Dragoš Cvetković, Tatjana Davidović

MULTIPROCESSOR INTERCONNECTION NETWORKS

Abstract. Homogeneous multiprocessor systems are usually modelled by undirected graphs. Vertices of these graphs represent the processors, while edges denote the connection links between adjacent processors. Let G be a graph with diameter D, maximum vertex degree Δ, the largest eigenvalue λ_1 and m distinct eigenvalues. The products $m\Delta$ and $(D+1)\lambda_1$ are called the tightness of G of the first and second type, respectively. In the recent literature it was suggested that graphs with a small tightness of the first type are good models for the multiprocessor interconnection networks. We study these and some other types of tightness and some related graph invariants and demonstrate their usefulness in the analysis of multiprocessor interconnection networks. A survey of frequently used interconnection networks is given. Load balancing problem is presented. We prove that the number of connected graphs with a bounded tightness is finite and we determine explicitly graphs with tightness values not exceeding 9. There are 69 such graphs and they contain up to 10 vertices. In addition we identify graphs with minimal tightness values when the number of vertices is $n = 2, \ldots, 10$.

Mathematics Subject Classification (2010): 05C50, 68M07, 68M10, 68M14

Keywords: Multiprocessor Systems, Interconnection Topologies, Load Balancing, Spectra of Graphs, Graph Invariants.

Contents

1. Introduction 36
2. Load balancing 37
3. Various types of tightness of a graph 39
4. A survey of frequently used interconnection networks 43
5. Complete quasi-regular trees 45
6. Graphs with small tightness values 49
7. Graphs with smallest tightness values 56
References 57
Appendix 59

1. Introduction

Usual models for multiprocessor interconnection networks [20] are (undirected, connected) graphs [31, 33]. Vertices of these graphs represent the processors, while edges denote the connection links between neighboring (adjacent) processors. The processors within a multiprocessor system communicate by sending or receiving messages through these communication links. The two main parameters of the graph that play an important role in the design of multiprocessor topologies are maximum vertex degree Δ and the diameter D. In other words, Δ directly corresponds to the number of neighboring processors (adjacent vertices in the graph model), while D represents the length of the longest path in processor graph, i.e. maximum distance between two processors. The main drawback of multiprocessor systems is the communication overhead [4, 35], the time required to exchange data between different processing units. Therefore, interconnection networks have to satisfy two contradictory properties: to minimize the "number of wires" and to maximize the data exchange rate. This means that the paths connecting each two processors have to be as short as possible while the average number of connections per processor has to be as small as possible.

Recently, the link between the design of multiprocessor topologies and the theory of graph spectra [14] has been recognized [19]. The general idea of using graph eigenvalues in multiprocessor interconnection networks can be also found in [30]. The main conclusion of [19] is that the product of the number m of distinct eigenvalues of a graph adjacency matrix and Δ has to be as small as possible. We call this product the *tightness of the first type* for a graph. In [6] we introduced the *tightness of the second type* as the product $(D+1)\lambda_1$, where λ_1 is the largest eigenvalue of the graph. Moreover, we defined some other types of graph tightness, and investigated the relation between the tightness values and the suitability of the corresponding multiprocessor architecture. We showed that the graphs with a small tightness of the second type are suitable for the design of multiprocessor topologies.

In the paper [5] we determined explicitly graphs with tightness values not exceeding $a = 9$. To explain why the value 9 has been chosen, note first that by

Theorem 1 the number of connected graphs with a bounded tightness is finite. If the selected upper bound a is high, the number of corresponding graphs could be very big and some of these graphs may have large number of vertices. It turned out that the value $a = 9$ is very suitable: i) it is big enough to include the Petersen graph (Fig. 13), known to be a very good interconnection network (see, for example, [37]), and ii) it is small enough so that only 69 graphs obey the bound with the number of vertices in these graphs not exceeding 10.

For basic definitions and some general results in the theory of graph spectra the reader is referred to the introductory chapter of this publication.

The paper is organized as follows. Section 2 is devoted to relations between the load balancing problem and the theory of graph spectra. Definitions and basic properties of various types of tightness are given in Section 3. Section 4 contains a survey of frequently used multiprocessor interconnection networks. Some results on a special class of trees in the role of interconnection networks are given in Section 5. Graphs with small values for different types of tightness are classified in Section 6. Graphs with smallest tightness values (among all graphs of the same order not exceeding 10) are identified within Section 7.

2. Load balancing

The job which has to be executed by a multiprocessor system is divided into parts that are given to particular processors to handle them. We can assume that the whole job consists of a number of elementary jobs (items) so that each processor gets a number of such elementary jobs to execute. Mathematically, elementary jobs distribution among processors can be represented by a vector x whose coordinates are non-negative integers. Coordinates are associated to graph vertices and indicate how many elementary jobs are given to corresponding processors.

Vector x is usually changed during the work of the system because some elementary jobs are executed while new elementary jobs are permanently generated during the execution process. Of course, it would be optimal that the number of elementary jobs given to a processor is the same for all processors, i.e., that the vector x is an integer multiple of the vector j whose all coordinates are equal to 1. Since this is not always possible, it is reasonable that processors with a great number of elementary jobs send some of them to adjacent processors so that the job distribution becomes uniform if possible. In this way the so called problem of *load balancing* is important in managing multiprocessor systems. The load balancing problem requires creation of algorithms for moving elementary jobs among processors in order to achieve the uniform distribution.

We shall present an algorithm for the load balancing problem which is based on the Laplacian matrix of a graph. A similar algorithm can be constructed using the adjacency matrix.

Let G be a connected graph on n vertices. Eigenvalues and corresponding ortonormal eigenvectors of the Laplacian $L = D - A$ of G are denoted by $\nu_1, \nu_2, \ldots, \nu_n = 0$ and $u_1, u_2, \ldots, u_n$, respectively. Any vector x from R^n can be represented as a linear combination of the form $x = \alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_n u_n$.

Suppose now that G has distinct Laplacian eigenvalues $\mu_1, \mu_2, \dots, \mu_m = 0$ with multiplicities $k_1, k_2, \dots, k_m = 1$, respectively. Vector x can now be represented in the form $x = y_1 + y_2 + \cdots + y_m$ where y_i belong to the eigenspace of μ_i for $i = 1, 2, \dots, m$. We also have $y_m = \beta j$ for some β.

Since $Lx = L(y_1 + y_2 + \cdots + y_m) = \mu_1 y_1 + \mu_2 y_2 + \cdots + \mu_m y_m$, we have $x^{(1)} = x - \frac{1}{\mu_1}Lx = \big(I - \frac{1}{\mu_1}L\big)x = \big(1 - \frac{\mu_2}{\mu_1}\big)y_2 + \cdots + \beta j$. We see that the component of x in the eigenspace of μ_1 has been cancelled by the transformation by the matrix $I - \frac{1}{\mu_1}L$ while the component in the eigencpace of μ_m remains unchanged. The transformation $I - \frac{1}{\mu_2}L$ will cause that the component of $x^{(2)} = \big(I - \frac{1}{\mu_2}L\big)x^{(1)}$ in the eigenspace of μ_2 disappears. Continuing in this way

$$x^{(k)} = \Big(I - \frac{1}{\mu_k}L\Big)x^{(k-1)}, \quad k = 1, 2, \dots, m-1 \tag{1}$$

we shall obtain $x^{(m-1)} = \beta j$.

We have seen how a vector x can be transformed to a multiple of j using the iteration process (1) which involves the Laplacian matrix of the multiprocessor graph G. It remains to see what relations (1) mean in terms of load moving.

Let vector $x^{(k)}$ have coordinates $x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}$. Relations (1) can be rewritten in the form

$$x_i^{(k)} = x_i^{(k-1)} - \frac{1}{\mu_k}\sum_{i*j}\Big(d_i x_i^{(k-1)} - x_j^{(k-1)}\Big) \tag{2}$$

where d_i is the degree of vertex i. This means that the current load at vertex i is changed in such a way that vertex (processor) i sends $\frac{1}{\mu_k}$-th part of its load to each of its d_i neighbors and, because this holds for every vertex, also receives $\frac{1}{\mu_k}$-th part of the load from each of its d_i neighbors.

In this way we have defined a load flow on the edge set of G. First, particular amounts of load flow should be considered algebraically, i.e., having in mind their sign. So, if $x_i^{(k-1)}$ is negative, then vertex i, in fact, receives the corresponding amount. For each edge ij we have two parts of the flow: the part which is sent (or received) by i and the part which is sent (or received) by j. These two amounts should be added algebraically and in this way we get final value of the flow through edge ij. This flow at the end has a non-negative value which is sent either from i to j or vice versa.

Although the load flow plan defined in this way by relations (1) theoretically solves the problem of load balancing, one should be careful when it has to be really applied. This is not the only flow plan which solves the problem. For example, one can apply relations (1) with various orders of eigevalues. Further, the flow plan that we get could be such that the load is sent to final destinations via long paths. Also, it is not clear that a flow plan is always realizable because it could happen that a vertex has not enough elementary jobs to send which it should send according to the flow plan. These facts indicate that one should further consider the load balancing and find, if possible, flow plans which would be optimal according to some criteria.

We shall not further elaborate the problem of load balancing and the interested reader can consult the literature (see, for example, [19] and references given there).

Here we point out the obvious fact that the number of iterations in (1) is equal to the number of non-zero distinct Laplacian eigenvalues of the underlying graph. Hence we see that from the point of view of complexity of the load balancing algorithms graphs with a small number of distinct Laplacian eigenvalues are suitable for modelling multiprocessor interconnection networks. In addition, maximum vertex degree Δ of G also affects computation of the balancing flow. Therefore, the complexity of the balancing flow calculations essentially depends on the product $m\Delta$ and that is why this quantity was proposed in [19] as a parameter relevant for the choice and the design of multiprocessor interconnection networks.

Although graphs with few distinct eigenvalues allow a quick execution of load balancing algorithms, it is not expected that infinite families of such graphs with small tightness can be constructed.

A graph is called *integral* if its spectrum consists entirely of integers. Each eigenvalue has integral eigenvectors and each eigenspace has a basis consisting of such eigenvectors.

In integral graphs load balancing algorithms, which use eigenvalues and eigenvectors, can be executed in integer arithmetics. The further study of integral graphs in connection to multiprocessor topologies seems to be a promising subject for future research (see [5, 9]).

See references [17, 18, 23, 26, 27] for a further study of the load balancing problem.

3. Various types of tightness of a graph

As we have already pointed out, the graph invariant obtained as the product of the number of distinct eigenvalues m and the maximum vertex degree Δ of G has been investigated in [19] related to the design of multiprocessor topologies. The main conclusion of [19] with respect to the multiprocessor design and, in particular to the load balancing within given multiprocessor systems was the following: if $m\Delta$ is small for a given graph G, the corresponding multiprocessor topology was expected to have good communication properties and has been called *well-suited*. It has been pointed out that there exists an efficient algorithm which provides optimal load balancing within $m - 1$ computational steps. The graphs with large $m\Delta$ were called *ill-suited* and were not considered suitable for design of multiprocessor networks.

