Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2012, Article ID 510180, 16 pages
doi:10.1155/2012/510180

Research Article

Graph Products and Its Applications in Mathematical Formulation of Structures

M. Nouri,[1] S. Talatahari,[2] and A. Salimi Shamloo[3]

[1] *Department of Structural Engineering, Islamic Azad University, Shabestar Branch, Shabestar, Iran*
[2] *Marand Faculty of Engineering, University of Tabriz, Tabriz 51667 71719, Iran*
[3] *Department of Mathematics, Islamic Azad University, Shabestar Branch, Shabestar, Iran*

Correspondence should be addressed to M. Nouri, nouri@iaushab.ac.ir

Received 15 June 2012; Revised 15 October 2012; Accepted 16 October 2012

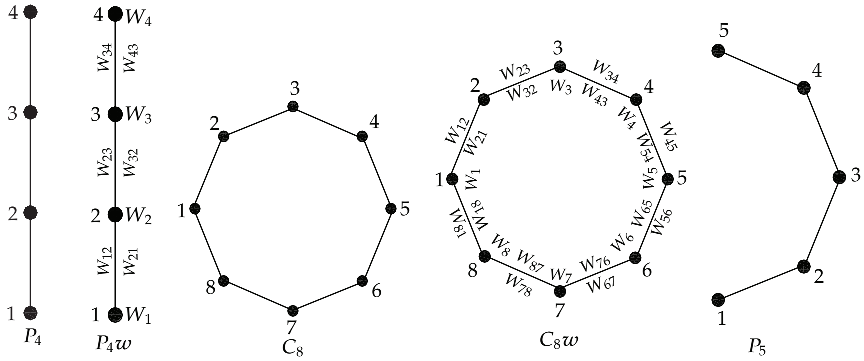
Academic Editor: C. Conca

The first step in the analysis of a structure is to generate its configuration. Different means are available for this purpose. The use of graph products is an example of such tools. In this paper, the use of product graphs is extended for the formation of different types of structural models. Here weighted graphs are used as the generators and the connectivity properties of different models are expressed in terms of the properties of their generators through simple algebraic relationships. In this paper by using graph product concepts and spatial structured matrices, a new algebraic closed form is proposed for mathematical formulation and presentation of structures. For clarification some examples are included.

1. Introduction

Data generation is the first step in the analysis of every structure. Configuration processing of large scale problems without automatic approaches can be erroneous and occasionally impossible. Formex configuration processing is one such a means introduced by Nooshin [1] and further developed by Nooshin et al. [2] and Nooshin and Disney [3]. Similar methods are developed based on set theory by Behravesh et al. [4]. Kaveh applied graph theory for this formation [5] (see also Kaveh et al. [6]). The use of product graphs in structural mechanics is suggested in [7, 8] and application of the corresponding concepts utilizing the directed and looped generators is due to Kaveh and Koohestani [9], weighted graph products by Kaveh and Nouri [10] and weighted triangular and circular graph products employed by Kaveh and Beheshti [11].

There are many other references in the field of data generation; however, most of them are prepared for specific classes of a problem. For example, many algorithms have been



(a) P_4 (b) P_4w (c) C_8 (d) C_8w (e) P_5

Figure 1: Examples of simple and weighted graphs.

developed and successfully implemented on mesh or grid generation; a complete review of which may be found in a paper by Thacker [12] and in the books by Thompson et al. [13], Liseikin [14], and Topping et al. [15].

In this paper the configuration processing of regular structures is considered. A structure is called *regular* if it can be considered as the product of two or three subgraphs (generators) [16]. The weighted graph products developed in [10] and their application are extended. Weighted paths and cycles are considered as the generators, and it is shown that many such product graphs can algebraically be expressed by simple relationships and a new algebraic closed form proposed for mathematical formulation and presentation of structures. Once this is done, then the existing methods can be applied to eigensolution and analysis of such structures [17–19]. However, this paper is limited to the generalization of product graphs for configuration processing of space structures. The methods of this paper can easily be adopted in the mesh generation of the finite element models.

2. Definitions from Graph Theory

A *graph* $S(N, M)$ consists of a set of elements, $N(S)$, called *nodes* and a set of elements, $M(S)$, called *members*, together with a relation of incidence which associates two distinct nodes with each member, known as its *ends*. If weights are assigned to the members and nodes of a graph, then it becomes a *weighted graph,* (Figure 1). Two nodes of a graph are called *adjacent* if these nodes are the end nodes of a member. A member is called *incident* with a node if that node is an end node of the member. The *degree* of a node is the number of members incident with that node. A *subgraph* S_i of a graph S is a graph for which $N(S_i) \subseteq N(S)$ and $M(S_i) \subseteq M(S)$, and each member of S_i has the same ends as in S. A *path graph* P is a simple connected graph with $N(P) = M(P) + 1$ that can be drawn in a way that all of its nodes and members lie on a single straight line. A path graph with n nodes is denoted by P_n, and a weighted path is shown by P_nw. A *cycle graph* C is a simple connected graph with identical number of nodes and members that can be drawn so that all of its nodes and members lie on a circle. A cycle graph with n nodes is shown by C_n, and a weighted cycle is denoted by C_nw. Examples of these graphs are shown in Figure 1. For further definitions the reader may refer to Kaveh [7, 20].

3. Algebraic Representation of Path and Cycles

Most of the space structures can be viewed as the product of some weighted paths and cycles. Therefore in this section some simple mathematical relationships are presented for defining such generators.

3.1. Weighted Path

The adjacency matrix of a path in general can be expressed as

$$\operatorname{adj}(P_n w) = \begin{bmatrix} W_1 & W_{1,2} & & & \\ W_{2,1} & W_2 & W_{2,3} & & \\ & \ddots & \ddots & \ddots & \\ & & W_{n-1,n-2} & W_{n-1} & W_{n-1,n} \\ & & & W_{n,n-1} & W_n \end{bmatrix}_{n\times n} = \begin{bmatrix} \ddots & \ddots & \ddots & \\ & L & D & U & \\ & & \ddots & \ddots & \ddots \end{bmatrix},$$
$$L = \begin{bmatrix} W_{2,1} \\ W_{3,2} \\ \vdots \\ W_{n-1,n-2} \\ W_{n,n-1} \end{bmatrix}_{(n-1)\times 1}, \quad D = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_{n-1} \\ W_n \end{bmatrix}_{n\times 1}, \quad U = \begin{bmatrix} W_{1,2} \\ W_{2,3} \\ \vdots \\ W_{n-2,n-1} \\ W_{n-1,n} \end{bmatrix}_{(n-1)\times 1}, \tag{3.1}$$

where the weights are divided into 3 groups, L, D, and U.

