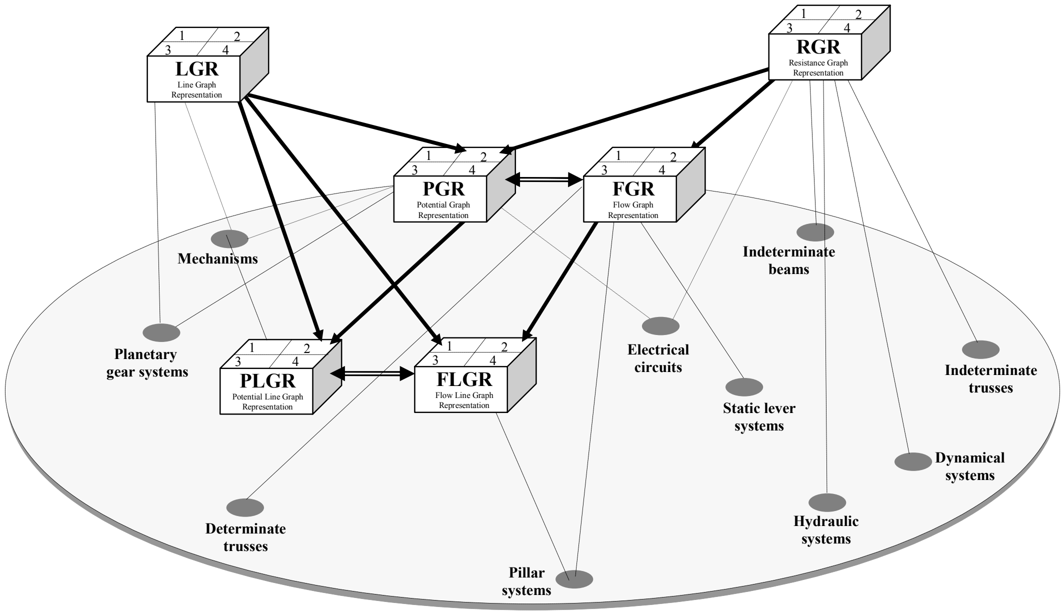


Figure 9. The hierarchic map of the graph representations and represented engineering domains.

3 Transforming systems between different engineering domains.

As was explained in section 2, an important feature of the suggested approach is the ability of transforming knowledge between the engineering fields that hereto were considered unrelated (Shai, 2001a; Shai, 2002a; Shai, 2001b). Once a relation between two engineering fields is established, it yields a knowledge transfer channel that makes possible to transfer theorems, methods (Shai, 2001c), patents (Shai, 2003) and other important types of knowledge from one engineering field to another. These channels enable the engineers of both fields to join their resources and work in full cooperation to solve the engineering problems (Shai and Reich, 2003).

Following two sections demonstrate two different ways of performing engineering transformations that are used to yield and to utilize the relations between engineering domains.

3.1 Transformations based on common graph representation.

Graph representations constitute a higher mathematical level where an engineering system is seen in a more abstract way, thus enabling to reason over it more systematically. The relation between the engineering domain and the graph representation makes possible to transfer engineering knowledge back and forth between these two levels of abstraction. Current section deals with advantages that are opened when two engineering domains are represented by the same graph representation, thus making the level of graph representations common to these domains.

A piece of engineering knowledge such as theorems, methods, concepts or systems, can be brought up to the common graph representation level and then down to other engineering domain. Doing so, produces authentic knowledge in that second domain, so that the whole process can substitute performing both analysis, design and even research tasks. Example 4 demonstrates this possibility by showing how a new design in mechanics is obtained by transferring a known design from electronics to the terminology of a common graph representation and then to mechanics.

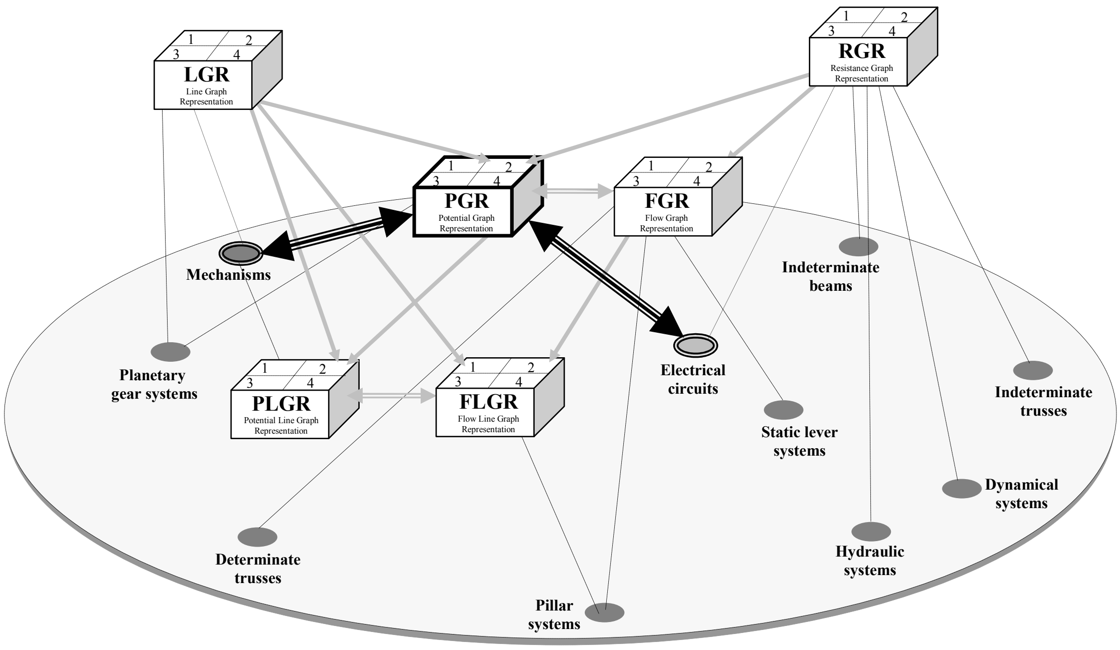


Figure 10. Relation between mechanisms and electronic circuits – the link between the two engineering domains is depicted by two bold arrows.

Example 4. Transforming electronic clipping circuit to the domain of mechanisms.

Mechanisms and electronic circuits are both representable by the Potential Graph Representation, as is highlighted in Table 1 and Figure 10. Current example demonstrates utilization of this relation.

The clipping circuit, shown in **Figure 11**, is a known engineering system (Smith 1987), that can be considered as a knowledge stored in the domain of electronics. The clipping circuit obtains a variable voltage function from the input voltage source - CO. If the value of the input voltage is below some constant value, the output voltage measured by the voltmeter BO is equal to the voltage in the input. Alternatively, if the voltage exceeds this value, the voltage at the output is freezed at that value.