Several families of graphs with a small product $m\Delta$ have been constructed. One such family is based on hypercubes. It is interesting that the ubiquitous Petersen graph appears also as a good candidate for multiprocessor interconnection networks.

On the other hand there are many known and widely used multiprocessor topologies based on graphs which appear to be ill-suited according to [19]. Such an example is the star graph $S_n = K_{1,n-1}$.

In order to extend and improve the application of the theory of graph spectra to the design of multiprocessor topologies, some other types of graph invariants (under common name tightness) have been defined in [6] and their suitability for describing the corresponding interconnection networks investigated.

As we can see, $m\Delta$ is the product of one spectral invariant m and one structural invariant Δ. Therefore, we will refer to this type of tightness as the *mixed tightness*. In [6], we introduced two alternative (homogeneous) definitions of tightness, the *structural* and the *spectral* one. Moreover, we introduced another mixed tightness, and therefore we end up with *type one mixed tightness* and *type two mixed tightness*. Here we recall all these definitions. New types of tightness involve another structural invariant (diameter) and another spectral invariant (the largest eigenvalue). Both invariants are important for communication properties of a network in general.

Definition 1. The *tightness* $t_1(G)$ of a graph G is defined as the product of the number of distinct eigenvalues m and the maximum vertex degree Δ of G, i.e., $t_1(G) = m\Delta$.

Definition 2. *Structural tightness* $\mathrm{stt}(G)$ is the product $(D+1)\Delta$ where D is diameter and Δ is the maximum vertex degree of a graph G.

Definition 3. *Spectral tightness* $\mathrm{spt}(G)$ is the product of the number of distinct eigenvalues m and the largest eigenvalue λ_1 of a graph G.

Definition 4. *Second type mixed tightness* $t_2(G)$ is defined as a function of the diameter D of G and the largest eigenvalue λ_1, i.e., $t_2(G) = (D+1)\lambda_1$.

If the type of tightness is not relevant for the discussion, all four types of tightness will be called, for short, tightness. In general discussions we shall use $t_1, t_2, \mathrm{stt}, \mathrm{spt}$ independently of a graph to denote the corresponding tightness. An alternative term for tightness could be the word *reach*.

The use of the largest eigenvalue, i.e. the index, of a graph instead of the maximum vertex degree in description of multiprocessor topologies seems to be appropriate for several reasons. By Theorem 1.12 of [14] the index of a graph is equal to a kind of mean value of vertex degrees, i.e. to the so called dynamical mean value, which takes into account not only immediate neighbors of vertices, but also neighbors of neighbors, etc. The index is also known to be a measure of the extent of branching of a graph, and in particular of a tree (see [11] for the application in chemical context and [10] for a treatment of directing branch and bound algorithms for the travelling salesman problem). The index, known also as a spectral radius, is a mathematically very important graph parameter as presented, for example, in a survey paper [12].

According to the well-known inequality $d_{\min} \leqslant \bar{d} \leqslant \lambda_1 \leqslant d_{\max} = \Delta$, [14, p. 85] we have that $\mathrm{spt}(G) \leqslant t_1(G)$. Here $d_{\min}$ and $d_{\max}$ denote minimum and maximum vertex degrees, respectively and $\bar{d}$ is used to denote the average value of vertex degrees.

The relation between $\mathrm{stt}(G)$ and $t_1(G)$ is $t_1(G) \geqslant \mathrm{stt}(G)$, since $m \geqslant 1 + D$ (see Theorem 3.13. from [14]). For distance-regular graphs [3] $m = 1 + D$ holds.

We also have $t_2(G) \leqslant \mathrm{spt}(G)$ and $t_2(G) \leqslant \mathrm{stt}(G)$.

The two homogeneous tightness appear to be incomparable. To illustrate this, let us consider star graph with $n = 5$ vertices ($S_5 = K_{1,4}$) given on Fig. 1a, and the graph $\bar{S}_5$ obtained if new edges are added to the star graph as it is shown on Fig. 1b.

a) b)

FIGURE 1. a) Star graph with $n = 5$ vertices and b) extended star graph

From [14, pp. 272–275, Table 1], we can see that for S_5 it holds $D = 2$, $\Delta = 4$, $m = 3$ and $\lambda_1 = 2$ and hence $\mathrm{spt}(S_5) = m\lambda_1 = 6 < 12 = (D+1)\Delta = \mathrm{stt}(S_5)$. On the other hand for the graph $\bar{S}_5$ we have $D = 2$, $\Delta = 4$, $m = 4$ and $\lambda_1 = 3.2361$ yielding to $\mathrm{spt}(\bar{S}_5) > \mathrm{stt}(\bar{S}_5)$.

The above mentioned table shows that this is not the only example. For $n = 5$, 21 different graphs exist. Only for 3 of them the two homogeneous tightness have the same value, while $\mathrm{stt}(G)$ is smaller for 9 graphs, and for the remaining 9 graphs $\mathrm{spt}(G)$ has a smaller value.

For two graph invariants $\alpha(G)$ and $\beta(G)$ we shall say that the relation $\alpha(G) \prec \beta(G)$ holds if $\alpha(G) \leqslant \beta(G)$ holds for any graph G. With this definition we have the Hasse diagram for the $\prec$ relation between various types of tightness given on Fig. 2.

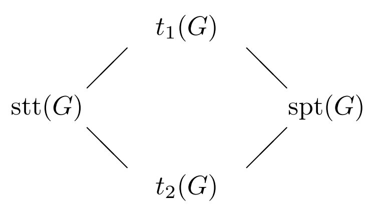

FIGURE 2. Partial order relation between different types of graph tightness

In order to study the behavior of a property or invariant of graphs when the number of vertices varies, it is important that the property (invariant) is scalable. *Scalability* means that for each n there exists a graph with n vertices having that property (invariant of certain value).

A family of graphs is called *scalable* if for any n there exists an n-vertex graph in this family. For example, in [19] the scalable families of sparse graphs (maximum vertex degree $O(\log n)$) with small number of distinct eigenvalues are considered. Obviously, sometimes it is difficult to construct scalable families of graphs for a given property.

We present a theorem which seems to be of fundamental importance in the study of the tightness of a graph.

Theorem 1. *For any kind of tightness, the number of connected graphs with a bounded tightness is finite.*

Proof. Let $t(G) \leqslant a$ for a given positive integer a, where $t(G)$ stands for any kind of tightness. In all four cases, we shall prove that there exists a number b such that both diameter D and maximum vertex degree Δ are bounded by b. We need two auxiliary results from the theory of graph spectra.

Having in view (1) and (2) from the introductory chapter of this publication, $t(G) \leqslant a$ implies

Case $t(G) = t_1(G)$. $m\Delta \leqslant a$, $m \leqslant a$ and $\Delta \leqslant a$, $D \leqslant a-1$, and we can adopt $b = a$;

Case $t(G) = \text{stt}(G)$. $(D+1)\Delta \leqslant a$, $D \leqslant a-1$ and $\Delta \leqslant a$, here again $b = a$;

Case $t(G) = \text{spt}(G)$. $m\lambda_1 \leqslant a$, $m \leqslant a$ and $\lambda_1 \leqslant a$, $D \leqslant a-1$, and $\Delta \leqslant \lambda_1^2 \leqslant a^2$, and now $b = a^2$;

Case $t(G) = t_2(G)$. $(D+1)\lambda_1 \leqslant a$, $D \leqslant a-1$, and $\Delta \leqslant a^2$, and again $b = a^2$.

It is well known that for the number of vertices n in G the following inequality holds

$$n \leqslant 1 + \Delta + \Delta(\Delta-1) + \Delta(\Delta-1)^2 + \cdots + \Delta(\Delta-1)^{D-1}. \tag{3}$$

To derive this inequality vertices of G are enumerated starting from a particular vertex and adding maximum number of neighbors at particular distances from that vertex. Based on this relation and assuming that both D and Δ are bounded by a number b, we have

$$\begin{aligned} n &< 1 + \Delta + \Delta^2 + \Delta^3 + \cdots + \Delta^D \leqslant 1 + \Delta + \Delta^2 + \Delta^3 + \cdots + \Delta^b \\ &\leqslant 1 + b + b^2 + b^3 + \cdots + b^b. \end{aligned}$$

In such a way we proved that the number of vertices of a connected graph with a bounded tightness is bounded. Therefore, it is obvious that there can be only finitely many such graphs and the theorem is proved. $\square$

Corollary 1. *The tightness of graphs in any scalable family of graphs is unbounded.*

Corollary 2. *Any scalable family of graphs contains a sequence of graphs, not necessarily scalable, with increasing tightness diverging to $+\infty$.*

The asymptotic behavior of the tightness, when n tends towards $+\infty$, is of particular interest in the analysis of multiprocessor interconnection networks. Typically, in suitable (scalable) families of graphs the tightness values have asymptotic behavior, for example, $O(\log n)$ or $O(\sqrt{n})$. Several cases are studied in [6] and reviewed also here in the next section.

4. A survey of frequently used interconnection networks

In this section we survey the graphs that are often used to model multiprocessor interconnection networks and examine the corresponding tightness values. Since the tightness is a product of two positive quantities, it is necessary for both of them to have small values to assure a small value of tightness.

1. An example of such a graph is the *d-dimensional hypercube* $Q(d)$. It consists of $n = 2^d$ vertices, each of them connected with d neighbors. The vertices are labeled starting from 0 to $n-1$ (considered as binary numbers). An edge connects two vertices with binary number differing in only one bit. For these graphs we have $m = d+1,\ D = d,\ \Delta = d,\ \lambda_1 = d$ and all four types of the tightness are equal to $(d+1)d = O((\log n)^2)$.

Since the connection is fully symmetric, for the diameter we have $D(Q(d)) = d$. The 1-, 2- and 3-dimensional hypercubes are illustrated on Fig. 3. □

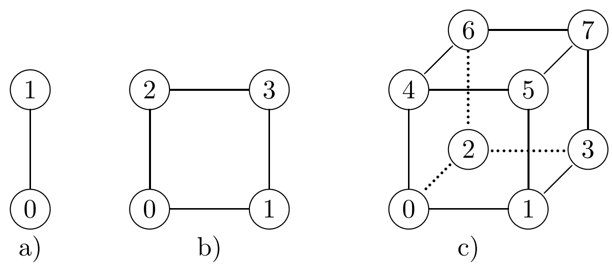

FIGURE 3. The examples of hypercube multiprocessor topologies

2. Another example is *butterfly* graph $B(k)$ containing $n = 2^k(k+1)$ vertices (Fig. 4). The vertices of this graph are organized in $k+1$ levels (columns) each containing 2^k vertices. In each column, vertices are labelled in the same way (from 0 to $2^k - 1$). An edge is connecting two vertices if and only if they are in the consecutive columns i and $i+1$ and their numbers are the same or they differ only in the bit at the i-th position. The maximum vertex degree is $\Delta = 4$ (the vertices from the two outer columns have degree 2 and the vertices in $k-1$ inner columns all have degree 4). Diameter D equals $2k$ while the spectrum is given in [19, Theorem 11]. Therefrom, the largest eigenvalue is $\lambda_1 = 4\cos(\pi/(k+1))$. However, it is not obvious how to determine parameter m. Therefore, we got only the values $\mathrm{stt} = 4(2k+1) = O(\log n)$ and $t_2 = 4(2k+1)\cos(\pi/(k+1)) = O(k) = O(\log n)$. □

Widely used interconnection topologies include some kind of trees, meshes and toruses [28]. We shall describe these structures in some details.