Using this definition a weighted path, in general, can be expressed as

$$P_n w \begin{bmatrix} L^T \\ D^T \\ U^T \end{bmatrix}. \tag{3.2}$$

3.2. Weighted Cycle

The adjacency matrix of a weighted cycle can similarly be expressed as

$$\operatorname{adj}(C_n w) = \begin{bmatrix} W_1 & W_{1,2} & & & W_{1,n} \\ W_{2,1} & W_2 & W_{2,3} & & \\ & \ddots & \ddots & \ddots & \\ & & W_{n-1,n-2} & W_{n-1} & W_{n-1,n} \\ W_{n,1} & & & W_{n,n-1} & W_n \end{bmatrix}_{n\times n} = \begin{bmatrix} \ddots & \ddots & \ddots & \\ & L & D & U & \\ & & \ddots & \ddots & \ddots \end{bmatrix},$$
$$L = \begin{bmatrix} W_{2,1} \\ W_{3,2} \\ \vdots \\ W_{n,n-1} \\ W_{1,n} \end{bmatrix}_{n\times 1}, \quad D = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_{n-1} \\ W_n \end{bmatrix}_{n\times 1}, \quad U = \begin{bmatrix} W_{1,2} \\ W_{2,3} \\ \vdots \\ W_{n-1,n} \\ W_{n,1} \end{bmatrix}_{n\times 1}, \tag{3.3}$$

where the weights are also divided into 3 groups L, D, and U.

Considering these, a weighted cycle, in general, can be shown as

$$C_n w \begin{bmatrix} L^T \\ D^T \\ U^T \end{bmatrix}. \tag{3.4}$$

3.3. Unit and Zero Vectors

The *unit vector* is defined as the following.

E_n is an n by 1 vector with all entries being 1. In addition $E_n(i)$ is a vector of the same dimension with all entries as 1 except the entry at ith row which is zero:

$$E_n = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1}, \qquad E_n(i) = \begin{bmatrix} 1 \\ \vdots \\ 0 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1} \longrightarrow i'\text{th row}. \tag{3.5}$$

The *zero vector* is defined as the following.

O_n is an n by 1 vector with all entries being 0. In addition $O_n(i)$ is a vector of the same dimension with all entries as 0 except the entry at the ith row which is 1:

$$O_n = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{n\times 1}, \qquad O_n(i) = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}_{n\times 1} \longrightarrow i'\text{th row}. \tag{3.6}$$

3.4. Extension of the Zero and Unit Vectors

In this section the zero and unit vectors are extended to represent $[L, D, U]$ in an efficient manner.

If we want to create a vector with some entries as 1 and the remaining also as 0, we use the following expression:

$$O_n(1, 2, i, k) = O_n(1) + O_n(2) + O_n(i) + O_n(k), \quad i, k \le n. \tag{3.7}$$

If we want to create a vector with the $I, i + k, i + 2k, \ldots$ as 1 and the remaining entries as 0, we use the following expression:

$$O_n(i : k) = O_n(i) + O_n(i + k) + O_n(i + 2k) + O_n(i + 3k) + \cdots + O_n(m), \quad n - k \le m \le n. \tag{3.8}$$

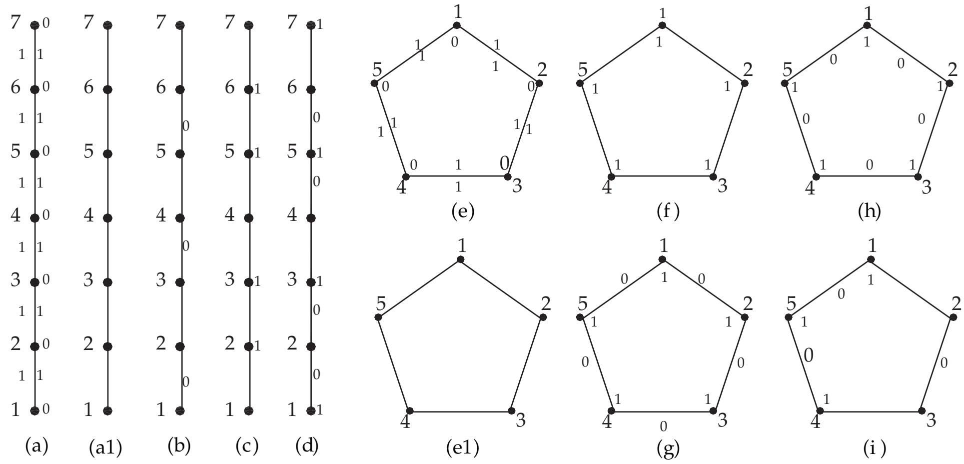


Figure 2: Some weighted graphs.

For creating a vector with the $I, i+k, i+2k, \ldots, i+m*k$ as 1 and the remaining entries as 0, we use the following expression:

$$O_n(i:k:m) = O_n(i) + O_n(i+k) + O_n(i+2k) + O_n(i+3k) + \cdots + O_n(i+mk), \quad i+mk \leq n \tag{3.9}$$

For creating a vector with the $(I, j, \ldots)$, $(i+k, j+k, \ldots)$, $(i+2k, j+3k, \ldots), \ldots$, and $(i+mk, j+mk, \ldots)$ as 1 and the remaining entries as 0, we use the following expression:

$$\begin{aligned} O_n&(i, j, \ldots : k : m) \\ &= \left[O_n(i) + O_n(j) + \cdots\right] + \left[O_n(i+k) + O_n(j+k) + \cdots\right] \\ &\quad + \left[O_n(i+2k) + O_n(j+2k) + \cdots\right] + \cdots + \left[O_n(i+mk) + O_n(j+mk) + \cdots\right]. \end{aligned} \tag{3.10}$$

In general case the following relation exists between the zero and unit vectors:

$$E_n(i) = E_n - O_n(i). \tag{3.11}$$

As an example the weighted graphs shown in Figure 2 are expressed in algebraic form.

In compact algebraic representation the difference between a simple and a weighted graph is illustrated. As an example, for Figures 2(a) and 2(a1), which are both simple paths, a is weighted and a1 is simple, the algebraic representations are as follows:

$$(\mathrm{a}) \Longrightarrow P_7 w \begin{bmatrix} L : E_6^T \\ D : O_7^T \\ U : E_6^T \end{bmatrix}, \qquad (\mathrm{a}_1) \Longrightarrow P_7. \tag{3.12}$$

Table 1: Operators of graph products.