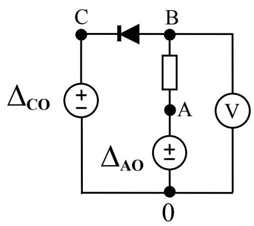


Figure 11. The clipping circuit – the input element is V_{CO} and the clipping value is defined by the voltage source V_{AO}.

In this example it will be shown how the knowledge captured in this electronic device can be transformed to another engineering field – kinematical systems, all through the common graph representation. First, the circuit has to be expressed in the terminology of the common graph representation – Potential Graph Representation (PGR). This is done by associating each element of the circuit with an edge in the graph: the voltage sources are associated with potential difference source edges, resistor with resistance edge and the diode with a unidirectional edge. The vertices of the graph are associated with the joints in the circuit. This way the equations of the potential law embedded in the representation reflect the Kirchoff Voltage Law for the circuit. The potential graph obtained through these construction rules appears in Figure 12 (detailed description of this representation construction process can found in (Shai 2001b, Shai 2003)).

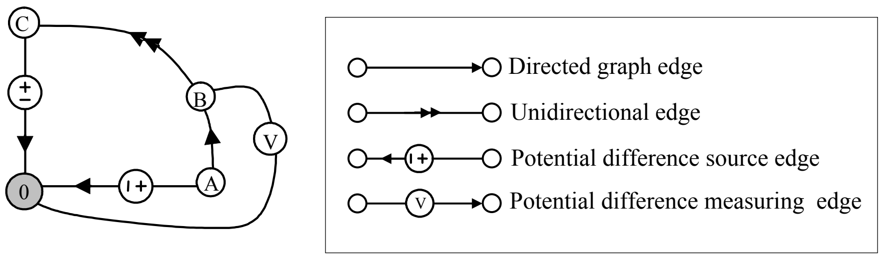


Figure 12. Potential Graph Representation of the clipping circuit.

After constructing the graph of Figure 12, it can be realized in different engineering domains. Namely, the knowledge transfer channel can be routed from the circuits through the potential graph representation to any of engineering domains related to this representation. In this example, mechanisms were chosen to be the secondary engineering domain. Figure 13 shows the gradual process of constructing the mechanism from the graph of Figure 12.

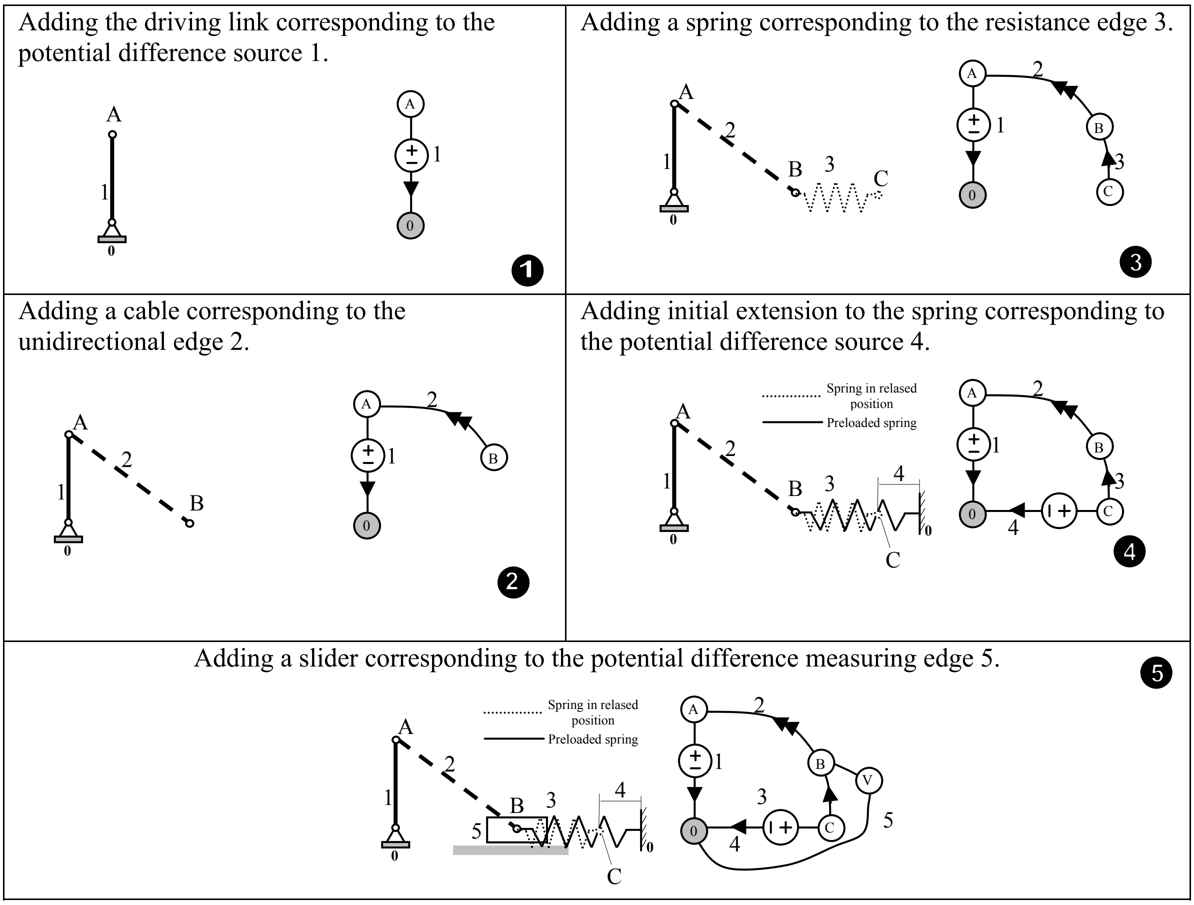


Figure 13. Gradual process of building 'intermittently stopping mechanism' from its graph representation

The mechanism obtained in the last stage of Figure 13 is a so-called 'intermittently stopping mechanism', where slider 5 performs intermittent stops during the cycle of the continuous motion of the driving link 1.

3.2 Transformations based on duality relation between graph representations.

One of the features of graph representations are the mathematical relations between them. These relations can also be used as knowledge transfer channels, while this time the knowledge remains on the graph representation level. Such channels can be integrated in the process of transferring knowledge between engineering domains that are not linked by a common graph representation. In this case the knowledge from one engineering domain is brought up to the graph representation level, then through the mathematical relations to other graph representation and then brought to other engineering domain.