3. *Stars* $S_n = K_{1,n-1}$ are considered as ill-suited topologies in [19], since the tightness $t_1(S_n)$ is large. However stars are widely used in the multiprocessor system design, the so-called master–slave concept is based on the star graph structure. This fact may be an indication that the classification of multiprocessor interconnection networks based on the value for t_1 is not always adequate.

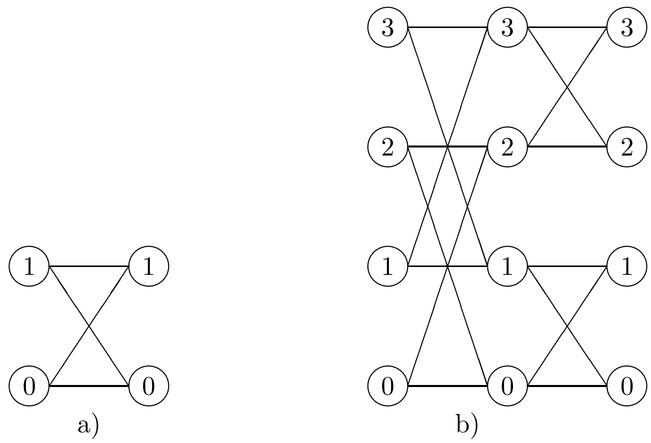

FIGURE 4. The examples of butterfly multiprocessor topologies

For S_n: $m = 3$, $\Delta = n - 1$, $D = 2$, $\lambda_1 = \sqrt{n-1}$ and we have

$$t_1(S_n) = 3(n-1),$$
$$\mathrm{stt}(S_n) = 3(n-1), \quad \mathrm{spt}(S_n) = 3\sqrt{n-1},$$
$$t_2(S_n) = 3\sqrt{n-1}.$$

Stars are only the special case in more general class of bipartite graphs. The main representative of this class are complete bipartite graphs K_{n_1,n_2} having vertices divided into two sets and edges connecting each vertex from one set to all vertices in the other set. For K_{n_1,n_2} we have $m = 3$, $\Delta = \max\{n_1, n_2\}$, $D = 2$, $\lambda_1 = \sqrt{n_1 n_2}$ and hence

$$t_1(K_{n_1,n_2}) = \mathrm{stt}(K_{n_1,n_2}) = 3\max\{n_1, n_2\},$$
$$\mathrm{spt}(K_{n_1,n_2}) = t_2(K_{n_1,n_2}) = 3\sqrt{n_1 n_2}.$$

In the case $n_1 = n_2 = n/2$ all tightness values are of order $O(n)$. However, for the star S_n we have $t_2(S_n) = O(\sqrt{n})$. This may be the indication that complete bipartite graphs are suitable for modelling multiprocessor interconnection networks only in some special cases. □

4. *Mesh* (or *grid*) (Fig. 5a) consists of $n = n_1 n_2$ vertices organized within layers. We can enumerate vertices with two indices, like the elements of an $n_1 \times n_2$ matrix. Each vertex is connected to its neighbors (the ones whose one of the indices is differing from its own by one). The inner vertices have 4 neighbors, the corner ones only 2, while the outer (but not corner ones) are of degree 3. Therefore, $\Delta = 4$, $D = n_1 + n_2 - 2$. Spectrum is given in [14, p. 74]. In particular, the largest eigenvalue is $\lambda_1 = 2\cos(\pi/(n_1+1)) + 2\cos(\pi/(n_2+1))$ and for the tightness of the second type we obtain $t_2 = (n_1 + n_2 - 1)(2\cos(\pi/(n_1+1)) + 2\cos(\pi/(n_2+1)))$. Hence, $t_2 = O(\sqrt{n})$ if $n_1 \approx n_2$. □

a) b)

FIGURE 5. a) Mesh of order 3×4 and b) corresponding torus architecture

5. *Torus* (Fig. 5b) is obtained if the mesh architecture is closed among both dimensions. We do not distinguish corner or outer vertices any more. The characteristics of a torus are $\Delta = 4$, $D = [n_1/2] + [n_2/2]$. Spectrum is given in [14, p. 75]. In particular, the largest eigenvalue is $\lambda_1 = 2\cos(2\pi/n_1) + 2\cos(2\pi/n_2)$ and thus $t_2 = ([n_1/2] + [n_2/2] + 1)(2\cos(2\pi/n_1) + 2\cos(2\pi/n_2))$. As in the previous case (for mesh) we have $t_2 = O(\sqrt{n})$ if $n_1 \approx n_2$. □

All these architectures satisfy both requirements of designing the multiprocessor topologies (small distance between processors and small number of wires). Those of them which have a small value for t_1 are called *well-suited interconnection topologies* in [19]. Other topologies are called *ill-suited*. Therefore, according to [19], well-suited and ill-suited topologies are distinguished by the value for the mixed tightness of the first type $t_1(G)$.

The star example suggests that $t_2(G)$ is a more appropriate parameter for selecting well-suited interconnection topologies than $t_1(G)$. Namely, the classification based on the tightness t_2 seems to be more adequate since it includes stars in the category of well-suited structures.

The obvious conclusion following from the Hasse diagram given on Fig. 2, is that the well-suited interconnection network according to the value for t_1 remain well-suited also when t_2 is taken into consideration. In this way, some new graphs become suitable for modelling multiprocessor interconnection networks. Some of these "new" types of graphs are already recognized by multiprocessor system designers (like stars and bipartite graphs). In the next section we propose a new family of t_2-based well-suited trees.

5. Complete quasi-regular trees

In this section we shall study properties of some trees and show that they are suitable for our purposes.

The complete quasi-regular tree $T(d, k)(d = 2, 3, \dots, k = 1, 2, \dots)$ is a tree consisting of a central vertex and k layers of other vertices, adjacencies of vertices being defined in the following way.

1. The central vertex (the one on the layer 0) is adjacent to d vertices in the first layer.

2. For any $i = 1, 2, \ldots, k-1$ each vertex in the i-th layer is adjacent to $d-1$ vertices in the $(i+1)$-th layer (and one in the $(i-1)$-th layer).

The graph $T(3,3)$ is given in Fig. 6.

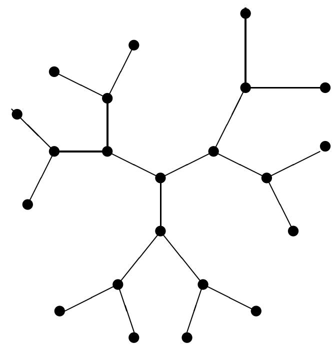

FIGURE 6. Quasi-regular tree $T(3,3)$

The graph $T(d,k)$ for $d > 2$, has $n = 1+d((d-1)^k-1)/(d-2)$ vertices, maximum vertex degree $\Delta = d$, diameter $D = 2k$ and the largest eigenvalue $\lambda_1 < d$. (The spectrum of $T(d,k)$ has been determined in [25]). We have $k = O(\log n)$ and, since $t_2(T(d,k)) = (D+1)\lambda_1 < (D+1)\Delta = \mathrm{stt}(T(d,k)) = (2k+1)d$, we obtain $t_2(T(d,k)) = O(\log n)$. This is asymptotically better than in the hypercube $Q(d)$ case, where $t_2(Q(d)) = O((\log n)^2)$ or in the case for star graph where $t_2(K_{1,n-1}) = O(\sqrt{n})$ (see Section 4). Note that the path P_n with $t_2(P_n) = 2n\cos(\pi/(n+1)) = O(n)$ also performs worse.

The coefficient of the main term in the expression for $t_2(T(d,k))$ is equal to $d/\log(d-1)$ with values of 4.328, 3.641, 3.607, 3.728, 3.907, 4.111, 4.328 and 4.551 for $d = 3, 4, 5, 6, 7, 8, 9, 10$, respectively. The coefficient is further an increasing function of d. Therefore the small values of d are desirable and we shall discuss in details only the case $d = 3$ since it is suitable for resolving the stability issues. The other cases with small values for d can be analyzed analogously.

To examine the suitability of graphs $T(3,k)$, we compared its tightness values with the corresponding ones for two interesting classes of trees: paths P_n and stars $S_n = K_{1,n-1}$ containing the same number of vertices $n = 3\cdot 2^k - 2$. The results for small values of k are summarized in the Table 1. 5.

Since for paths and quasi-regular trees the mixed tightness of the second type has almost the same value as the mixed tightness of the first type, we put only the values for the first type mixed tightness for paths, while for $T(n,k)$ the structural tightness is given.

The last column (for stars) contains the values for two tightness, first for the mixed tightness of the first type and then the value for the mixed tightness of the second type in the parentheses.

As can be seen from the Table 1, the tightness values for paths P_n are significantly larger than the values $\mathrm{stt}(T(3,k))$. Star architecture seems to be better for small values of k, but starting from $k = 6$, we have $t_2(T(3,k)) < \mathrm{stt}(T(3,k)) < t_2(S_n)$.

TABLE 1. Tightness values for some trees

k	n	P_n $t_1(\geqslant t_2)$	$T(3,k)$ stt$(\geqslant t_2)$	S_n t_1	(t_2)
1	4	$4\cdot 2$	$3\cdot 3$	$3\cdot 3$	$(3\cdot\sqrt{3})$
2	10	$10\cdot 2$	$5\cdot 3$	$3\cdot 9$	$(3\cdot\sqrt{9}=3\cdot 3)$
3	22	$22\cdot 2$	$7\cdot 3$	$3\cdot 21$	$(3\cdot\sqrt{21}<3\cdot 5)$
4	46	$46\cdot 2$	$9\cdot 3$	$3\cdot 45$	$(3\cdot\sqrt{45}<3\cdot 7)$
5	94	$94\cdot 2$	$11\cdot 3$	$3\cdot 93$	$(3\cdot\sqrt{93}<3\cdot 10)$
6	190	$190\cdot 2$	$13\cdot 3$	$3\cdot 189$	$(3\cdot\sqrt{189}>3\cdot 13)$
7	382	$382\cdot 2$	$15\cdot 3$	$3\cdot 381$	$(3\cdot\sqrt{381}>3\cdot 19)$

The intention when comparing complete quasi-regular trees $T(3,k)$ with paths P_n and stars S_n is to examine their place between two kinds of trees, extremal for many graph invariants. In particular, among all trees with a given number of vertices, the largest eigenvalue λ_1 and maximum vertex degree Δ have minimal values for the path and maximal for the star, while, just opposite, the number of distinct eigenvalues m and the diameter D have maximal values for the path and minimal for the star. Since the tightness (of any type) is a product of two graph invariants having, in the above sense, opposite behavior it is expected that its extreme value is attained "somewhere in the middle". Therefore, for a tree of special structure (like the quasi-regular trees are) we expect both tendencies to be in an equilibrium.