Product	Operator
Cartesian	$\times$
Strong Cartesian	$\otimes$
Direct	$\bigcirc$

For the weighted case L, D, and U are E_6^T, O_7^T, E_6^T, respectively.

In Figure 2(i) a weighted cycle is shown, where L, D, and U are $E_5^T(4,5), O_5^T(1,4,5), E_5^T(2)$, respectively. In the following weighted paths and cycles of Figure 2 are represented algebraically:

$$(\mathrm{a}, \mathrm{a}_1) \Longrightarrow P_7 = P_7 w \begin{bmatrix} E_6^T \\ O_7^T \\ E_6^T \end{bmatrix}, \qquad (\mathrm{b}) \Longrightarrow P_7 w \Big[L : E_6^T(1:2)\Big], \qquad (\mathrm{c}) \Longrightarrow P_7 w \Big[D : O_7^T(2, 3:3)\Big],$$
$$(\mathrm{d}) \Longrightarrow P_7 w \begin{bmatrix} L : E_6^T(1, 2:3) \\ D : O_7^T(1:2) \end{bmatrix}, \qquad (\mathrm{e}, \mathrm{e}_1) \Longrightarrow C_5 = C_5 w \begin{bmatrix} E_5^T \\ O_5^T \\ E_5^T \end{bmatrix}, \qquad (\mathrm{f}) \Longrightarrow C_5 w \Big[D : E_5^T\Big],$$
$$(\mathrm{g}) \Longrightarrow C_5 w \begin{bmatrix} L : O_5^T \\ D : E_5^T \end{bmatrix}, \qquad (\mathrm{h}) \Longrightarrow C_5 w \begin{bmatrix} D : E_5^T \\ U : O_5^T \end{bmatrix}, \qquad (\mathrm{i}) \Longrightarrow C_5 w \begin{bmatrix} E_5^T(4,5) \\ O_5^T(1,4,5) \\ E_5^T(2) \end{bmatrix}. \tag{3.13}$$

4. Graph Products

In this section, weighted graph products which are introduced in [10] are formulated and generalized for configuration processing of structural models. These products are formulated in the algebraic form defined in Section 3. The operators used for each product are provided in Table 1.

Graph products of simple and weighted graphs are fully explained in [10]. For weighted case the first step is the formation of the coordinates of the nodes using the nodes of the generators. As an example, for two paths these nodes are generated in Figure 3.

4.1. Weighted Cartesian Product

In this product after the formation of the nodes according to the nodes of the generators (Figure 3), a member is added between two typical nodes (U_i, V_j) and (U_k, V_l), (Figure 4), if the following conditions are fulfilled.

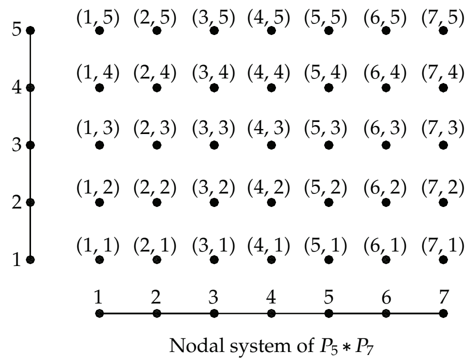


Figure 3: Nodal coordinate system of the Boolian product of P_7 and P_5.

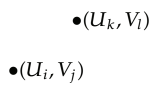


Figure 4: Two random nodes selected from a product domain.

We use the weights $-1, 0$, and $+1$ to assign to the nodes and elements in order to control the generation of the members and nodes:

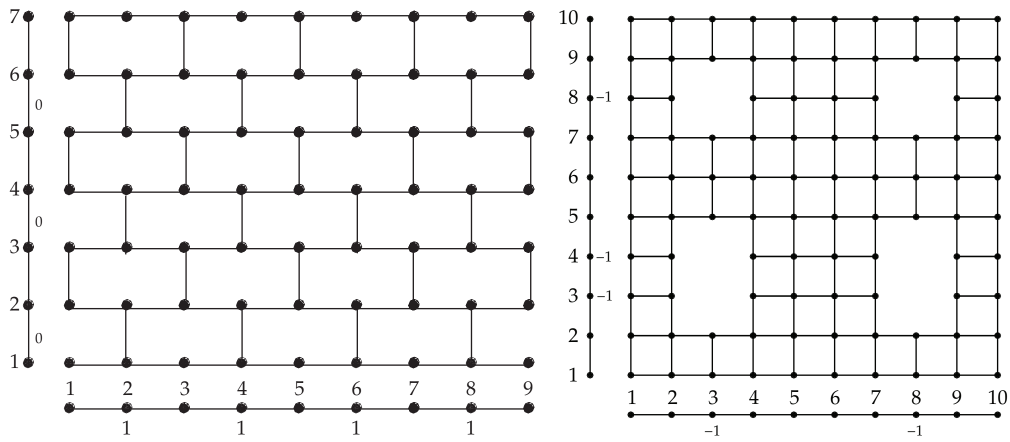
$$
\begin{aligned}
&\text{if } \left[\left(U_i = U_k \ \& \ W_i = -1 \ \& \ \left(W_j \ \& \ W_l\right) \neq -1 \ \& \ \left(W_{jl} \text{ or } W_{lj}\right) \neq 0\right) \text{ or}\right. \\
&\left.\left(V_j = V_l \ \& \ W_j = -1 \ \& \ (W_i \ \& \ W_k) \neq -1 \ \& \ (W_{ik} \text{ or } W_{ki}) \neq 0\right)\right] \text{ or} \\
&\text{if } \left[\left(U_i = U_k \ \& \ W_i = 0 \ \& \ \left(W_{jl} \ \& \ W_{lj}\right) \neq 0\right) \text{or} \left(V_l = V_j \ \& \ W_j = 0 \ \& \ (W_{ik} \ \& \ W_{ki}) \neq 0\right)\right] \text{ or} \\
&\text{if } \left[\left(U_i = U_k \ \& \ W_i = 1 \ \& \ \left(W_{lj} x \text{ or } W_{jl}\right) \neq 0\right) \text{or} \left(V_l = V_j \ \& \ W_j = 1 \ \& \ (W_{ik} x \text{ or } W_{ki}) \neq 0\right)\right].
\end{aligned}
\tag{4.1}
$$

As an example, Figure 5 illustrates some weighted Cartesian products and their compact representation.