This section employs the duality relation between the representations that is based on the mathematical duality between linear graphs (Swamy and Thulasiraman, 1981). Here it is employed to transform knowledge for deriving a new type of force variable in statics through the relation between the Flow and Potential Graph Representations.

It was shown in (Shai, 2001a) that the Flow Graph Representation (FGR) and the Potential Graph Representation (PGR) are dual. This relation was formalized as follows:
For each FGR (designated by G^F) there exists a PGR (designated by G^Δ) satisfying:

- *There is an edge in G^Δ for each edge in G^F.*
- *The set of edges forming a cutset in G^F correspond to a set of edges forming a circuit in G^Δ (basic property of dual linear graphs).*
- *The analysis equations corresponding to both representations are the same, thus - the flow in each edge of G^F is equal to the potential difference of the corresponding edge in G^Δ.*

Since trusses can be represented by FGR and mechanisms by PGR, the duality relation between the representations yields a duality relation between the represented engineering systems. Truss - mechanism duality is formulated as follows (Shai 2001a, Shai 2002a):

For each truss there exists a mechanism satisfying:

- *There is a link in the mechanism for each rod in the truss.*
- *The analysis equations corresponding to both engineering systems are the same, thus:*
- *The force in each rod of the truss has the same value as the relative linear velocity in the corresponding mechanism link.*

Several examples for dual systems, including different types of mechanisms and trusses are shown in **Table 2**.

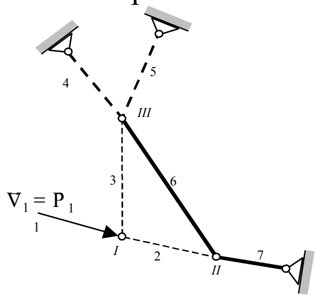
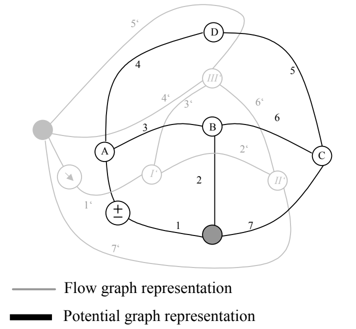
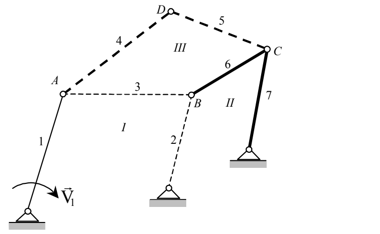
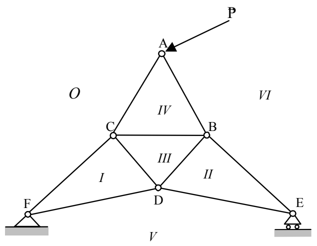
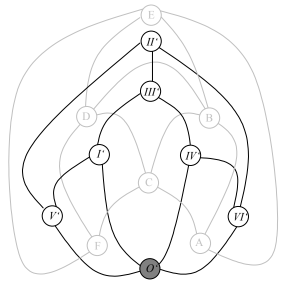
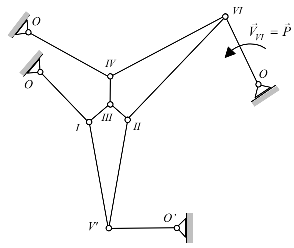
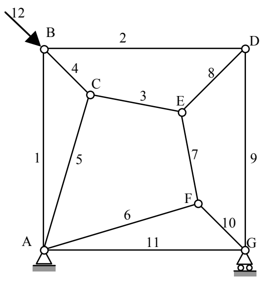
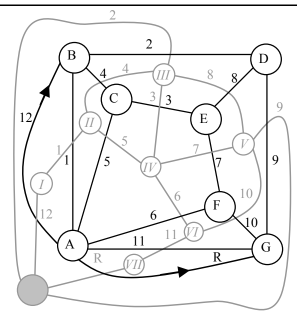
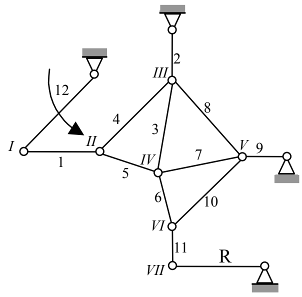
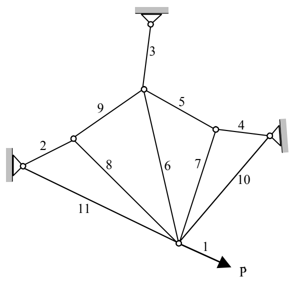
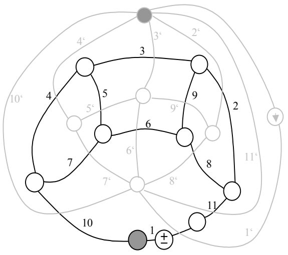
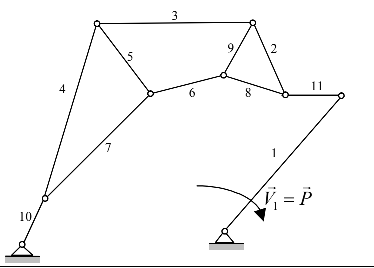
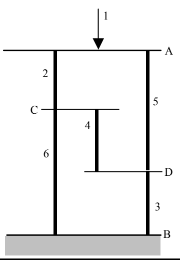
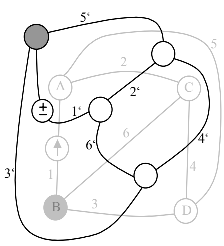
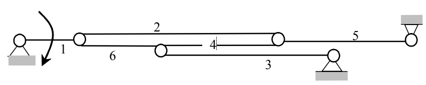
Trusses	**Corresponding graph representations**	**Mechanisms**
Simple trusses 		Mechanisms decomposable to dyads. 
Compound trusses 		Mechanisms containing higher order Assur groups 
Unstable trusses 		Immobile linkages 
Indeterminate trusses 		Linkages with more than one degree of freedom. 
One-dimensional trusses 		Infi mechanisms 

Table 2. Duality relations between different types of mechanisms and trusses.

This relation constitutes a powerful tool for transforming knowledge between the two engineering fields, thus providing an engineer with an alternative way for dealing with design and analysis tasks. Example 5 shows one of such tasks resolved by means of this relation. This example demonstrates the ability of the approach to contribute beyond solving design and analysis problems, by using it to derive new concepts by means of the duality transformation. This is done by establishing a new type of force variable in structures called the "face force" on the basis of the duality relation between trusses and mechanisms.