It is not difficult to extend the family of complete quasi-regular trees to a scalable family. A *quasi-regular tree* is a tree obtained from a complete quasi-regular tree by deleting some of its vertices of degree 1. If none or all vertices of degree 1 are deleted from a complete quasi-regular tree we obtain again a complete quasi-regular tree. Hence, a complete quasi-regular tree is also a quasi-regular tree. While a complete quasi-regular tree is unique for the given number of vertices, there are several non-isomorphic quasi-regular trees with the same number of vertices which are not complete. Therefore, there are several ways to construct a scalable family of quasi-regular trees. The following way is a very natural one.

Consider a complete quasi-regular tree $T(d,k)$ and perform the breadth first search through the vertex set starting from the central vertex. Adding to $T(d,k-1)$ pendant vertices of $T(d,k)$ in the order they are traversed in the mentioned breadth first search defines the desired family of quasi-regular trees.

The constructed family has the property that each its member has the largest eigenvalue λ_1 among all quasi-regular trees with the same number of vertices [34]. At first glance this property is something what we do not want since we are looking for graphs with the tightness t_2 as small as possible. Instead we would prefer, unlike the breadth first search, to keep adding pendant vertices to $T(d,k-1)$ in such a balanced way around that we always get a quasi-regular tree with largest eigenvalue as small as possible. Such a way of adding vertices is not known and its finding represents a difficult open problem in the spectral graph theory.

A scalable family of trees with $O((\log n)^2)$ distinct eigenvalues has been studied in [19]. An open question remains to compare the performances of these two families.

In our context interesting are also *fullerene graphs* corresponding to carbon compounds called *fullerenes*. Mathematically, fullerene graphs are planar regular graphs of degree 3 having as faces only pentagons and hexagons. It follows from the Euler theorem for planar graphs that the number of pentagons is exactly 12. Although being planar, fullerene graphs are represented (and this really corresponds to actual positions of carbon atoms in a fullerene) in 3-space with its vertices embedded in a quasi-spherical surface.

A typical fullerene C_{60} is given in Fig. 7. It can be described also as a truncated icosahedron and has the shape of a football.

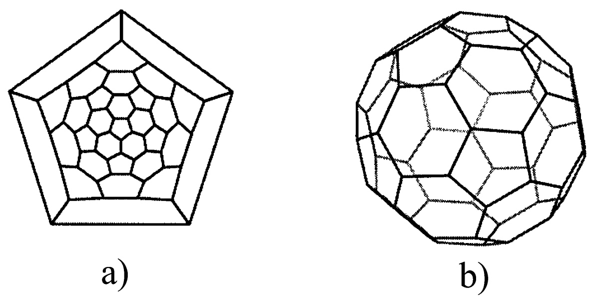

FIGURE 7. a) Planar and b) 3D visualization of the icosahedral fullerene C_{60}

Without elaborating details we indicate the relevance of fullerene graphs to our subject by comparing them with quasi-regular trees.

For a given number of vertices the largest eigenvalues of the two graphs are roughly equal (equal to 3 in fullerenes and close to 3 in quasi-regular trees) while the diameters are also comparable. This means that the tightness t_2 is approximately the same in both cases. In particular, the values of relevant invariants for the fullerene graph C_{60} are $n = 60$, $D = 9$ (see [21]), $m = 15$ (see [22]), $\Delta = \lambda_1 = 3$. Hence, $\text{stt} = t_2 = 30$. A quasi-regular tree on 60 vertices has diameter $D = 9$ and we also get $\text{stt} = 30$.

Note that the tightness t_1 is not very small since it is known that fullerene graphs have a large number of distinct eigenvalues [22].

It is also interesting that fullerene graphs have a nice 3D-representation in which the coordinates of the positions of vertices can be calculated from the eigenvectors of the adjacency matrix (the so called *topological coordinates* which were also used in producing the atlas [22]).

6. Graphs with small tightness values

In this section we classify graphs with small tightness values. In particular, we find graphs with tightness values not exceeding $a = 9$.

As explained in Introduction, it turned out that the value $a = 9$ is very suitable: we established that exactly 69 graphs obey the bound with the number of vertices in these graphs not exceeding 10 (see [5, 9]). The obtained graphs should be considered as reasonably good models for multiprocessor interconnection networks. A more modest task, finding graphs with tightness values not exceeding 8 is solved in [8].

We are interested in the 69 graphs given in Figs. 8–13 under names $\Omega_{n,k}$, where n ($2 \leqslant n \leqslant 10$) denotes the number of vertices and $k \geqslant 1$ (being a counter).

In Appendix, we give in Table 3 some data on these 69 graphs.