4.2. *Weighted Strong Cartesian Product*

In this product after the formation of the nodes, according to the nodes of the generator, (Figure 3), a member is added between two typical nodes (U_i, V_j) and (U_k, V_l), (Figure 4), if the following conditions are fulfilled:

$$
\begin{aligned}
&\text{if } \left(U_i = U_k \ \& \ \left(W_{jl} \text{ or } W_{lj}\right) \neq 0\right) \ \& \ \left[W_i = 0 \text{ or } \left(W_i = -1 \ \& \ \left(W_j \ \& \ W_l\right) \neq -1\right)\right] \text{ or} \\
&\text{if } \left(V_j = V_l \ \& \ (W_{ik} \text{ or } W_{ki}) \neq 0\right) \ \& \ \left[W_j = 0 \text{ or } \left(W_j = -1 \ \& \ (W_i \ \& \ W_k) \neq -1\right)\right] \text{ or} \\
&\qquad \left((W_{ik} \text{ or } W_{jl}\right) \neq 0 \text{ or } \left(W_{ki} \ \& \ W_{lj}\right) \neq 0).
\end{aligned}
\tag{4.2}
$$



(a) $P_9w[D : O_9^T(2:2)] \times P_7w[L : E_6^T(1:2)]$ (b) $P_{10}w[D : -O_{10}^T(3,8)] \times P_{10}w[D : -O_{10}^T(3,4,8)]$

Figure 5: Examples of two weighted Cartesian products.

Examples of strong Cartesian products of weighted graphs and their compact presentations are provided in Figure 6. As it can be observed, the compact products of paths and/or cycles are a powerful means for configuration processing and can be employed similarly to Formex configuration processing of Nooshin [1].

4.3. *Weighted Direct Product*

In this product after the formation of the nodes according to the nodes of the generator (Figure 3), a member is added between two typical nodes (U_k, V_l) and (U_i, V_j), (Figure 4), if the following conditions are fulfilled:

$$\text{if } \left[\left((W_i \ \& \ W_j) \neq -1 \text{ or } (W_k \ \& \ W_l) \neq -1\right) \ \& \ \left((W_{ik} \ \& \ W_{jl}) \neq 0 \text{ or } (W_{ki} \ \& \ W_{lj}) \neq 0\right)\right]. \tag{4.3}$$

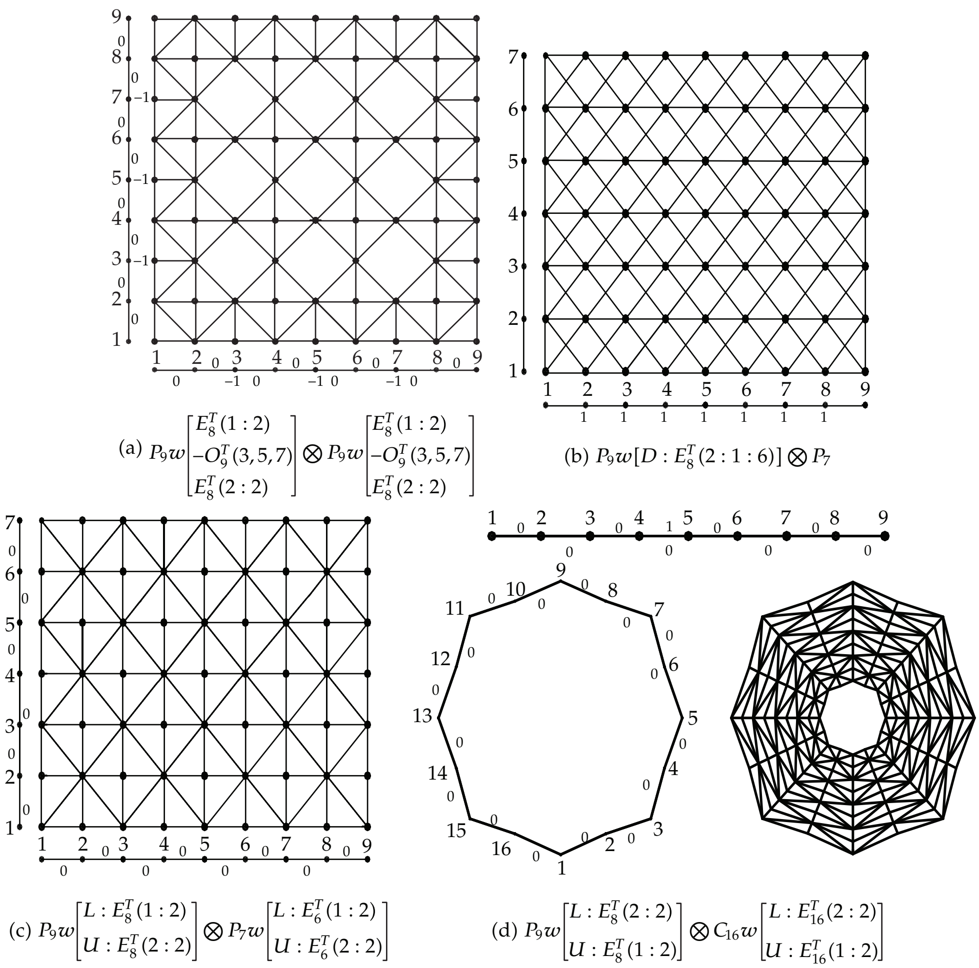
Some examples of these weighted products and their compact representations are illustrated in Figure 7.

5. Geometrical Transformation of Graph Products

In this section using simple transformations, the weighted graph products of the previous section are employed for configuration processing of different types of space structures.