Example 5. Deriving a new type of variables through the duality transformation

The complete correspondence between trusses and mechanisms implies that for each variable in one system there exists a variable in the other possessing the same value. Nevertheless, a closer look into this relation reveals that there is no known type of variable in the truss corresponding to the linear velocity of a joint in the mechanism, as is depicted in **Figure 14**.

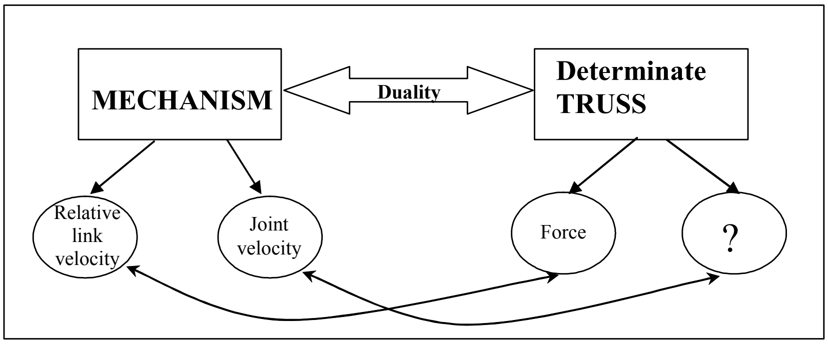


Figure 14. Duality between trusses and mechanisms: the duality relation between these two engineering systems implies the complete correspondence between the variables describing them. While the relative link velocity in the mechanism corresponds to the force in the truss, there is no existing variable corresponding to the joint velocity.

This gap reveals a new type of variable that can be used for analysis of trusses and gives a new insight into the analysis process. Furthermore, the derivation of such a new variable is easy and straightforward, since it can be done simply by employing the dualism relations and translating the properties of the linear velocity of a joint to the terminology of trusses, as is done in Table 3.

Properties of the joint velocity in mechanism	**Translation to the terminology of the truss**
The variable is a two-dimensional vector corresponding to each joint in the mechanism.	The new variable is a two-dimensional vector corresponding to each face (area bounded by the truss rods) of the truss.
Relative velocity of a mechanism link is equal to the linear velocity difference of the end joints of this link.	The force in the corresponding truss rod is equal to the difference between the new variables corresponding to the faces separated by this rod.
The linear velocity of a fixed joint is equal to zero.	The new variable corresponding to the external face of the truss is equal to zero.

Table 3. Establishing the properties of the new variable in trusses.

It can be concluded from Table 3 that the new variable established in trusses can be thought of as a multidimensional generalization of the “mesh current” in electrical circuits (Balabanian and Bickart, 1969). Here it will be called the "face force" in the truss. The action of the face forces in the truss is outlined in the example of Figure 15. The truss of Figure 15c contains three faces, thus there are three face forces. The face forces can be seen as forces that are acting in all the elements of the corresponding face. The force in rod 2’, for example, is comprised of two face forces – one acting in face formed by 1’ and 2’, and the other acting in the face formed by 2’ and 3’.

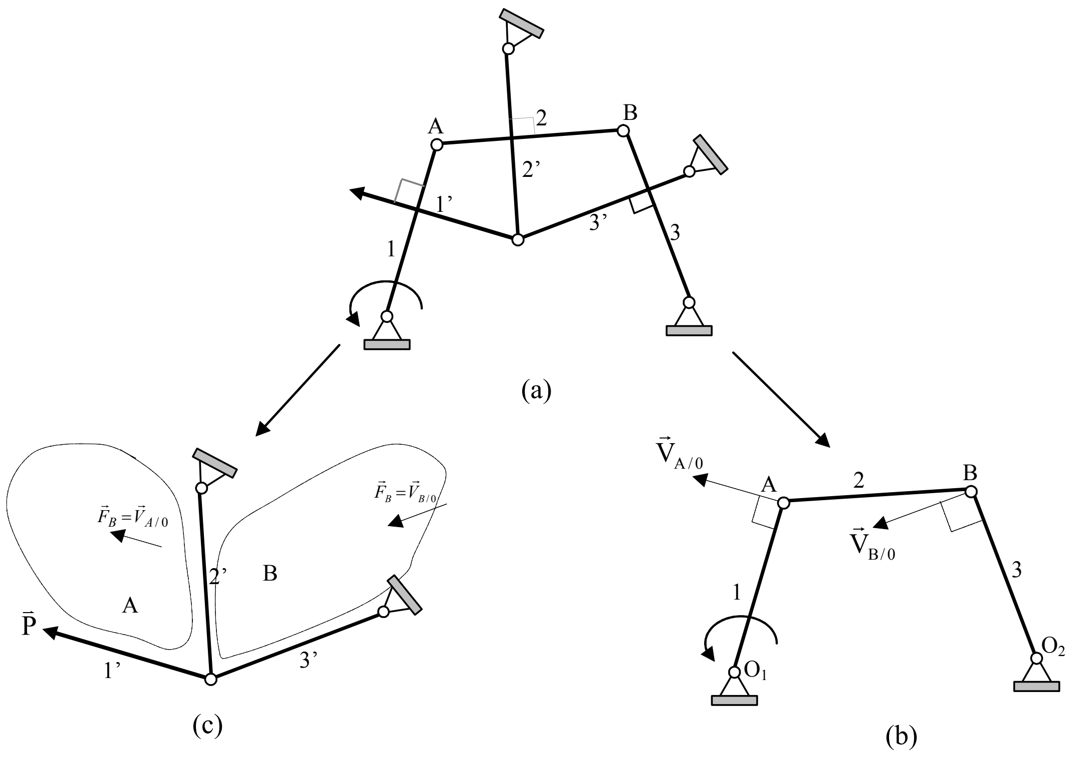


Figure 15. The face forces in the dual truss.
a) The mechanism and its dual truss superimposed. b) The mechanism whose

joint velocities are depicted by arrows. c) The dual truss – the two face forces are shown within the faces outlined by loops.

Establishing the "face force" variable is not only important from the theoretical point of view, but has practical analysis applications as well (Shai 2002a).

It is interesting to note that Gabriel Kron once wrote: “It is easy to say ‘mesh current’ and let it go at that. But to say ‘force acting in a closed mesh’ requires a philosophical dissertation that would leave the mechanical engineer unresponsive.” (Kron 1963). It seems that the concept foreseen by Kron is actually the one appearing in this section.

4 Systematic solving of an engineering problem through graph representation transformations

In the previous section it was shown that knowledge transformations can be used to yield creative engineering solutions. Current section shows how these transformations can be monitored to provide a general methodology for solving some engineering problems, belonging both to analysis and design.