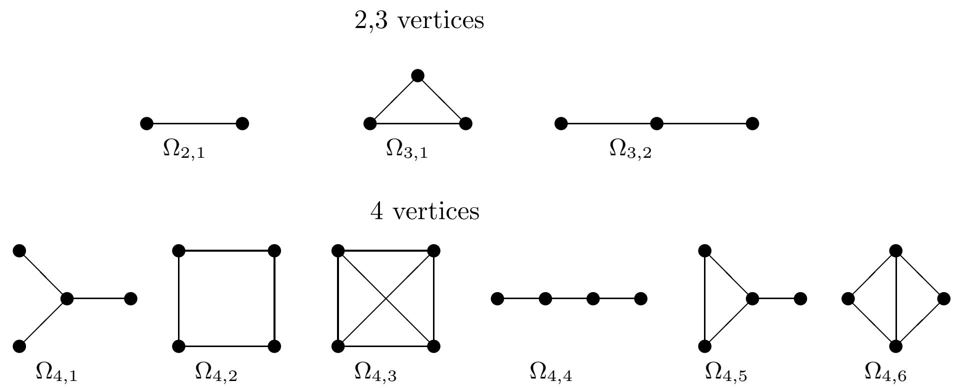

FIGURE 8. Graphs up to 4 vertices with small tightness

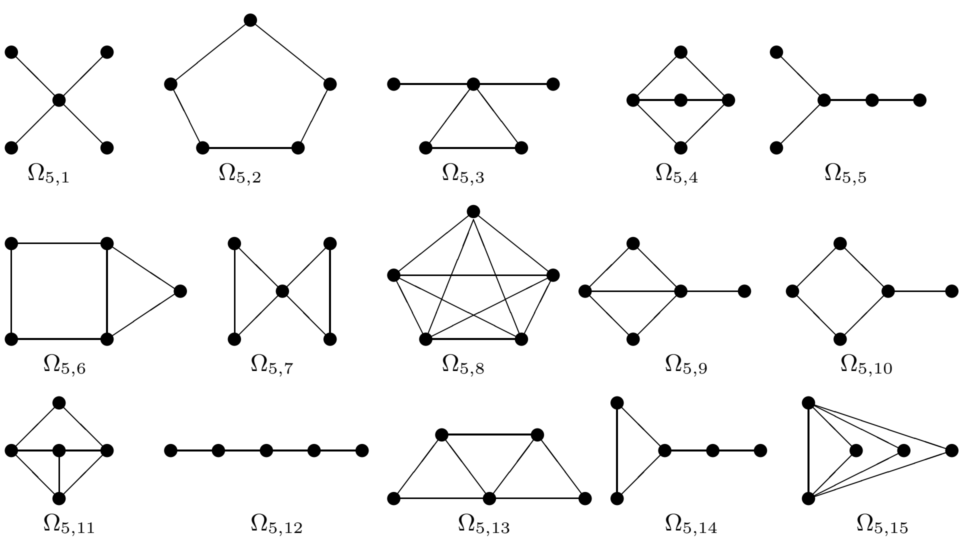

FIGURE 9. Graphs on 5 vertices with small tightness

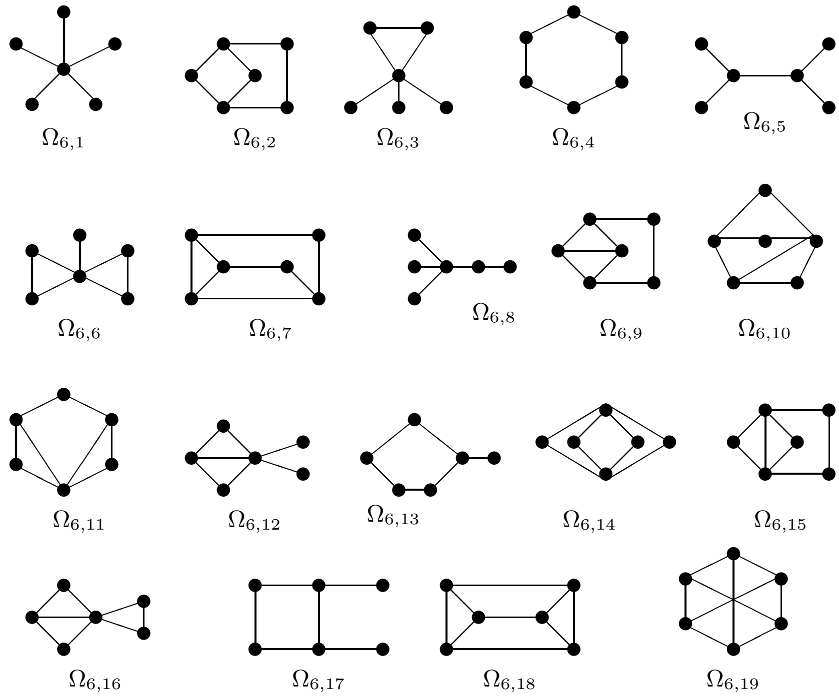

FIGURE 10. Graphs on 6 vertices with small tightness

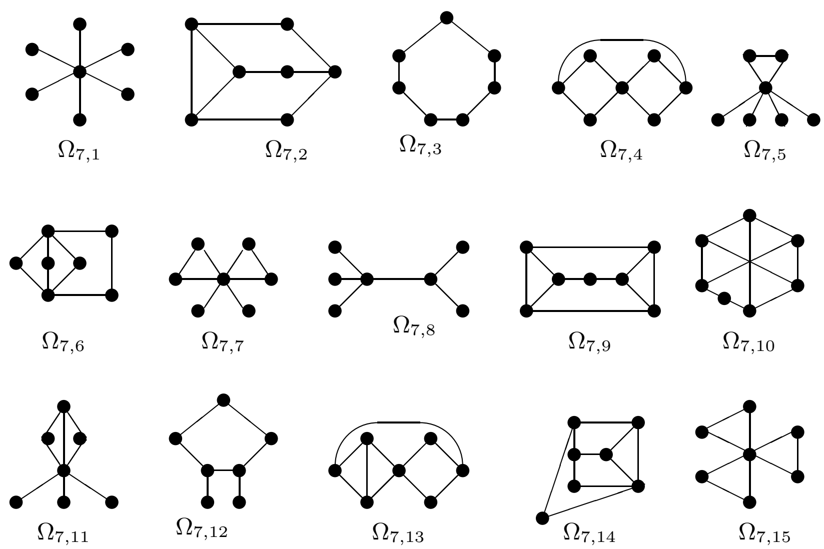

FIGURE 11. Graphs on 7 vertices with small tightness

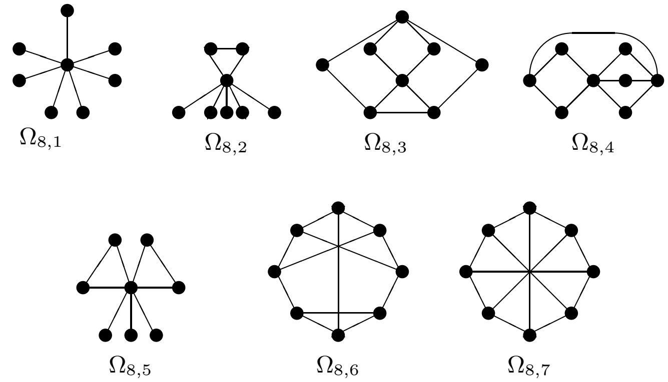

FIGURE 12. Graphs on 8 vertices with small tightness

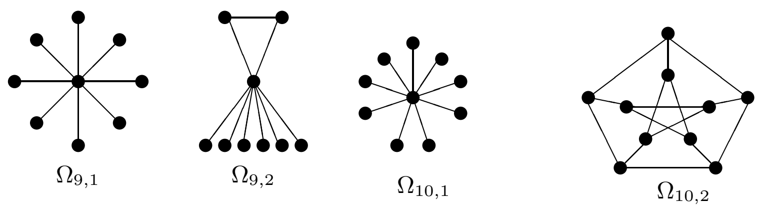

FIGURE 13. Graphs on 9 and 10 vertices with small tightness

The main result of [5] is the next theorem. In [5] only a sketch of a proof is given. The proof is completed in [9].

Theorem 2. *The only non-trivial connected graphs G such that $t_2(G) \leqslant 9$ are the* 69 *graphs $\Omega_{n,k}$, depicted on Figs.* 8–13.

Proof of Theorem 2. We have the following cases:

- a°**:** $D = 1$, $\lambda_1 \leqslant 4.5$. We have complete graphs $\Omega_{2,1}$, $\Omega_{3,1}$, $\Omega_{4,3}$, $\Omega_{5,8}$.
- b°**:** $D = 2$, $\lambda_1 \leqslant 3$. Denote the set of graphs satisfying these conditions by $\mathcal{A}_1$. According to (2) from the introductory chapter we have $\Delta \leqslant \lambda_1^2 \leqslant 9$ and by formula (3) we get $n \leqslant 1 + 9 + 9 \cdot 8 = 82$. For example, the star $\Omega_{10,1} \in \mathcal{A}_1$. The set $\mathcal{A}_1$ is completely determined in Lemma 2.
- c°**:** $D = 3$, $\lambda_1 \leqslant 2.25$. Denote the set of graphs satisfying these conditions by $\mathcal{A}_2$. Now, $\Delta \leqslant 5$ since $\lambda_1^2 < 6$, and we have $n \leqslant 1 + 5 + 5 \cdot 4 + 5 \cdot 4^2 = 106$. Graphs belonging to the set $\mathcal{A}_2$ are listed in Lemma 3.
- d°**:** $D = 4$, $\lambda_1 \leqslant 1.8$. It is easy to see that the only graph in this case is the path $\Omega_{5,12}$ (see information on Smith graphs in Section 2 of the introductory chapter).
- e°**:** $D \geqslant 5$, $\lambda_1 \leqslant 1.5$. There are no graphs satisfying these conditions.

To treat the cases b° and c° in Lemmas 2 and 3 we need an auxiliary result.
Let R be the set of graphs satisfying the conditions $D = 2$, $\Delta = 3$.

Lemma 1. *The set R consists of the following* 17 *graphs:* $\Omega_{4,1}$, $\Omega_{4,5}$, $\Omega_{4,6}$, $\Omega_{5,4}$, $\Omega_{5,6}$, $\Omega_{5,11}$, $\Omega_{6,2}$, $\Omega_{6,7}$, $\Omega_{6,9}$, $\Omega_{6,18}$, $\Omega_{6,19}$, $\Omega_{7,2}$, $\Omega_{7,9}$, $\Omega_{7,10}$, $\Omega_{8,6}$, $\Omega_{8,7}$ *and* $\Omega_{10,2}$.

Proof. By formula (3) graphs from R have at most 10 vertices. Consider a graph $G \in R$. It has a vertex v of degree 3. Let f be the number of edges in the subgraph of G induced by the three neighbours of v. We have the following possibilities:

If $f = 3$, we have $G = \Omega_{4,3}$ which is excluded since $D = 1$.

Consider $f = 2$. Now we start from vertex v and its neighbours and add new vertices and edges in such a way that conditions $D = 2$, $\Delta = 3$ are not violated. We readily get $G = \Omega_{4,6}$, or $G = \Omega_{5,11}$ given on Fig. 9, or G is isomorphic to $\Omega_{6,9}$ from Fig. 10.

In the case $f = 1$ the obtained graphs up to 7 vertices are presented on Fig. 14. Finally, we get the graph $\Omega_{8,6}$ from Fig. 12 on $n = 8$ vertices.

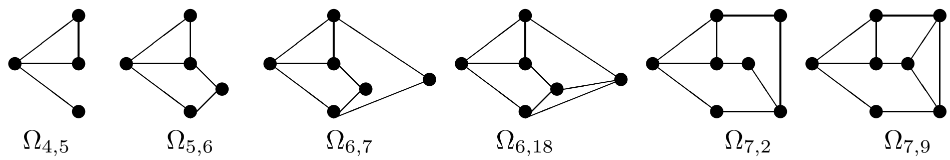

FIGURE 14. Some graphs from the set R

If $f = 0$, we first have complete bipartite graphs $\Omega_{4,1}$, $\Omega_{5,4}$, and $\Omega_{6,19}$, and $\Omega_{6,2}$. For $n = 7$ we again come across graph $\Omega_{7,2}$, and the graph $\Omega_{7,10}$. For $n = 8$ the graphs $\Omega_{8,6}$, $\Omega_{8,7}$ from Fig. 12 appear. The Petersen graph $\Omega_{10,2}$ on 10 vertices belongs here. There are no graphs on 9 vertices. □

Lemma 2. *The set $\mathcal{A}_1$ consists of* 52 *graphs given below.*

$n = 3$: $\Omega_{3,2}$;

$n = 4$: $\Omega_{4,1}$, $\Omega_{4,2}$, $\Omega_{4,6}$, $\Omega_{4,5}$;

$n = 5$: $\Omega_{5,1}$, $\Omega_{5,2}$, $\Omega_{5,3}$, $\Omega_{5,4}$, $\Omega_{5,6}$, $\Omega_{5,7}$, $\Omega_{5,9}$, $\Omega_{5,11}$, $\Omega_{5,13}$, $\Omega_{5,15}$;

$n = 6$: $\Omega_{6,1}$, $\Omega_{6,2}$, $\Omega_{6,3}$, $\Omega_{6,6}$, $\Omega_{6,7}$, $\Omega_{6,9}$, $\Omega_{6,10}$, $\Omega_{6,11}$, $\Omega_{6,12}$, $\Omega_{6,14}$, $\Omega_{6,15}$, $\Omega_{6,16}$, $\Omega_{6,18}$, $\Omega_{6,19}$;

$n = 7$: $\Omega_{7,1}$, $\Omega_{7,2}$, $\Omega_{7,4}$, $\Omega_{7,5}$, $\Omega_{7,6}$, $\Omega_{7,7}$, $\Omega_{7,9}$, $\Omega_{7,10}$, $\Omega_{7,11}$, $\Omega_{7,13}$; $\Omega_{7,14}$, $\Omega_{7,15}$;

$n = 8$: $\Omega_{8,1}$, $\Omega_{8,2}$, $\Omega_{8,3}$, $\Omega_{8,4}$, $\Omega_{8,5}$, $\Omega_{8,6}$, $\Omega_{8,7}$;

$n = 9$: $\Omega_{9,1}$, $\Omega_{9,2}$;

$n = 10$: $\Omega_{10,1}$, $\Omega_{10,2}$ (*the Petersen graph*).

Proof. We shall first prove that there are no graphs on $n > 10$ vertices with diameter 2 and index less than or equal to 3.

Assume to the contrary that G is a graph on $n > 10$ vertices such that $\operatorname{diam}(G) = 2$ and $\lambda_1(G) \leqslant 3$.

We first claim that $\Delta(G) \leqslant 8$. Otherwise, if $\Delta(G) \geqslant 9$ then $\lambda_1(G) > \lambda_1(S_{\Delta+1}) = \sqrt{\Delta} \geqslant 3$, a contradiction. If $\delta(G) = 1$, let v be a pendant vertex G, and w its neighbour. Since the eccentricity of v is at most 2, w must be adjacent to all vertices of G, but then $n \leqslant 10$, a contradiction.

Therefore, we can assume further on that $\delta(G) > 1$ and $\Delta(G) < 9$. Let e be the number of edges of G. Then,

$$3 \geqslant \lambda_1(G) \geqslant \frac{2e}{n} = \bar{d}$$

and the average vertex degree is less than or equal to 3, with equality if and only if G is regular. If G is 3-regular graph with diameter 2, by (4) G can have at most $1 + 3 + 3 \cdot 2 = 10$ vertices, a contradiction.

So the average vertex degree of G is less than 3, and since none of them is of degree 1, nor all are of degree 3, there exists at least one vertex in G, say u, of degree 2. Denote with v and w its neighbours. Let the remaining vertices ($n-3$ in total) be partitioned as follows: A contains the vertices that are adjacent only to v; B contains the vertices that are adjacent only to w; C contains the vertices that are adjacent to both, v and w. If so

$$|A| + |C| \leqslant 7 \qquad \text{and} \qquad |B| + |C| \leqslant 7.$$

Since $|A| + |B| + |C| = n - 3$ and $n > 10$, we have $|A| > 0$ and $|B| > 0$.

Let all edges incident to v or w be coloured in blue, while the other edges, non-incident to v or w (incident only to vertices from $A \cup B \cup C$) be coloured in red. Let e_b and e_r be the number of blue and red edges in G, respectively. Clearly, $e_b \geqslant n - 1 + |C|$.

We now claim that $e_r \geqslant |A| + |B| - 1$. To see this, assume first that $H = \langle A \cup B \cup C \rangle$ (the subgraph induced by the vertex set $A \cup B \cup C$) is connected. Then, $e_r \geqslant |A| + |B| + |C| - 1 \geqslant |A| + |B| - 1$ and we are done. Let x and y be the vertices belonging to different components of H. Since G is of diameter 2, there is a vertex z adjacent to both vertices x and y. Clearly, $z \neq u$ (otherwise, if $z = u$ then $x = v$ and $y = w$, a contradiction). If $z \in A \cup B \cup C$, then x and y are not in different components of H. So $z = v$ or $z = w$. If $z = v$ then $x, y \in A \cup C$; while if $z = w$ then $x, y \in B \cup C$. It follows that all vertices from the sets A and B are in the same component of H (since x and y cannot belong to $A \cup B$), and therefore $e_r \geqslant |A| + |B| - 1$, as required.