5.1. *Transformation between Cartesian Coordinate System and Oblique System*

In Cartesian coordinate systems (or rectangular coordinates), the “address” of a point P is given by two real numbers indicating the positions of the perpendicular projections from



(a) $P_9w\begin{bmatrix}E_8^T(1:2)\\-O_9^T(3,5,7)\\E_8^T(2:2)\end{bmatrix}\otimes P_9w\begin{bmatrix}E_8^T(1:2)\\-O_9^T(3,5,7)\\E_8^T(2:2)\end{bmatrix}$

(b) $P_9w[D:E_8^T(2:1:6)]\otimes P_7$

(c) $P_9w\begin{bmatrix}L:E_8^T(1:2)\\U:E_8^T(2:2)\end{bmatrix}\otimes P_7w\begin{bmatrix}L:E_6^T(1:2)\\U:E_6^T(2:2)\end{bmatrix}$

(d) $P_9w\begin{bmatrix}L:E_8^T(2:2)\\U:E_8^T(1:2)\end{bmatrix}\otimes C_{16}w\begin{bmatrix}L:E_{16}^T(2:2)\\U:E_{16}^T(1:2)\end{bmatrix}$

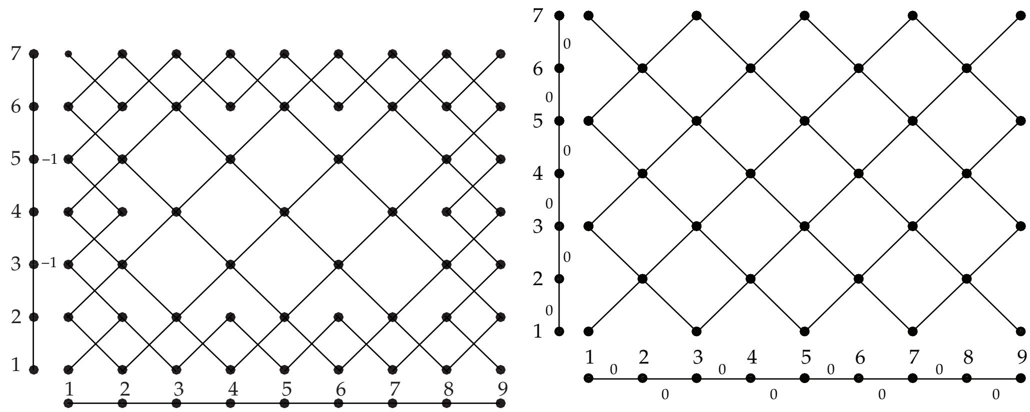
Figure 6: Different weighted strong Cartesian products of two simple weighted graphs.

the point to two fixed perpendicular lines, known as the x-axis and the y-axis, and we write $P = (x, y)$, (Figure 8).

In this figure $P = (4, 3)$, $Q = (-1.3, 2.5)$, $R = (-1.5, -1.5)$, $S = (3.5, -1)$, and $T = (4.5, 0)$. The axes divide the plane into four quadrants: P is in the first quadrant, Q in the second, R in the third, and S in the fourth. T is on the positive x-axis.

The following generalization of Cartesian coordinates is useful for configuration processing of space structures. Consider two axes, intersecting at the origin but not necessarily perpendicularly. Let the angle between these axes be ω. In this system of oblique coordinates, a point P is given by two real numbers indicating the positions of the projections from the point to each axis, in the direction of the other axis (Figure 9). The first axis (x-axis) is generally drawn horizontally. The case $\omega = 90°$ yields a Cartesian coordinate system.

In this coordinate system we have $P = (4, 3)$, $Q = (-1.3, 2.5)$, $R = (-1.5, -1.5)$, $S = (3.5, -1)$, and $T = (4.5, 0)$. Compare to Figure 8.



(a) $P_9w[D : -O_9^T(3,5,7)] \circ P_7w[D : -O_6^T(3,5)]$

(b) $P_9w\begin{bmatrix} L : E_6^T(2:2) \\ U : E_6^T(1:2) \end{bmatrix} \circ P_7w\begin{bmatrix} L : E_8^T(2:2) \\ U : E_8^T(1:2) \end{bmatrix}$

Figure 7: Examples of weighted direct products: product of $P_7w \circ P_9w$.

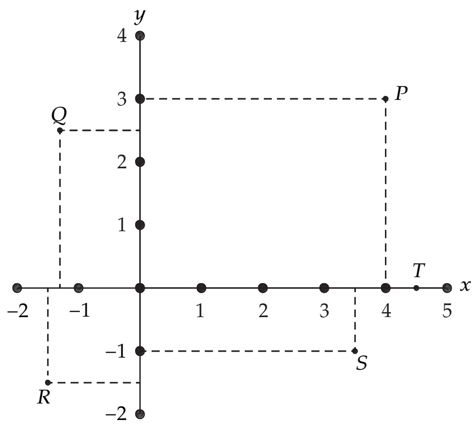


Figure 8: Points in the Cartesian coordinate system.

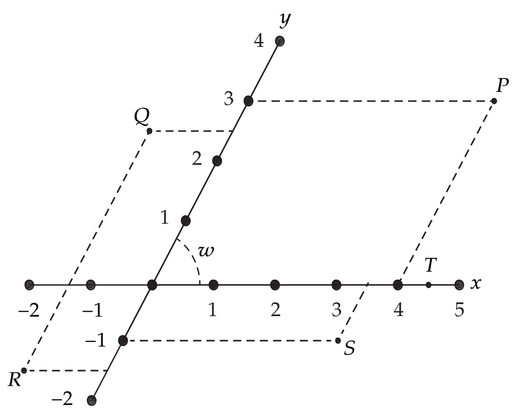
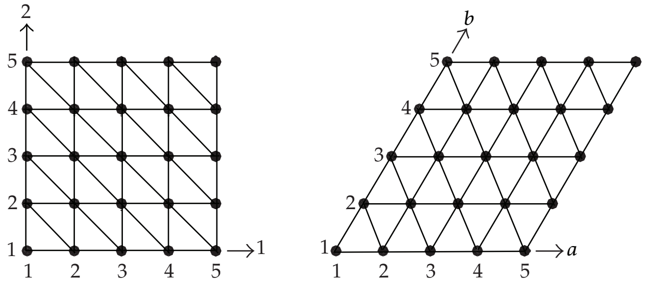
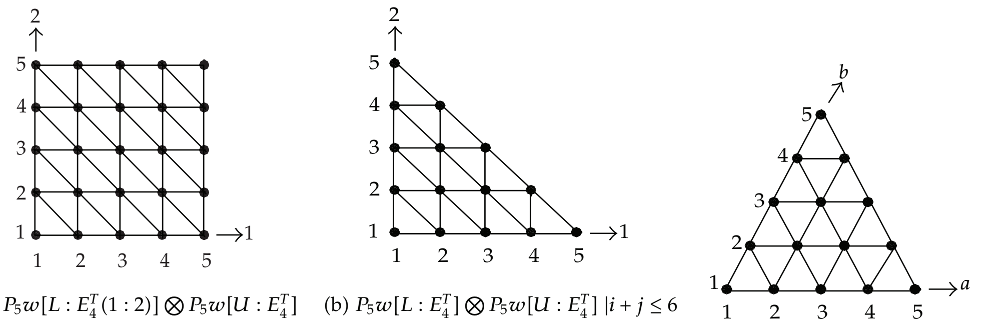


Figure 9: Points in an oblique coordinate system.