Following four subsections summarize the four general steps for solving problems by means of the proposed approach:

Step 1: Choosing the proper representation

First, it should be determined through which graph representation the problem is to be solved. This step is of major impact on the success of the whole process, since it determines which representation inherent knowledge is to be involved. Following are the main criteria considered when choosing the representation:

- The primary criterion for choosing the representation is the correspondence between the representation and the engineering domain in question. The correspondence between specific representations and engineering domains is given in the third column of Table 1.
- In many cases there is more than one representation that is applicable to represent the engineering domain. In these cases additional factors concerning the specific problem in question and the properties inherent in the representations are to be considered. An example for such a case can be seen in Table 1, where the representations PGR and PLGR are both applicable to represent mechanisms.
- Different representations enable exposing different properties existing in the engineering system, thus a criterion for choosing a representation are the properties relevant for solving the given engineering problem. For example, when the mechanism is represented by the PGR all of its engineering properties related to the linear velocities are exposed (Shai 2001a) while representing it by PLGR exposes the properties concerning the angular velocities (Shai, 2002b).

Step 2: Transforming the problem from the terminology of the engineering domain to the terminology of the graph representation.

The transformation from the terminology of the engineering domain to the terminology of the graph representation is performed through the representation construction rules (component 4 of the representation). Now the engineering problem becomes a problem in the representation, and all the knowledge embedded in the representation and in the interrelations between the representations become available.

Step 3: Solving the problem in the domain of the graph representations.

Once the problem possesses an abstract formulation in the terminology of graphs, it is matched with a corresponding solution, by means of one of the following three possible ways:

1. Employing the knowledge, methods and algorithms embedded in the representation to solve the problem.
2. Searching for problems from other engineering domains that possess the same formulation in this representation, and have known solutions. Then, transforming the found solution to the terminology of the representation.
3. Transforming the problem to other graph representations through the graph duality relations between them and solve the problem there.

In each of these three possibilities, the obtained solution consists of knowledge that was available for us ever since, but was not associated with our specific problem. All the three options can be carried out by applying the mathematically based tools described in the previous sections of the paper.

Step 4: Transferring the obtained solution from the terminology of the graph representations to the terminology of the engineering domain

This transformation is also performed by means of the graph construction rules. Once the transformation is performed, the solution for the original engineering problem is obtained.

The above four-step process is demonstrated on two different engineering problems: analysis problem (Example 6) and engineering design problem (Example 7). These two examples show that the process is indifferent of whether the problem is an analysis or a design problem, since both are transformed to problems in the graph representations.

4.1 Solving analysis problems through graph representation

During the representing process, the elements of the engineering system are associated with the edges or vertices of the graph representation, while the physical parameters of the elements are associated with the variables assigned to these vertices and edges. The purpose of the representation construction rules is to assure that the representation constructed from the engineering system is isomorphic to the system. This means that its implicit mathematical rules and properties map the physical laws and properties underlying the represented engineering system. Thus, the mathematical behavior of the graph representation that is deduced from the embedded mathematical knowledge of the representation models the actual behavior of the engineering system. Therefore,

although the behavior of the system is not considered during the representation process, it is automatically implicit in the representation.
Such approach significantly differs from the approach adopted in other works done in engineering design community. Gero, for example, used a distinct graph, called a 'behavioral graph' to express the behavior of the system(Qian and Gero, 1992).

This section describes the process of deducing the behavior of an integrated system, i.e. solving the engineering analysis problem, by means of the knowledge embedded in the graph representation.

Example 6. Analysis of MEMS comb-driven resonator.

Figure 16 shows an integrated MEMS system - micro resonator, comprising elements both from electrical and dynamical systems (Tang 1990). The problem is to model the behavior of the system, on the basis of the given system parameters and to derive the system equations.

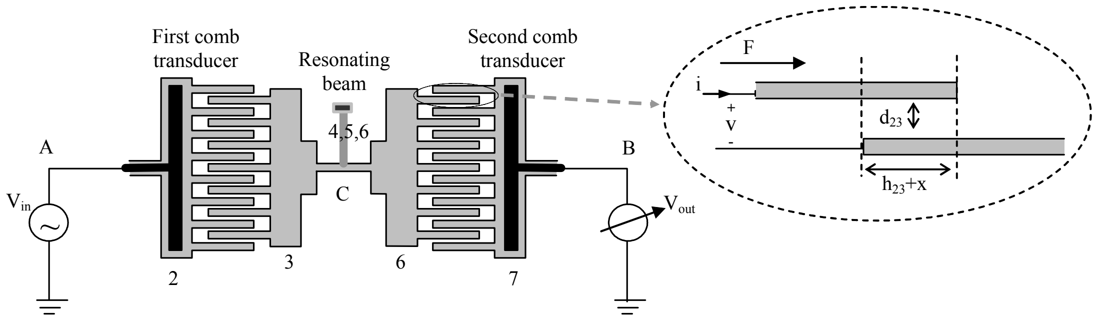


Figure 16. Comb-driven micro-resonator.

Step 1. Choosing the proper graph representation.

Since the system contains elements from two engineering domains, electricity and dynamics, the representation is to be applicable to both of these domains. According to Table 1 such a representation is the Resistance Graph Representation (Shai, 2001c). It should be noted that the ability of Resistance Graph Representation to represent a wide variety of engineering domains made it a good candidate for representing many integrated systems (Shai and Rubin, 2003).

Step 2. Transforming to the terminology of the graph representation.

Transforming the system of Figure 16 to the terminology of the resistance graph representation means constructing the representation of the system by means of its construction rules. Figure 17 shows the graph derived after applying the construction rules. The high coupling between the different parts of the resonator is expressed through matrix relations between the variables associated with edge couples 2,3 and 6,7, which represent, in the graph, the electromechanical transducers of the system.

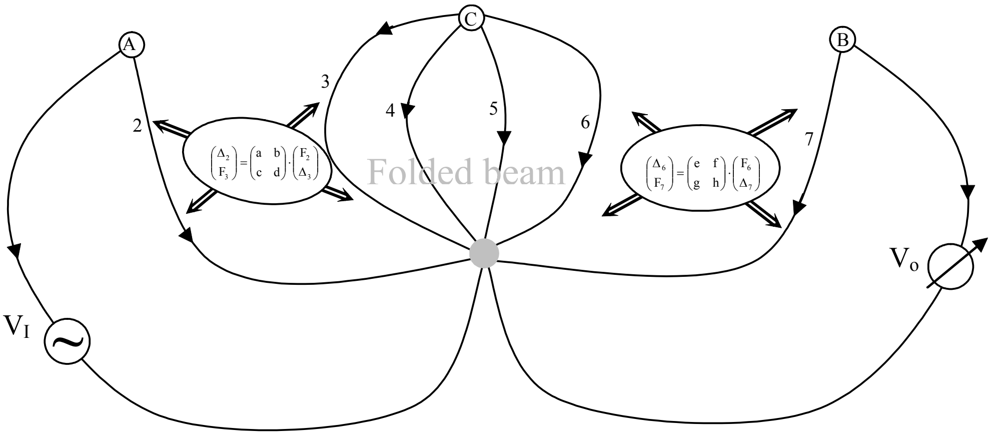


Figure 17. Resistance Graph Representation of the comb-driven micro resonator.