Consequently, we have

$$\frac{3n}{2} \geqslant e = e_b + e_r \geqslant (n - 1 + |C|) + (|A| + |B| - 1) = 2n - 5.$$

But this is equivalent to $n \leqslant 10$, a contradiction.

Hence, there are no graphs on $n > 10$ vertices with diameter 2 and index less than or equal to 3.

By an exhaustive search of connected graphs up to ten vertices one can verify that only the 52 graphs, quoted in the statement of the lemma fulfill the requirements. □

Remark 1. (i) The exhaustive search in [5] was performed by the program **nauty**.

We used publicly available library of programs **nauty** [29] to generate all connected graphs with up to 10 vertices. **nauty** is a program for computing automorphism groups of graphs and digraphs. It can also produce a canonical graph labelling. **nauty** is an open source available function library written in a portable subset of **C**, and runs on a considerable number of different systems. We used its functions for generating all connected graphs on a given number of vertices. The implemented algorithm for generation of graphs is very efficient and provides a compact representation which is not readable by ordinary users. **nauty** library also provides several functions for converting this compact representation into "user friendly" form.

(ii) Another possibility to find the 52 graphs from Lemma 2 is to use computer assisted reasoning.

Graphs up to 7 vertices can be found using existing graph tables [15, 16] (up to 6 vertices), [13] (7 vertices).

Using an interactive graph package we follow the effect of adding vertices and edges to the largest eigenvalue λ_1. (We have used the package newGRAPH available at the address `http://www.mi.sanu.ac.rs/newgraph/`.)

If $\Delta = k$, then there exists a subgraph in the form of the star S_{k+1}.

If $\Delta = 9$, the only solution is $\Omega_{10,1} = S_{10}$, in all other cases $\lambda_1 > 3$.

If $\Delta = 8$, only one edge can be added and we get $\Omega_{9,1} = S_9$ and $\Omega_{9,2}$. Adding a vertex yields $\lambda_1 > 3$.

If $\Delta = 7$, at most two edges can be added and we get $\Omega_{8,1} = S_8$, $\Omega_{8,2}$ and $\Omega_{8,5}$.

If $\Delta = 6$, addition of at most three edges is possible and we get $\Omega_{7,1} = S_7$, $\Omega_{7,5}$, $\Omega_{7,7}$, $\Omega_{7,11}$, $\Omega_{7,15}$.

If $\Delta = 5$, again by adding at most three edges we get $\Omega_{6,1} = S_6$, $\Omega_{6,3}$, $\Omega_{6,6}$, $\Omega_{6,12}$, $\Omega_{6,15}$. Now adding vertices in a specific way is possible and we get $\Omega_{8,4}$.

If $\Delta = 4$, we get $\Omega_{8,3}$ and graphs with less than 8 vertices can be found by graph tables.

The case $\Delta = 3$ is covered by Lemma 1, while the cases $\Delta < 3$ are trivial.

Lemma 3. *The set $\mathcal{A}_2$ consists of* 12 *graphs listed below.*

$n = 4: \quad \Omega_{4,4};$ $\qquad$ $n = 5: \quad \Omega_{5,5},\ \Omega_{5,10},\ \Omega_{5,14};$

$n = 6: \quad \Omega_{6,4},\ \Omega_{6,5},\ \Omega_{6,8},\ \Omega_{6,13},\ \Omega_{6,17};$ $\qquad$ $n = 7: \quad \Omega_{7,3},\ \Omega_{7,8},\ \Omega_{7,12}.$

Proof. By Table 3 given in Appendix the above 12 graphs clearly belong to the set $\mathcal{A}_2$. We shall show that no other graphs H belong to $\mathcal{A}_2$.

Maximal degree of H cannot be at least 5 since in this case H would contain S_6 with an additional vertex (since $D = 3$). Such a subgraph would have $\lambda_1 > 2.25$ which is forbidden.

If $\Delta = 4$, H contains a subgraph isomorphic to S_5. We cannot add an edge to S_5, since then we obtain $\Omega_{5,3}$ with $\lambda_1 > 2.25$ (see Table 3). However, S_5 can be extended with new vertices to graphs $\Omega_{6,8}$ and $\Omega_{7,8}$. No other extensions of vertices and edges are feasible.

Next we have to consider the case $\Delta \leqslant 3$. Now formula (3) gives that H can have at most 10 vertices which completes the proof using Lemma 1. $\square$

This completes the proof of Theorem 2.

Let $\mathcal{G}_c$ be the set of connected graphs with at least two vertices. Let us introduce the following notation:

$$T_1^a = \{G : G \in \mathcal{G}_c,\ t_1(G) \leqslant a\}, \quad T_{\text{stt}}^a = \{G : G \in \mathcal{G}_c,\ \text{stt}(G) \leqslant a\},$$
$$T_{\text{spt}}^a = \{G : G \in \mathcal{G}_c,\ \text{spt}(G) \leqslant a\}, \quad T_2^a = \{G : G \in \mathcal{G}_c,\ t_2(G) \leqslant a\}.$$

It is obvious that $T_1^a \subseteq T_{\text{stt}}^a \subseteq T_2^a$ and $T_1^a \subseteq T_{\text{spt}}^a \subseteq T_2^a$ because of the partial order between tightness values given on Fig. 2.

Using Table 3 from Appendix we can immediately verify the following corollaries of Theorem 2.

Corollary 3. *The only non-trivial connected graphs G such that $t_1(G) \leqslant 9$ are* 14 *graphs $\Omega_{i,j}$, where (i,j) is:*

$$(2,1),\ (3,1),\ (3,2),\ (4,j)\ (j \in \{1,\dots,4\}),$$
$$(5,j)\ (j \in \{2,4,8\}),\ (6,4),\ (6,19),\ (7,3),\ (10,2).$$

Corollary 4. *The only non-trivial connected graphs G such that* $\text{stt}(G) \leqslant 9$ *are* 27 *graphs $\Omega_{i,j}$, where (i,j) is:*

$$(2,1),(3,1),(3,2),\ (4,j)\ (j \in \{1,\dots,6\}),\ (5,j)\ (j \in \{2,4,6,8,11\},$$
$$(6,j)\ (j \in \{2,4,7,9,18,19\}),\ (7,j)\ (j \in \{2,3,9,10\}),\ (8,6),(8,7),(10,2).$$

Corollary 5. *The only non-trivial connected graphs G such that* $\text{spt}(G) \leqslant 9$ *are* 21 *graphs $\Omega_{i,j}$, where (i,j) is:*

$$(2,1),\ (3,1),\ (3,2),\ (4,j)\ (j \in \{1,\dots,5\}),\ (5,j)\ (j \in \{1,2,4,8,\}),$$
$$(6,j)\ (j \in \{1,4,14,19\}),\ (7,1),\ (7,3),\ (8,1),\ (10,2).$$

Corollaries 1–3 have been proved in [5] in another way.

Remark 2. In fact in [5] we have proved that $T_2^9 = Q \cup R' \cup S' \cup V'$, where $T_1^9 = Q$, $T_{\text{stt}}^9 = Q \cup R'$, $T_{\text{spt}}^9 = Q \cup S'$ and $|T_2^9| = 69$.

Here we have

$$Q = \{K_2, K_3, K_4, K_5, P_3, P_4, C_4, C_5, C_6, C_7, K_{1,3}, K_{2,3}, K_{3,3}, PG\},$$
$$S' = \{P_5, K_{1,4}, K_{1,5}, K_{1,6}, K_{1,7}, K_{1,8}, K_{1,9}\},$$
$$R' = \{\Omega_{4,5}, \Omega_{4,6}, \Omega_{5,6}, \Omega_{5,11}, \Omega_{6,2}, \Omega_{6,7}, \Omega_{6,9}, \Omega_{6,18}, \Omega_{7,2}, \Omega_{7,9}, \Omega_{7,10}, \Omega_{8,6}, \Omega_{8,7}\}$$

and V' consists of the remaining 35 graphs. Here, PG denotes the Petersen graph. We see that the sets Q and S' (related to tightness t_1 and spt) contain only the standard graphs. When considering stt and t_2, the graphs with non-standard names occur.

7. Graphs with smallest tightness values

One of the goals in this work is to identify graphs with smallest tightness values for all four types of tightness.

Based on Corollary 1 of Theorem 2 we are in a position to find the best configurations w.r.t. t_1 up to 10 vertices.

Theorem 3. *Among connected graphs G on n ($n \leqslant 10$) vertices the value $t_1(G)$ is minimal for the following graphs:*

$$\begin{array}{lll} K_2 \textit{ for } n = 2, & C_5 \textit{ for } n = 5, & C_8 \textit{ for } n = 8, \\ K_3 \textit{ for } n = 3, & C_6 \textit{ for } n = 6, & C_9 \textit{ for } n = 9, \\ K_4 \textit{ for } n = 4, & C_7 \textit{ for } n = 7, & \textit{the Petersen graph for } n = 10. \end{array}$$

Proof. By Theorem 2, all connected graphs G with $t_1(G)$ at most 9 are known. Among them it is easy to identify graphs with minimal tightness for $n \leqslant 7$ and $n = 10$. The cases $n = 8, 9$ remain. Since m and Δ are both integers, the next unexamined value for t_1 is ten. We easily find that for C_8 and C_9, having $m = 5$ and $\Delta = 2$, tightness value $t_1 = 10$. □

In a similar way we can identify extremal graphs for other types of tightness based on the results presented in the previous section. The obtained graphs are summarized in Table 2. Together with extremal graphs, the corresponding tightness values are given in parentheses.

TABLE 2. Minimal graphs with their tightness values

n	t_1	stt	spt	t_2
2	K_2 (2)	K_2 (2)	K_2 (2)	K_2 (2)
3	K_3 (4)	K_3 (4)	K_3 (4)	K_3 (4)
4	K_4, C_4 (6)	K_4, C_4 (6)	S_4 (5.196)	S_4 (5.196)
5	C_5 (6)	C_5 (6)	C_5, S_5 (6)	C_5, S_5 (6)
6	C_6 (8)	C_6 (8)	S_6 (6.708))	S_6 (6.708))
7	C_7 (8)	C_7 (8)	S_7 (7.348)	S_7 (7.348)
8	C_8 (10)	$N(8, 6660), N(8, 8469)$ (9)	S_8 (7.937)	S_8 (7.937)
9	C_9 (10)	C_9 (10)	S_9 (8.485)	S_9 (8.485)
10	PG (9)	PG (9)	PG, S_{10} (9)	PG, S_{10} (9)

Several interesting observations can be made.

For $n = 2$ and $n = 3$ complete graphs (in a trivial way) are minimal graphs for all four types of tightness. Starting from $n = 4$, tightness spt and t_2 start to suggest stars as best interconnection networks while tightness t_1 and stt start to suggest circuits as the best ones. Surprises come for $n = 8$ and $n = 10$.