$P_5w[L : E_4^T] \otimes P_5w[U : E_4^T]$

Figure 10: Transformation between Cartesian and oblique coordinate systems.



(a) $P_5w[L : E_4^T(1:2)] \otimes P_5w[U : E_4^T]$ (b) $P_5w[L : E_4^T] \otimes P_5w[U : E_4^T] \mid i + j \leq 6$

(c) In oblique coordinate system

Figure 11: Geometrical conditions and transformation between Cartesian and oblique coordinate systems applied to a weighted graph product.

Connectivity and topological properties of a graph do not depend on its view in a coordinate system. One can present a graph with the same connectivity and different shapes in a different coordinate system.

We use Cartesian and oblique coordinate systems and the transformation between these systems for configuration processing of the space structures, as illustrated in Figure 10.

5.2. Coordinate Conditions

Additing or restricting the conditions on the domains of the weighted graph products result in different configurations. As an example, additing of the condition $(i + j \leq 6)$ on $P_5w[L : E_4^T] \otimes P_5w[U : E_4^T]$ and transforming the coordinate system, one can obtain the configuration shown in Figure 11.

5.3. Stretching of Nodal Point

Moving certain nodes in a graph model can produce different suitable configurations. Examples of such operations from [10] are shown in Figure 12.

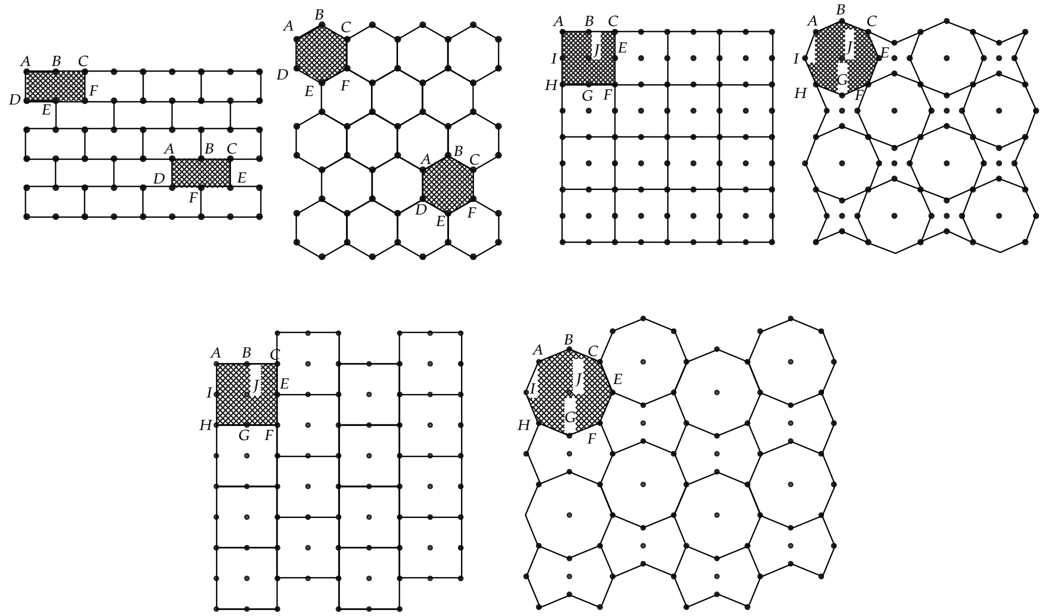


Figure 12: Suitable transformations of nodal coordinates.

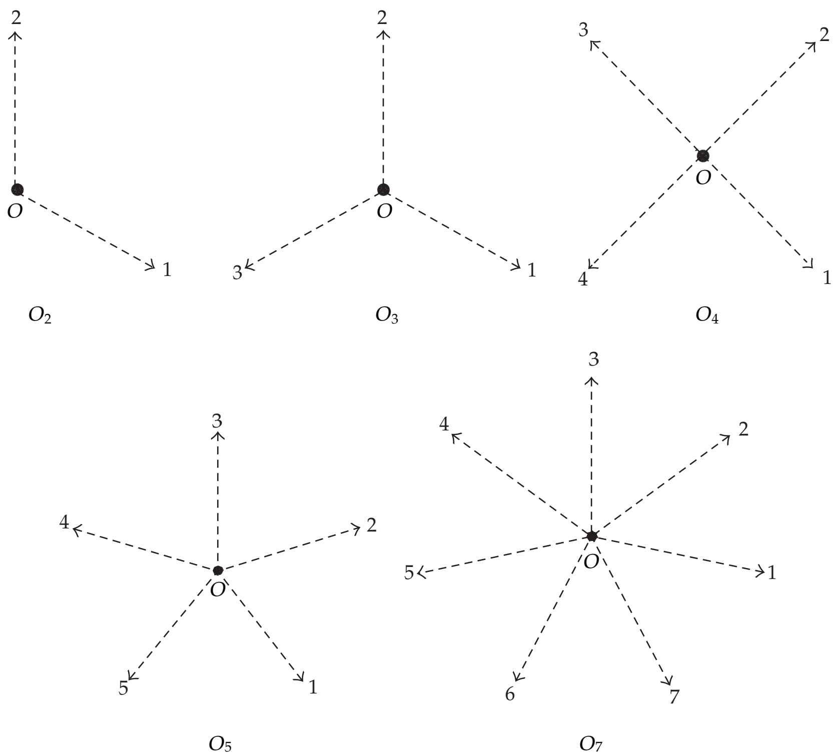


Figure 13: Generalized coordinate systems.