Step 3. Solving the problem in the graph representation.
The mixed variable method embedded in the resistance graph representation (Shai and Rubin 2003; Ta'aseh and Shai 2002) can be employed to derive the system equations of the comb-driven resonator - equations of the mixed variable method for the graph of Figure 17 appear in Figure 18.

$$\begin{pmatrix} a & 0 & b & 0 \\ 0 & e+R_6 & -1 & f \\ c & 1 & d+1/R_4+1/R_5 & 0 \\ 0 & g & 0 & h \end{pmatrix} \cdot \begin{pmatrix} F_2 \\ F_6 \\ \Delta_3 \\ \Delta_7 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \Delta_1$$

Figure 18. Mixed variable method applied to the graph of Figure 17. The independent variables of the graph are the flows in edges 2 and 6, and the potential differences in edges 3 and 7. The rest of the variables of the system are straightforwardly obtained from these four after applying the flow and potential laws (Table 1).

Step 4. Transforming the solution to the terminology of the engineering system.
Unlike other cases, performing this step in the current case can be considered rather trivial. The values obtained for the variables of the graph in step 3 are translated into the values of the corresponding variables of the engineering system.

As was mentioned above, one of the exceptionalities of this approach is that it is applied in a systematic way both for analysis and design problems, all in the same way. In the following section, the same steps will be applied to solve a design problem.

4.2 Solving design problems through graph representation transformations

Current section demonstrates obtaining a solution of a design engineering problem by transforming knowledge existing in electricity for solving the design problem in the field of mechanical engineering.

Example 7. Design a gear train where the direction of rotation of the output link is constant disregarding the direction of rotation of the input link, while the velocity magnitudes of both links are equal.

Step 1. Choosing the proper representation.
As can be seen from Table 1, gear trains have been represented by three different graph representations: PGR, PLGR and LGR. In the first representation the links of the gear train are the system elements thus are represented by edges, whereas the latter two representations deal more with the interconnections between the gears thus the relations between the engineering elements are represented by edges. Since the statement of the design problem explicitly concerns with the links of the system, it would be more convenient to express the problem statement through PGR, thus it is the preferred representation for this design problem.

Step 2. Transforming the problem into the terminology of the graph representation.

According to (Shai, 2001a), the problem stated in step 1, obtains in the terminology of PGR the following form: Find a potential graph with a potential difference source edge of arbitrary potential difference function, and an output edge in which the potential difference is equal to the absolute value of the potential difference in the source.

Figure 19 depicts the problem statements in the original and the transformed terminologies.

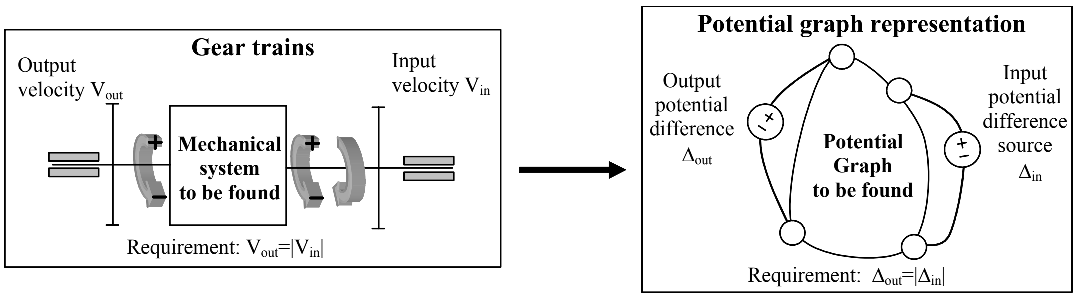


Figure 19. Transforming the problem from the terminology of gear trains to the terminology of the graph representation. In the latter the problem becomes a problem of finding a potential graph representation satisfying certain requirements.

Step 3. Solving the problem in the domain of the graph representation.

Instead of trying to solve the problem directly, the proposed approach suggests to search for an existent solution in other engineering domains, represented by the same representation. Then to transform the design solution from the secondary to the original domain through the Graph Representation. According to Table 1, since electronic circuits can also be represented by PGR, a search for existing solution is performed in the domain of electronics.

In terminology of electronic circuits the problem would possess the following form: find an electronic circuit having a voltage source with arbitrary voltage function and an output element, the voltage in which is equal to the absolute value of the voltage at the input.

The search for the electronic systems satisfying these requirements immediately yields several known electronic rectifier circuits (Smith, 1987). One of these circuits is the “bridge rectifier” (Figure 20).

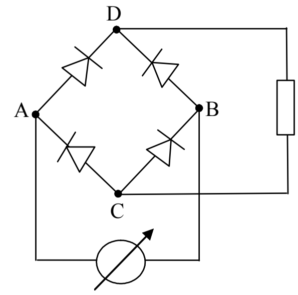


Figure 20. Bridge rectifier circuit.

Building the graph representation of the circuit of Figure 20, as shown in Figure 21, is done by applying the construction rules of this representation, where a unidirectional edge corresponds to a diode and a voltage generator to a potential source, as is explained in more details in (Shai, 2003). Obtaining this graph concludes Step 3.

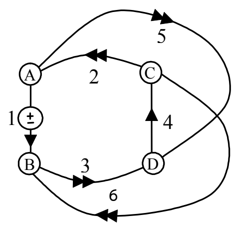


Figure 21. Potential Graph Representation of the bridge rectifier.

Step 4 . Building gear train from the graph representation.

The gear train satisfying the requirement of the problem can now be derived from its graph representation after applying a series of systematic steps based on the constructions rules, as appears in more details in (Shai, 2003). Figure 22a presents the mechanical gear train obtained as a result of applying this process. The mechanical rectifier has been built in the mechanism laboratory of Tel-Aviv University (see photograph in Figure 22b) and successfully tested.