For $n = 8$ according to the tightness stt we get two cubic graphs $N(8, 6660)$ and $N(8, 8469)$ (graphs in which all vertex degrees are equal to 3) of diameter 2. These graphs break the circuit sequence of minimal graphs for stt. They also represent the only case (among small graphs) when t_1 and stt have different minimal values.

For $n = 10$ the Petersen graph (PG) appears in all four cases. It is also a cubic graph of diameter 2. In addition, it is strongly regular, which means that any two adjacent vertices have a fixed number (0 in this case) of common neighbors and any two non-adjacent vertices have a fixed number (1 in this case) of common neighbors. Such an extraordinary structure is the reason why the Petersen graph appears frequently in graph theory as example or counterexample in numerous studies. Here it appears that the Petersen graph should be considered as a very good multiprocessor interconnection network. It is also remarkable that tightness t_1 and stt cannot be smaller than 10 for $n = 9$ and that only with one vertex more, when $n = 10$ their value can become 9 for the Petersen graph.

However, by tightness spt and t_2, the star on 10 vertices is as equally good topology as the Petersen graph.

The results for spt and t_2 perhaps suggest that stars are candidates for optimal topologies in general. However, such a conclusion is correct only for small graphs. In [6] it was shown that stars have tightness spt and t_2 asymptotically equal to $O(\sqrt{n})$ while hypercubes have equal values for all four types of tightness with asymptotical behavior $O((\log n)^2)$. On the other hand, 3-dimensional hypercube seems to be less suitable not only than the star S_8; $N(8, 6660)$, $N(8, 8469)$, C_8 and some other graphs also have smaller tightness values. Moreover, graphs $N(8, 6660)$ and $N(8, 8469)$ provide a smaller diameter with the same maximum vertex degree.

The problem of finding graphs with the smallest tightness values for a given number of vertices remains open in general.

References

[1] U. Borštnik, M. Hodoščček, and D. Janežič, *Fast parallel molecular simulations*, Croat. Chem. Acta, 78(2):211–216, 2005.

[2] V. Brankov, D. Cvetković, S. Simić, and D. Stevanović, *Simultaneous editing and multilabelling of graphs in system newGRAPH*, Univ. Beograd, Publ. Elektrotehn. Fak. Ser. Mat. 17:112–121, 2006.

[3] A. E. Brouwer, A. M. Cohen, and A. Neumaier, *Distance-regular Graphs*, Springer, Berlin, 1989.

[4] G. T. Byrd and B. A. Delagi, *Considerations for multiprocessor topologies*, Technical report, STAN-CS-87-1144, Department of Computer Science, Stanford University, 1987.

[5] D. Cvetković and T. Davidović, *Multiprocessor interconnection networks with small tightness*, Internat. J. Foundations Computer Sci., 20(5):941–963, 2009.

[6] D. Cvetković and T. Davidović, *Application of some graph invariants to the analysis of multiprocessor interconnection networks*, Yugosl. J. Oper. Res. 18(2):173–186, 2008.

[7] D. Cvetković and T. Davidović, *Exhaustive search for multiprocessor interconnection networks with small tightness value*, in: D. Teodorović, editor, *Proc. SYM-OP-IS 2008*, Soko Banja, Serbia, Sept. 14–17, 2008, pp. 329–332.

[8] D. Cvetković and T. Davidović, *Well-suited multiprocessor topologies with small number of processors*, Novi Sad J. Math., 38(3):209–217, 2008.

[9] D. Cvetković, T. Davidović, A. Ilić, S.K. Simić, *Graphs for small multiprocessor interconnection networks*, Appl. Math. Computation 217:2468–2480, 2010.

[10] D. Cvetković, V. Dimitrijević, and M. Milosavljević, *A survey of some non-standard traveling salesman problems*, Yugosl. J. Oper. Res. 2(2):163–185, 1992.

[11] D. Cvetković and I. Gutman, *Note on branching*, Croat. Chem. Acta 49:115–121, 1977.

[12] D. Cvetković and P. Rowlinson, *The largest eigenvalue of a graph – a survey*, Linear Multilinear Algebra 28:3–33, 1990.

[13] D. M. Cvetković, M. Doob, I. Gutman, and A. Torgašev, *Recent Results in the Theory of Graph Spectra*, North Holland, Amsterdam, 1988.

[14] D. M. Cvetković, M. Doob, and H. Sachs, *Spectra of Graphs: Theory and Applications*, (III edition), Johann Ambrosius Barth Verlag, Heidelberg–Leipzig, 1995.

[15] D. M. Cvetković and M. Petrić, *A table of connected graphs on six vertices*, Discrete Math. 50:37–49, 1984.

[16] D. Cvetković, P.'Rowlinson, S. Simić, *An Introduction to the Theory of Graph Spectra*, Cambridge University Press, Cambridge, 2009.

[17] T. Decker, B. Monien, and R. Preis, *Towards optimal load balancing topologies*, in *Proc. 6th EuroPar Conf.*, Lect. Notes Comput. Sci. 1900, Springer, Berlin, p.p. 277–287

[18] R. Dickmann, A. Frommer, and B. Monien, *Efficient scheme for nearest neighbor load balancing*, Parallel Comput. 25:789–812, 1999.

[19] R. Elsässer, R. Královič, and B. Monien, *Sparse topologies with small spectrum size*, Theor. Comput. Sci. 307:549–565, 2003.

[20] A. Ferreira and M. Morvan, *Models for parallel algorithm design: An introduction*, in: A. Migdalas, P. Pardalos, and S. Storøy, editors, *Parallel Computing in Optimization*, Kluwer, Dordrecht/Boston/London, 1997, p.p. 1–26.

[21] P. W. Fowler, G. Caporossi, and P. Hansen, *Distance matrices, Wiener indices, and related invariants of fullerenes*, J. Chem. Phys. A 105:6232–6242, 2001.

[22] P. W. Fowler and D. E. Manolopoulos, *An Atlas of Fullerenes*, Clarendon Press, Oxford, 1995.

[23] B. Ghosh, S. Muthukrishnan, and M. H. Schultz, *First and second order diffusive methods for rapid, coarse, distributed load balancing*, in: *Proc. 8th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'96)*, ACM Press, 1996, pages 72–81,

[24] A. Gupta and P. A. Dinda, *Inferring the topology and traffic load of parallel programs running in a virtual machine environment*, Lect. Notes Comput. Sci. 3277:125–143, 2005.

[25] I. Gutman, *Characteristic and matching polynomials of some compound graphs*, Publ. Inst. Math., Nouv. Sér. 27(41):61–66, 1980.

[26] Y. F. Hu and R. J. Blake, *An improved diffusion algorithm for dymanic load balancing*, Parallel Comput. 25:417–444, 1999.

[27] Y. F. Hu, R. J. Blake, and D. R. Emerson, *An optimal migration algorithm for dymanic load balancing*, Concurrency Pract. Exp. 10:467–483, 1998.

[28] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures, Arrays, Trees, Hypercubes*, Morgan Kaufmann, San Mateo, 1992.

[29] B. D. McKay, *Nauty user's guide (version 2.2)*, Technical report, http://cs.anu.edu.au/˜bdm/nauty, 2006.

[30] K. Qiu and S. K. Das, *Interconnection networks and their eigenvalues*, Int. J. Found. Comput. Sci. 14(3):371–389, 2003.

[31] V. Sarkar, *Partitioning and Scheduling Parallel Programs for Multiprocessors*, M.I.T. Press, Cambridge, MA, 1989.

[32] M. Sevkli and M. E. Aydin, *Parallel variable neighbourhood search algorithms for job shop scheduling problems*, IMA J. Math. Manage. 18(2):117–133, April 2007.

[33] G. C. Sih and E. A. Lee, *A compile-time scheduling heuristic for interconnection-constrained heterogeneous processor architectures*, IEEE Trans. Paral. Dist. Syst. 4(2):175–187, February 1993.

[34] S. Simić and D. Tošić, *The index of trees with specified maximum degree*, MATCH Commun. Math. Comput. Chem 54:351–362, 2005.

[35] T. Sørevik, *A programmer's view of parallel computers*, in: A. Migdalas, P. Pardalos, and S. Storøy, editors, *Parallel Computing in Optimization*, Kluwer, Dordrecht/Boston/London, 1997, p.p. 57–72.

[36] E. van Dam, *Graphs with few eigenvalues, An interplay between combinatorics and algebra*, PhD thesis, Center for Economic Research, Tilburg University, 1996.

[37] L. Wang, Z. P. Chen, and X. H. Jiang, *Ringed Petersen spheres connected hypercube interconnection networks*, in: *Proc. 10th IEEE Int. Conf. Eng. Complex Comput. Sys. (ICECCS'05)*, IEEE Computer Society, 2005, p.p. 127–131,

Appendix

The Table 3 given below contains some relevant data about 69 graphs with second type mixed tightness not exceeding 9.

Graphs are ordered first by n (the number of vertices), and within the groups with fixed n, by t_2. Columns of the table provide graph name, the number of vertices n, the number of edges e, the name(s) under which the graph appeared in [5], diameter D, maximum vertex degree Δ, the number of distinct eigenvalues m, the spectrum starting with the largest eigenvalue λ_1. Last four columns contain the values of the four types of tightness $t_1, \mathrm{stt}, \mathrm{spt}, t_2$.

As "the old names" we used different notation. First we distinguish the well known graphs such as complete graphs, circuits, stars, complete bipartite graphs, and so on. For graphs up to 5 vertices we used the notation from [14], while graphs on $n = 6$ vertices are marked primarily as in [15]. $N(n, j)$ denotes the j-th graph on n vertices generated by program **nauty**. PG denotes the well known Petersen graph.