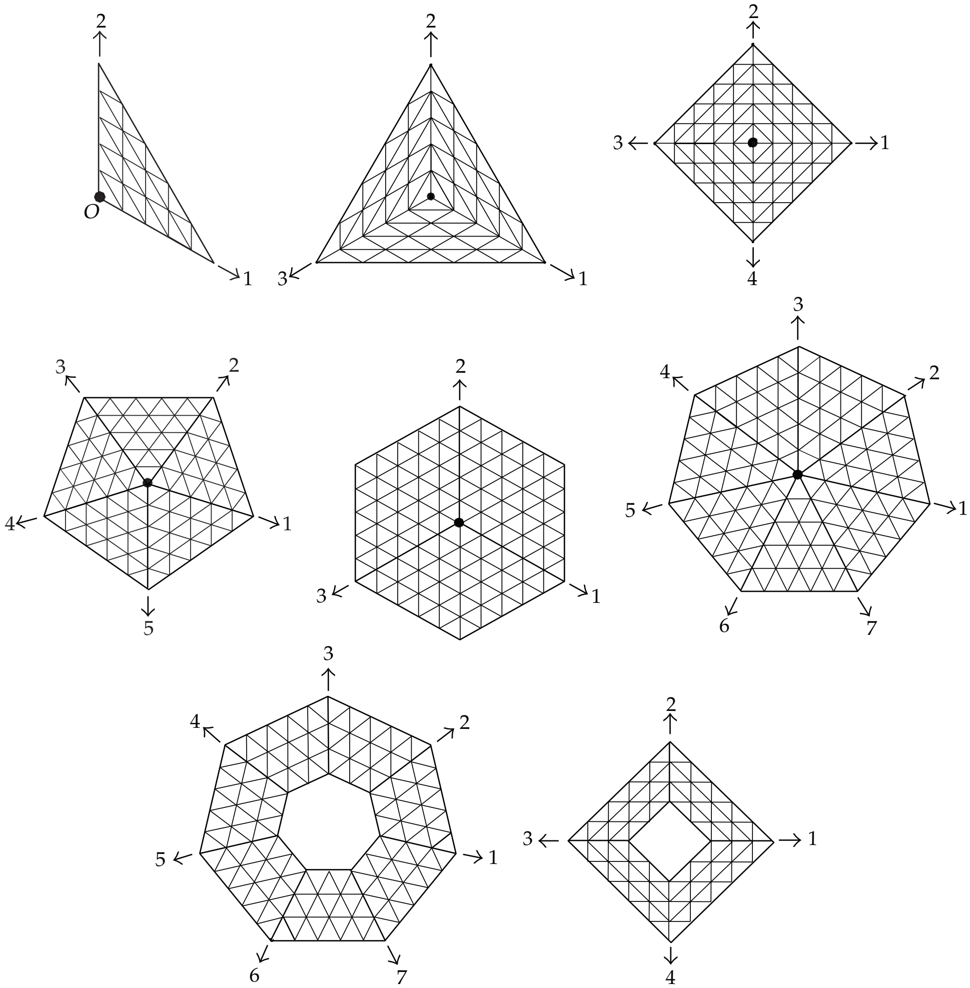


Figure 14: Weighted graph products in the shown coordinate system.

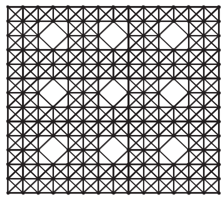
6. Generalized Weighted Graph Products

In this section using the previously defined products, transforming the coordinate systems, moving the nodes, adding new conditions to the conditions of different graph products, and also using generalized coordinate systems, the domain of the applications of graph products in configuration processing of space structures is extended.

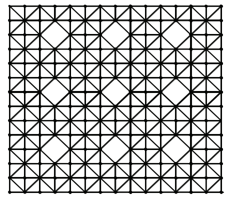
For configuration processing using the graph products, we extend the forms by defining the coordinate systems shown in Figure 13.

Product of adjacent axes of each coordinate system's new weighted graph products can be produced. As an example some products of this kind are illustrated in Figure 14. The algebraic form of each configuration is shown in Table 2.

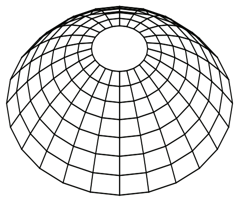
Figure 14(a) is obtained by the multiplication of axis 1 and axis 2, where the characteristics of the axes are shown in algebraic form in Table 2. Figure 14(b) is formed by multiplication of axis 1 by 2, axis 2 by 3, and axis 3 by 1. The remaining configurations of



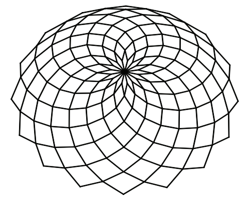
(a) $P_{15}w[D : O_{15}^T(4:4)] \otimes P_{15}w[D : O_{15}^T(4:4)]$



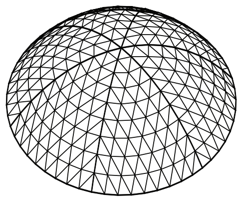
(b) $P_{16}w\begin{bmatrix} E_{15}^T(1:2) \\ -O_{16}^T(4:4) \\ E_{15}^T(2:2) \end{bmatrix} \otimes P_{15}w\begin{bmatrix} E_{15}^T(1:2) \\ -O_{16}^T(4:4) \\ E_{15}^T(2:2) \end{bmatrix}$



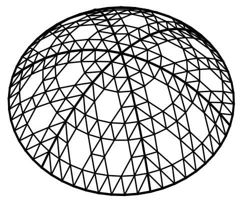
(c) $P_8 \times C_{24}$



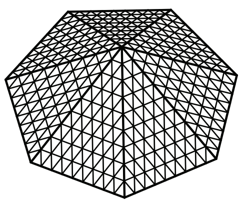
(d) $C_{32}w\begin{bmatrix} L : E_{32}^T(1:2) \\ U : E_{32}^T(2:2) \end{bmatrix} \circ P_{11}w\begin{bmatrix} L : E_{10}^T(1:2) \\ U : E_{10}^T(2:2) \end{bmatrix}$



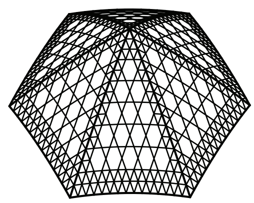
(e) $O_6 : P_{10}w[E : O_9^T] \otimes P_{10}w[U : O_9^T] | i + j \leq 10$



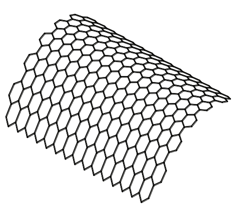
(f) $O_6 : P_{10}w\begin{bmatrix} E : O_9^T \\ D : O_9^T(5,8) \end{bmatrix} \otimes P_{10}w\begin{bmatrix} L : O_9^T \\ D : O_9^T(5,8) \end{bmatrix} | i + j \leq 10$



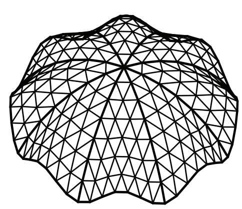
(e) $O_6 : P_{10}w[E : O_9^T] \otimes P_{10}w[U : O_9^T] | i + j \leq 10$



(h) $O_6 : P_{16}w\begin{bmatrix} E : O_{15}^T \\ D : O_{16}^T(5:2:4) \end{bmatrix} \otimes P_{16}w\begin{bmatrix} U : O_{15}^T \\ D : O_{16}^T(5:2:4) \end{bmatrix} | i + j \leq 16$



(i) $P_{32}w[D : O_{32}^T(1:2)] \times P_{12}w[L : E_{11}^T(1:2)]$



(j) $O_6 : P_{10}w[E : O_9^T] \otimes P_{10}w[U : O_9^T] | i + j \leq 10$

Figure 15: Different configurations in generalized graph products.