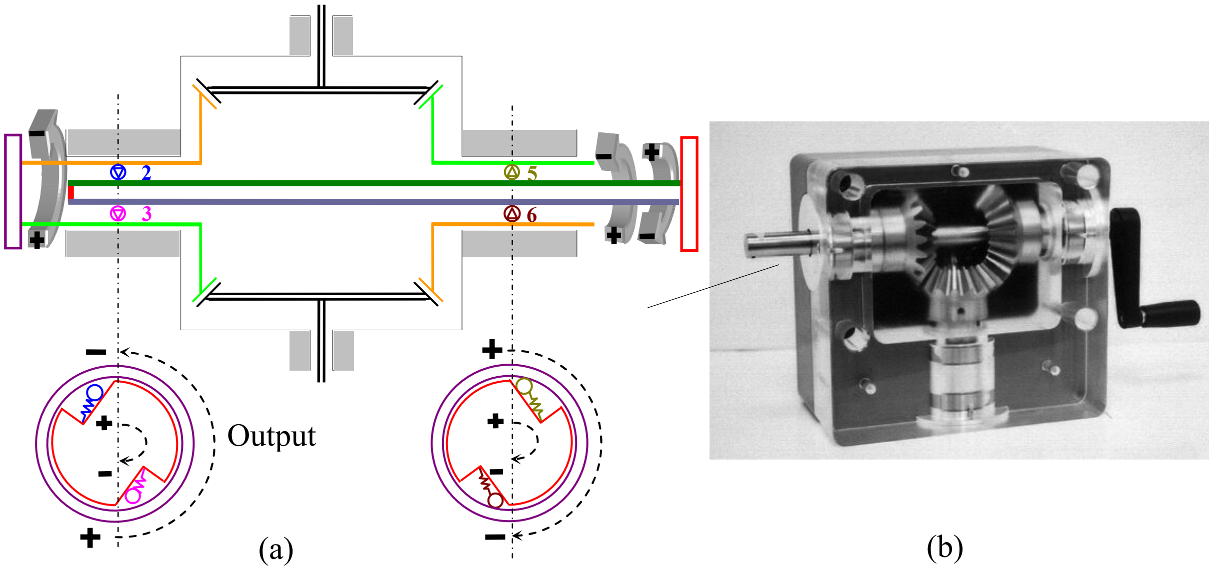


Figure 22. The mechanical rectifier device:
(a) The schematic description with side cuts of the unidirectional bearings, (b) The photograph of the device built in the laboratory. The input shaft '1' can be rotated in an arbitrary direction while the output shaft '4' rotates exclusively in counterclockwise direction.

Figure 23 depicts the process performed above by means of a diagram.

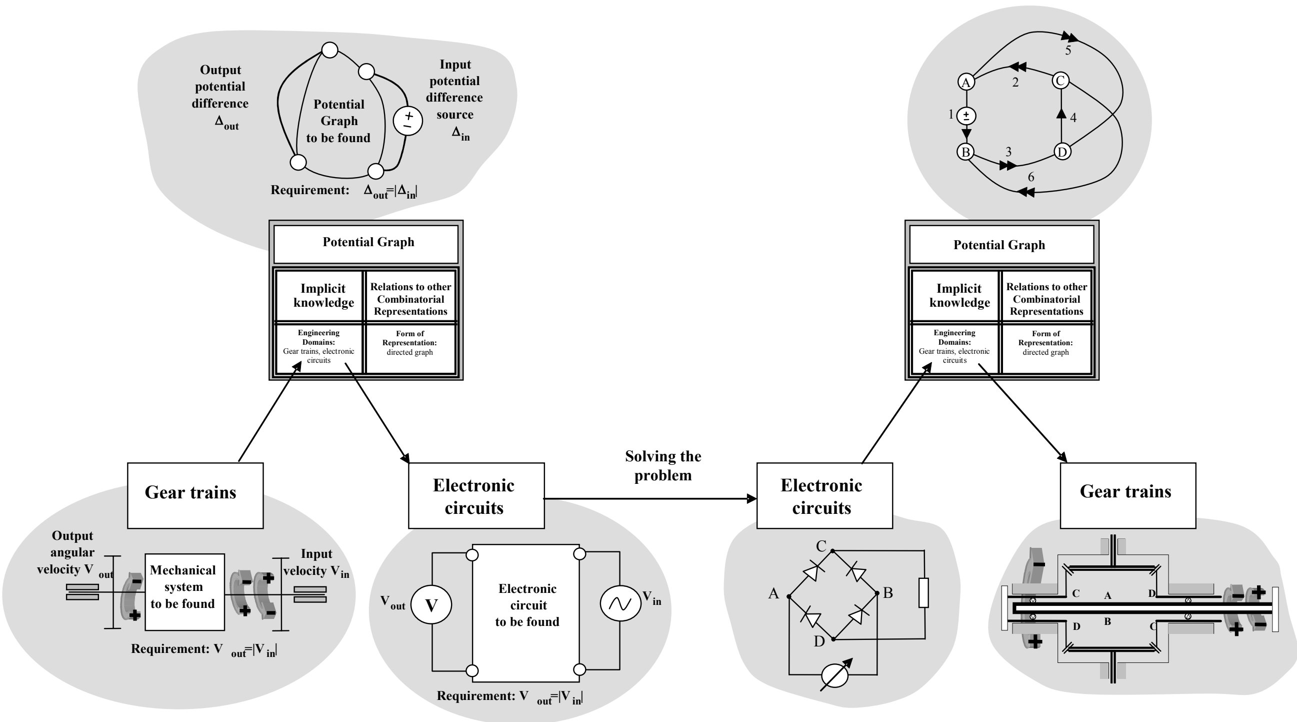


Figure 23. The process of design of mechanical rectifier through graph representation transformations: The problem formulation is transformed to the terminology of the graph representation and then to the terminology of electronic circuits for which a solution is then found. Then the solution is transformed to the graph representations and then back to the domain of gear trains.

5 Discussion

The paper has provided an overview over a general engineering problem solving approach based on transforming the engineering problems through graph representations linked to a variety of engineering domains. The approach provides a mathematically based mechanism for systematic transformation between different engineering domains. The domain specific knowledge was first transferred to the terminology of the corresponding graph representation, or in other words brought to a higher abstract mathematical level. Once at that level, the knowledge can be brought back down to some other engineering domain related to the representations, thus yielding authentic knowledge in this domain.