TABLE 3. Graphs on up to 10 vertices with small tightness

graph	n	e	old name(s)	D	Δ	m	$\lambda_1, \lambda_2, \ldots, \lambda_n$	t_1	stt	spt	t_2
$\Omega_{2,1}$	2	1	$G_1 = K_2 = P_2$	1	1	2	**1**, -1	2	2	2	2
$\Omega_{3,1}$	3	3	$G_2 = K_3 = C_3$	1	2	2	**2**, -1, -1	4	4	4	4
$\Omega_{3,2}$		2	$G_3 = P_3 = S_3 = K_{1,2}$	2	2	3	**1.41**, 0, -1.41	6	6	4.23	4.23
$\Omega_{4,1}$	4	3	$G_8 = S_4 = K_{1,3}$	2	3	3	**1.73**, 0, 0, -1.73	9	9	5.19	5.19
$\Omega_{4,2}$		4	$G_7 = C_4 = K_{2,2}$	2	2	3	**2**, 0, 0, -2	6	6	6	6
$\Omega_{4,3}$		6	$G_4 = K_4$	1	3	2	**3**, -1, -1, -1	6	6	6	6
$\Omega_{4,4}$		3	$G_9 = P_4$	3	2	4	**1.62**, 1.62, 0.62, -0.62, -1.62	8	8	6.47	6.47
$\Omega_{4,5}$		4	G_6	2	3	4	**2.17**, 0.31, -1, -1.48	12	9	8.68	6.51
$\Omega_{4,6}$		5	G_5	2	3	4	**2.56**, 0, -1, -1.56	12	9	10.24	7.68
$\Omega_{5,1}$	5	5	$G_{28} = S_5 = K_{1,4}$	2	4	3	**2**, 0, 0, 0, -2	12	12	6	6
$\Omega_{5,2}$		5	$G_{27} = C_5$	2	2	3	**2**, 0.62, 0.62, -1.62, -1.62	6	6	6	6
$\Omega_{5,3}$		5	G_{23}	2	4	5	**2.34**, 0.47, 0, -1, -1.81	20	12	11.71	7.03
$\Omega_{5,4}$		6	$G_{22} = K_{2,3}$	2	3	3	**2.45**, 0, 0, 0, -2.45	9	9	7.35	7.35
$\Omega_{5,5}$		4	G_{29}	3	3	5	**1.85**, 0.77, 0, -0.77, -1.85	15	12	9.24	7.39
$\Omega_{5,6}$		6	G_{21}	2	3	5	**2.48**, 0.69, 0, -1.17, -2	15	9	12.41	7.44
$\Omega_{5,7}$		6	G_{20}	2	4	4	**2.56**, 1, -1, -1, -1.46	16	12	10.25	7.68
$\Omega_{5,8}$		10	$G_{10} = K_5$	1	4	2	**4**, -1, -1, -1, -1	8	8	8	8
$\Omega_{5,9}$		6	G_{18}	2	4	5	**2.69**, 0.33, 0, -1.27, -1.75	20	12	13.43	8.06
$\Omega_{5,10}$		5	G_{26}	3	3	5	**2.14**, 0.66, 0, -0.66, -2.14	15	12	10.68	8.54
$\Omega_{5,11}$		7	G_{17}	2	3	5	**2.86**, 0.32, 0, -1, -2.18	15	9	14.28	8.57
$\Omega_{5,12}$		4	$G_{30} = P_5$	4	2	5	**1.73**, 1, 0, -1, -1.73	10	10	8.66	8.66
$\Omega_{5,13}$		7	G_{16}	2	4	5	**2.94**, 0.62, -0.46, -1.47, -1.62	20	12	14.68	8.81
$\Omega_{5,14}$		5	G_{25}	3	3	5	**2.21**, 1, -0.54, -1, -1.67	15	12	11.07	8.86
$\Omega_{5,15}$		7	G_{15}	2	4	4	**3**, 0, 0, -1, -2	16	12	12	9

Table 3: Graphs on up to 10 vertices with small tightness (cont.)

graph	n	e	old name(s)	D	Δ	m	$\lambda_1, \lambda_2, \ldots, \lambda_n$	t_1	stt	spt	t_2
$\Omega_{6,1}$	6	5	$S_6 = K_{1,5} = CP(107) = N(6,1)$	2	5	3	**2.24**, 0, 0, 0, 0, -2.24	15	15	6.71	6.71
$\Omega_{6,2}$		7	$CP(93) = N(6,35)$	2	3	6	**2.39**, 0.77, 0.62, 0, -1.62, -2.16	18	9	14.35	7.17
$\Omega_{6,3}$		6	$CP(94) = N(6,3)$	2	5	5	**2.51**, 0.57, 0, 0, -1, -2.09	25	15	12.57	7.54
$\Omega_{6,4}$		6	$C_6 = CP(106) = N(6,49)$	3	2	4	**2**, 1, 1, -1, -1, -2	8	8	8	8
$\Omega_{6,5}$		5	$CP(109) = N(6,5)$	3	3	5	**2**, 1, 0, 0, -1, -2	15	12	10	8
$\Omega_{6,6}$		7	$CP(79) = N(6,17)$	2	5	5	**2.71**, 1, 0.19, -1, -1, -1.90	25	15	13.55	8.13
$\Omega_{6,7}$		8	$CP(72) = N(6,89)$	2	3	6	**2.74**, 0.71, 0.62, -0.23, -1.62, -2.22	18	9	16.45	8.22
$\Omega_{6,8}$		5	$CP(108) = N(6,2)$	3	4	5	**2.07** , 0.84, 0, 0, -0.84, -2.07	20	16	10.37	8.30
$\Omega_{6,9}$		8	$CP(69) = N(6,90)$	2	3	6	**2.79**, 1, 0.62, -1, -1.62, -1.79	18	9	16.75	8.37
$\Omega_{6,10}$		8	$CP(71) = N(6,36)$	2	4	5	**2.80**, 0.85, 0, 0, -1.20, -2.45	20	12	13.98	8.39
$\Omega_{6,11}$		8	$CP(68) = N(6,57)$	2	4	6	**2.81**, 1, 0.53, -1, -1.34, -2	24	12	16.88	8.44
$\Omega_{6,12}$		7	$CP(75) = N(6,8)$	2	5	5	**2.81**, 0.53, 0, 0, -1.34, -2	25	15	14.07	8.44
$\Omega_{6,13}$		6	$CP(105) = N(6,19)$	3	3	6	**2.11**, 1, 0.62, -0.25, -1.62, -1.86	18	12	12.69	8.46
$\Omega_{6,14}$		8	$K_{2,4} = CP(73) = N(6,13)$	2	4	3	**2.83**, 0, 0, 0, 0, -2.83	12	12	8.49	8.49
$\Omega_{6,15}$		8	$CP(66) = N(6,39)$	2	4	5	**2.90**, 0.81, 0, 0, -1.71, -2	20	12	14.52	8.71
$\Omega_{6,16}$		8	$CP(61) = N(6,32)$	2	5	6	**2.95**, 1.16, 0, -1, -1.29, -1.82	30	15	17.68	8.84
$\Omega_{6,17}$		6	$CP(102) = N(6,18)$	3	3	6	**2.25**, 0.80, 0.55, -0.55, -0.80, -2.25	18	12	13.48	8.99
$\Omega_{6,18}$		9	$CP(51) = N(6,93)$	2	3	4	**3**, 1, 0, 0, -2, -2	12	9	12	9
$\Omega_{6,19}$		9	$K_{3,3} = CP(52) = N(6,71)$	2	3	3	**3**, 0, 0, 0, 0, -3	9	9	9	9
$\Omega_{7,1}$	7	6	$S_7 = K_{1,6} = N(7,1)$	2	6	3	**2.45**, 0, 0, 0, 0, 0, -2.45	18	18	7.35	7.35
$\Omega_{7,2}$		9	$N(7,337)$	2	3	5	**2.66**, 1.21, 0.62, 0.62, -1.62, -1.62, -1.87	15	9	13.28	7.97
$\Omega_{7,3}$		7	$C_7 = N(7,292)$	3	2	4	**2**, 1.255, 1.25, -0.45, -0.45, -1.80, -1.80	8	8	8	8
$\Omega_{7,4}$		9	$N(7,156)$	2	4	6	**2.68**, 1, 0.64, 0, 0, -2, -2.32	24	12	16.09	8.04
$\Omega_{7,5}$		7	$N(7,3)$	2	6	5	**2.68**, 0.64, 0, 0, 0 ,-1, -2.32	30	18	13.41	8.04

Table 3: Graphs on up to 10 vertices with small tightness (cont.)

graph	n	e	old name(s)	D	Δ	m	$\lambda_1, \lambda_2, \ldots, \lambda_n$	t_1	stt	spt	t_2
$\Omega_{7,6}$	7	9	$N(7,75)$	2	4	6	**2.75**, 0.84, 0.62, 0, 0, -1.62, -2.59	24	12	16.51	8.25
$\Omega_{7,7}$		8	$N(7,23)$	2	6	6	**2.86**, 1, 0.32, 0, -1, -1, -2.18	36	18	17.13	8.57
$\Omega_{7,8}$		6	$N(7,5)$	3	4	5	**2.18**, 1.13, 0, 0, 0, -1.13, -2.18	20	16	10.88	8.70
$\Omega_{7,9}$		10	$N(7,624)$	2	3	7	**2.90**, 1.41, 0.81, 0, -1.41, -1.71, -2	21	9	20.32	8.71
$\Omega_{7,10}$		10	$N(7,514)$	2	3	6	**2.90**, 0.81, 0.73, 0, 0, -1.71, -2.73	18	9	17.42	8.71
$\Omega_{7,11}$		8	$N(7,8)$	2	6	5	**2.94**, 0.66, 0, 0, 0, -1.37, -2.24	30	18	14.72	8.83
$\Omega_{7,12}$		7	$N(7,92)$	3	3	6	**2.21**, 1, 1, 0, -0.54, -1.68, -2	18	12	13.29	8.86
$\Omega_{7,13}$		10	$N(7,448)$	2	4	7	**2.98**, 1.33, 0.65, 0, -1, -1.77, -2.19	28	12	20.86	8.94
$\Omega_{7,14}$		9	$N(7,324)$	2	4	7	**2.97**, 0.80, 0.70, 0.45, -0.55, -2.12, -2.25	28	12	20.77	8.90
$\Omega_{7,15}$		9	$N(7,219)$	2	6	4	**3**, 1, 1, -1, -1, -1, -2	24	18	12	9
$\Omega_{8,1}$	8	7	$S_8 = K_{1,7} = N(8,1)$	2	7	3	**2.65**, 0, 0, 0, 0, 0, 0, -2.65	21	21	7.94	7.94
$\Omega_{8,2}$		8	$N(8,3)$	2	7	5	**2.54** , 0.69, 0, 0, 0, 0, -1, -2.84	35	21	14.22	8.53
$\Omega_{8,3}$		11	$N(8,1039)$	2	4	8	**2.90**, 1.30, 0.81, 0.62, 0, -1.62, -1.71, -2.30	32	12	23.23	8.71
$\Omega_{8,4}$		11	$N(8,342)$	2	5	6	**2.98**, 1.13, 0.65, 0, 0, 0, -2.07, -2.68	30	15	17.86	8.93
$\Omega_{8,5}$		9	$N(8,30)$	2	7	6	**3**, 1, 0.41, 0, 0, -1, -1, -2.41	42	21	18	9
$\Omega_{8,6}$		12	$N(8,8469)$	2	3	6	**3**, 1.56, 0.62, 0.62, 0, -1.62, -1.62, -2.56	18	9	18	9
$\Omega_{8,7}$		12	$N(8,6660)$	2	3	5	**3**, 1, 1, 0.41, 0.41, -1, -2.41,-2.41	15	9	15	9
$\Omega_{9,1}$	9	8	$S_9 = K_{1,8} = N(9,1)$	2	8	3	**2.83**, 0, 0, 0, 0, 0, 0, 0, -2.83	24	24	8.49	8.49
$\Omega_{9,2}$		9	$N(9,3)$	2	8	5	**3**, 0.73, 0, 0, 0, 0, 0, -1, -2.73	40	24	15	9
$\Omega_{10,1}$	10	9	$S_{10} = K_{1,9} = N(10,1)$	2	9	3	**3**, 0, 0, 0, 0, 0, 0, 0, 0, -3	27	27	9	9
$\Omega_{10,2}$		15	$PG = N(10,8027956)$	2	3	3	**3**, 1, 1, 1, 1, 1, -2, -2, -2, -2	9	9	9	9