Table 2: Compact algebraic representation of graph products presented in Figure 14.

(a)	$O_2 : P_6w[L : E_5^T] \otimes P_6w[U : E_5^T] \mid i + j \leq 6$
(b)	$O_3 : P_6w[L : E_5^T] \otimes P_6w[U : E_5^T] \mid i + j \leq 6$
(c)	$O_4 : P_6w[L : E_5^T] \otimes P_6w[U : E_5^T] \mid i + j \leq 6$
(d)	$O_5 : P_6w[L : E_5^T] \otimes P_6w[U : E_5^T] \mid i + j \leq 6$
(e)	$O_3 : P_6w[L : E_5^T] \otimes P_6w[U : E_5^T]$
(f)	$O_7 : P_6w[L : E_5^T] \otimes P_6w[U : E_5^T] \mid i + j \leq 6$
(g)	$O_7 : P_6w[L : E_5^T] \otimes P_6w[U : E_5^T] \mid 3 < i + j < 6$
(h)	$O_4 : P_6w[L : E_5^T] \otimes P_6w[U : E_5^T] \mid 3 < i + j < 6$

Figure 14 are obtained similarly by multiplication of each pair of adjacent axes. The properties of the axes are provided in Table 2.

The mathematical formulations of the configurations in Figure 14 are provided in Table 2. In these relationships the type of the coordinate system, the generators, the type of the products, and the imposed conditions are provided.

7. Examples

In this section, the generalized weighted graph products examples of different configurations are formulated. First the configuration is formed and then appropriate geometric transformations are imposed to generate the final configuration of the models.

Examples of Cartesian, strong Cartesian, and direct products are illustrated in Figure 15. For each case, the compact formulation is provided underneath the corresponding figure.

8. Conclusions

In this paper the graph products and their applications in configuration processing are extended. Topology of a structure is viewed as the product of two weighted subgraphs like paths and/or cycles as its generators. The paths and cycles are formulated in a mathematical form, and the configuration of a space structure is expressed as different products of these weighted subgraphs as one expression. In the presented method the topological information of space structures can be stored as simple algebraic relationships. More complex configurations can be formulated using different graph theory operators and new conditions can be added to the domains of the products. The application of the introduced products of weighted graphs can also be extended to the mesh generation of finite element models.

Acknowledgment

M. Nouri is grateful for the support of the Shabestar Branch, Islamic Azad University.

References

[1] H. Nooshin, "Algebraic representation and processing of structural configurations," *Computers and Structures*, vol. 5, no. 2-3, pp. 119–130, 1975.

[2] H. Nooshin, P. Disney, and C. Yamamoto, *Formian*, Multi-Science Publishers, Essex, UK, 1993.
[3] H. Nooshin and P. Disney, "Formex configuration processing III," *International Journal of Space Structures*, vol. 17, no. 1, pp. 1–50, 2002.
[4] A. Behravesh, A. Kaveh, M. Nani, and S. Sabet, "A set theoretical approach for configuration processing," *Computers & Structures*, vol. 30, no. 6, pp. 1293–1302, 1988.
[5] A. Kaveh, "A graph-theoretical approach to configuration processing," *Computers and Structures*, vol. 48, no. 2, pp. 357–363, 1993.
[6] A. Kaveh, X. Jia, and Q. Weng, "Rotation as a general operation for configuration processing," *Scientia Iranica*, vol. 17, no. 2, pp. 131–140, 2010.
[7] A. Kaveh, *Optimal Structural Analysis*, John Wiley & Sons, Somerset, UK, 2nd edition, 2006.
[8] A. Kaveh and H. Rahami, "An efficient method for decomposition of regular structures using graph products," *International Journal for Numerical Methods in Engineering*, vol. 61, no. 11, pp. 1797–1808, 2004.
[9] A. Kaveh and K. Koohestani, "Graph products for configuration processing of space structures," *Computers and Structures*, vol. 86, no. 11-12, pp. 1219–1231, 2008.
[10] A. Kaveh and M. Nouri, "Weighted graph products for configuration processing of planar and space structures," *International Journal of Space Structures*, vol. 24, no. 1, pp. 13–26, 2009.
[11] A. Kaveh and S. Beheshti, "Weighted triangular and circular graph products for configuration processing," *Periodica Polytechnica*, vol. 56, pp. 1–9, 2012.
[12] W. C. Thacker, "A brief review of techniques for generating irregular computational grids," *International Journal for Numerical Methods in Engineering*, vol. 15, no. 9, pp. 1335–1341, 1980.
[13] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation, Foundations and Applications*, Elsevier, Amsterdam, The Netherlands, 1985.
[14] V. D. Liseikin, *Grid Gneration Methods*, Scientific Computation, Springer, Berlin, Germany, 1999.
[15] B. H. V. Topping, J. Muylle, and P. Ivanyi, *Finite Element Mesh Generation*, Saxe Coburg, Pubn, UK, 2004.
[16] W. Imrich and S. Klavžar, *Product Graphs*, Structure and Recognition, Wiley-Interscience, New York, NY, USA, 2000.
[17] A. Kaveh and H. Rahami, "A unified method for eigendecomposition of graph products," *Communications in Numerical Methods in Engineering*, vol. 21, no. 7, pp. 377–388, 2005.
[18] A. Kaveh, M. Nikbakht, and H. Rahami, "Improved group theoretic method using graph products for the analysis of symmetric-regular structures," *Acta Mechanica*, vol. 210, no. 3-4, pp. 265–289, 2010.
[19] A. Kaveh and H. Rahami, "An efficient analysis of repetitive structures generated by graph products," *International Journal for Numerical Methods in Engineering*, vol. 84, no. 1, pp. 108–126, 2010.
[20] A. Kaveh, *Structural Mechanics: Graph and Matrix Methods*, Research Studies Press, Somerset, UK, 3rd edition, 2004.