A strong correlation exists between the solving process suggested in the paper and the heuristic classification problem solving strategy of Clancey (Clancey, 1985). This correlation can be traced through all the four main steps of the process described in Section 4:

- Choosing the representation – is a preliminary step of choosing the type of abstraction to be employed for solution of a specific problem. This step is partially carried out through expert knowledge and rules associated with graph representations.
- Transforming the problem to the terminology of the graph representation – in this step that engineering problem is transformed to the graph representation level. The outcome of this step is the problem formulation in an abstract language of graph theory, thus it relates to the data abstraction stage in the Clancey's classification method.
- Solving the problem in the graph representation – the abstract formulation of the problem is associated or “matched” with a solution formulated in the terminology of the graph representation level, which is actually the abstract solution of the problem. This is done by applying the graph theory duality or transforming existent knowledge from some other engineering domain, or deduced from the mathematical knowledge embedded in the representation itself.
- Transforming the solution from the terminology of graph representations to the engineering domain – turning the abstract solution into the solution of the given engineering problem. In this step, the knowledge in its abstract form, terminology of graph representations, is refined to the physical level of the real engineering system.

Two possible routes of knowledge transfer were demonstrated in the paper. The first through a common graph representation, where the same graph representation was linked to the engineering domains between which the knowledge was transferred. The second route is between two mutually dual graph representations each related to other engineering domain. The validness of both possibilities is proved through the well-established mathematical basis underlying the graph representations, their duality, and their relations to the engineering domains.

Accordingly, graph representation transformations can be seen as a form of implementation of heuristic matching for a large set of engineering problems. The most significant advantage of such an implementation is the systematicity stemming from the mathematical foundation, upon which the methodology is established.

Unified perspective on analysis and design – In the paper, it was shown that the graph representations are isomorphic to the engineering systems, namely their mathematical behavior maps the physical behavior of the engineering system. Thus, analysis of the engineering system becomes a process of elaborating the behavior of the graph representation. In the paper this idea was demonstrated in section 4.1 where the behavior equations of a comb-driven micro resonator were derived from the knowledge embedded in its corresponding graph representation.
In this method, a design problem becomes a problem of searching for a graph representation, which exhibits the required abstract behavior. This idea is demonstrated in the paper in example 7, where the design task is to build a mechanical rectifier. After transforming the problem into the terminology of the graph representations, a graph which possesses the needed behavior was found by constructing a graph representation corresponding to a diode bridge rectifier electronic circuit.

Applicability of the method - The engineering domains to which this method is applicable are listed in Table 1. This list of engineering domains is not fixed and it is expanded with the advance of the presented research in two directions: representing more engineering domains with existent representations and devising new graph representations. Thus, much additional engineering domains are expected to be associated with graph representations, enlarging not only the range of problems that can be solved, but also the space of all possible abstract solutions.
As can be seen from the examples given throughout the paper, the problems covered so far by the approach relate exclusively to the physical behavior of engineering systems. Furthermore, the suggested method is founded on graph theory, which is a topic in

discrete mathematics known to be the mathematical foundation of computer science enabling it to handle extensive data amounts. Thus, despite the apparent simplicity of the provided examples, there is no limitation for the approach to be implemented in industrial scale applications.

References

Balabanian N. and Bickart T. A., 1969, *Electrical Network Theory*, New York, John Wiley & Sons.

Clancey W. J., 1985, "Heuristic Classifcation", *Artifical Intelligence*, Vol. 27, pp. 289-350.

Even S., 1979, *Graph Algorithms*, Computer Science Press.

Korf R. E., 1980, "Toward a Model of Representation Changes", *Artificial Intelligence*, Vol. 14, pp. 41-78.

Kron G., 1963, *Diakoptics – the piecewise solution of large-scale systems*, London, Macdonald.

Lovasz L. and Yemini Y., 1982, "On generic rigidity in the plane", *SIAM J. Algebraic and Discrete Methods*, 3, 91–98.
Newell A., 1966, "Limitations of the current stock of ideas about problem solving", in :Kent, A. and Taulbee, O.(Eds.), *Electronic Information Handling*, Spartan Books, Washington.
Nilsson N. J., 1971, *Problem-Solving Methods in Artificial Intelligence*, McGraw- Hill Books, NY,
Norton R. L., 1992, *Design of Machinery*, McGRAW-HILL.
Qian L. and Gero J.S., 1992, "A design support system using analogy", *Proc. Second International Conference on AI in Design*, pp. 795-813, Kluwer Academic Publishers.
Recski A., 1989, *Matroid Theory and its Applications in Electric Network Theory and in Statics*, Springer, Berlin.
Shai O., 2001a, "The Duality Relation between Mechanisms and Trusses", *Mechanism and Machine Theory*, Vol. 36, No. 3, pp. 343-369.
Shai O., 2001b, "The Multidisciplinary Combinatorial Approach and its Applications in Engineering", *AIEDAM – AI for Engineering Design, Analysis and Manufacturing*, Vol. 15, No. 2, pp.109-144.
Shai O., 2001c, "Deriving Structural Theorems and Methods Using Tellegen's Theorem and Combinatorial Representations", *International Journal of Solids and Structures*, Vol. 38, pp. 8037-8052.
Shai O., 2002a, "Utilization of the Dualism between Determinate Trusses and Mechanisms", *Mechanism and Machine Theory*, Vol. 37, No. 11, pp. 1307-1323.
Shai O., 2002b, "Duality between Statical and Kinematical Engineering Systems", *The Six International Conference on Computational Structures Technology,* Prague, Czech Republic, 4-6 September.
Shai O., 2003, "Design through Common Graph Representations", *ASME Design Engineering Technical Conferences*, Chicago, September 2-6, 2003
Shai O. and Preiss K., 1999, "Isomorphic Representations and Well-Formedness of Engineering Systems", *Engineering with Computers* 15, 303-314.
Shai O. and Reich Y., 2003, "Infused Design: a Brief Introduction", International Conference on Engineering Design ICED 03, Stockholm, August 19-21.
Shai O. and Rubin D., 2003, "Representing and Analyzing Integrated Engineering Systems through Combinatorial Representations", accepted for publication in *Engineering with Computers*.
Simon H. A., 1969, *The Sciences of the Artificial,* MIT Press, Cambridge.
Smith R., 1987, *Electronics: circuits and devices*, John Willey and Sons.
Swamy M.N. and Thulasiraman K., 1981, *Graphs*: *Networks and Algorithms*, John Wiley & Sons, NY.
Ta'aseh N. and Shai O., 2002, "Derivation of Methods and Knowledge in Structures by Combinatorial Representations", The Six *International Conference on Computational Structures Technology*, Prague, Czech Republic.
Tang W. C., 1990, *Electrostatic Comb Drive for Resonant Sensor and Actuator Applications*, Ph.d. dissertation, University of California.