DOI: 10.14529/jsfi150402

Creating interconnect topologies by algorithmic edge removal: MOD and SMOD graphs

Marek T. Michalewicz [1, 2, 3], *Łukasz P. Orłowski* [1, 3, 4], *Yuefan Deng* [1, 3, 4]

We introduce a method of constructing classes of graphs by algorithmic removal of entire groups of edges. Our approach to creating new classes of graphs is to focus entirely on the structure and properties of an adjacency matrix. At an initialisation step of the algorithm we start with a complete (fully connected) graph.

In Part I we present MOD and arrested MOD graphs resulting from removal of square blocks of edges at each iteration and substitution of removed blocks with a diagonal matrix with one extra pivotal element along the main diagonal. The MOD graphs possess unique and useful properties. All important graph measures are easily expressed in analytical form and are presented in the paper. Several important properties of MOD graphs are compared very favourably with graphs representing common interconnect topologies: hypercube, 3D and 5D tori, TOFU and dragonfly. This lead us to consider MOD and arrested MOD graphs as interesting candidates for effective supercomputer interconnects.

In Part II, at each iterative step we successively remove triangular shapes from the adjacency matrix. This iterative process leads to the final matrix which has two Sierpiński gaskets aligned along the main diagonal. It will be shown below that this new class of graphs *is not* a Sierpiński graph, since it is the adjacency matrix which has a structure of a Sierpiński gasket, and not a graph described by this matrix. We call this new class of graphs Sierpiński-Michalewicz-Orłowski-Deng (SMOD) graphs. The most remarkable property of the SMOD class of graphs, is that irrespective of the graph order, the diameter is *constant* and equals 2. The size of the graph, or the total number of edges, is about 10% of the size of a complete graph of the same order.

We analyse important graph theoretic characteristics related to the topology such as diameter as a function of graph order, size, mean path length, ratio of the graph size to the size of a complete graph of the same order, and some spectral properties.

Keywords: supercomputer interconnects, big data, exascale computing, graph theory, topology of graphs, classes of graphs, graph generation.

Introduction

The critical design characteristics of the future exascale supercomputer systems will be their interconnect topology, routing algorithms, and connect bandwidths. In this paper we propose a new algorithmic method of constructing well connected graphs, with reasonably small diameter, and low mean path, which may be useful in consideration of future exascale systems. We focus exclusively on the graph theoretic characteristics of the interconnect, hence the analysis of real supercomputer interconnects is restricted to properties derived from graph topology.

During the past 30 years, graphs like hypercube [16], tori [11], and trees [24] have been widely adopted as the topologies of choice for supercomputer networks [18]. More recently, butterfly graphs [22] and dragonfly [13, 19, 21] are entering the market. As the number of computing units in a supercomputer grows to a few million processing cores such as the fastest Tianhe-2 [1, 20] computer in June 2015 with 3.12 million cores, conventional wisdom in selecting a

[1] A*STAR Computational Resource Centre, Singapore 138632, Singapore
[2] A*STAR Institute of High Performance Computing, Singapore 138632, Singapore
[3] Institute for Advanced Computational Science, Stony Brook University, New York 11794-3600, USA
[4] Department of Applied Mathematics and Statistics, Stony Brook University, New York 11794-3600, USA

topology might not lead to the optimal interconnect design. We must leverage on mathematics to discover the state-of-the-art topologies for more efficient networks.

Design of efficient networks requires optimization in multiple dimensions. First, in mathematical dimension, we must design the best network topologies. Second, in engineering dimension, these topologies must be implementable using existing electronic technologies and fit the current architecture standards. Third, in computer science dimension, the wired topologies must be consistent with an efficient routing scheme to generate reliable and fast networks that are easy to use. Finally, in economics dimension, the system must not be too costly. In this report, we focus on optimization of the first dimension, i.e., mathematical aspect. Hence, we are not concerned here with the study of bi-sectional bandwidth, routing tables and routing or physical layout of cables and fibres.

Our method is to explore graph generation techniques that start with adjacency matrices of arbitrary size and subsequently operate solely on adjacency matrices, which yield graphs, rather than generating graphs via diagram constructions first and treating the adjacency matrix as a derived characteristic [15]. Additionally, in our approach we always start with a complete graph of a given order and we proceed by eliminating the edges, never by starting from a smaller order graph and growing it bigger by embedding or other decorations.

We only focus on the topology design, excluding the underlying fabric, therefore the adjacency matrices that we build, and in what follows always denote by A, are symmetric with values of **0** when there is no edge, and **1** when vertices are connected. This translates into unweighted, undirected and connected graphs.

We study graphs whose topologies would potentially yield optimal interconnects for Supercomputers and Big Data systems, hence we turn our attention to the following characteristics:

1. Minimum diameter
2. Good paths distribution
3. Low mean path length
4. Small size, *i.e.* number of edges
5. Small degrees of vertices

Part I

1. Michalewicz-Orłowski-Deng (MOD) Algorithm

We consider graphs $G = (N, L)$, where N is a set of vertices and L is a set of undirected edges (i.e. set of unordered pairs of vertices). The graph order $|N|$, *i.e.* numbers of vertices, is restricted to powers of 2: $|N| = n = 2^m$, $m \in \mathbb{N}$, $m \geq 2$.

Michalewicz-Orłowski-Deng (MOD) algorithm begins with a complete graph and in every step reduces the number of remaining edges by about a half, resulting in a graph with $O(nlog_2 n)$ edges. The algorithm converges in $log_2 n - 1 = m - 1$ steps.

1.1. The algorithm sequence

The MOD algorithm proceeds through the following sequence of steps:

1. Start with an adjacency matrix of a complete graph
2. Treat the entire adjacency matrix as a single block on the main diagonal

3. Split the block(s) on the main diagonal into quadrant(s) of 4 sub blocks: two diagonal blocks and; upper and lower counter-diagonal blocks
4. Substitute every upper counter-diagonal block with a sum of identity matrix plus pivotal edge
5. Substitute every lower counter-diagonal block with transpose of upper counter-diagonal block of step 4.
6. Repeat steps 3-5 or stop if the size of blocks is 2.

Mathematically, the algorithm looks in the following way:

$$A = J - I$$

a) Initialization of the MOD algorithm - step 1 above

$$A \leftarrow \left(\begin{array}{c|c} B_d^{(1)} & B_u^{(1)} \\ \hline B_l^{(1)} & B_d^{(1)} \end{array}\right)$$
$$B_u^{(1)} = I + P_e$$
$$B_l^{(1)} = B_u^{(1)T}$$

b) Iteration 1 of the MOD algorithm - steps 2-5 above

$$\vdots$$

$$B_d^{(1)} \leftarrow \left(\begin{array}{c|c} B_d^{(2)} & B_u^{(2)} \\ \hline B_l^{(2)} & B_d^{(2)} \end{array}\right)$$
$$B_u^{(2)} = I + P_e$$
$$B_l^{(2)} = B_u^{(2)T}$$

c) Iteration 2 of the MOD algorithm - step 6 above

$$B_d^{(m-1)} \leftarrow \left(\begin{array}{c|c} B_d^{(m)} & B_u^{(m)} \\ \hline B_l^{(m)} & B_d^{(m)} \end{array}\right)$$
$$B_u^{(m)} = I + P_e$$
$$B_l^{(m)} = B_u^{(m)T}$$

d) Final, m^{th} iteration of the algorithm.

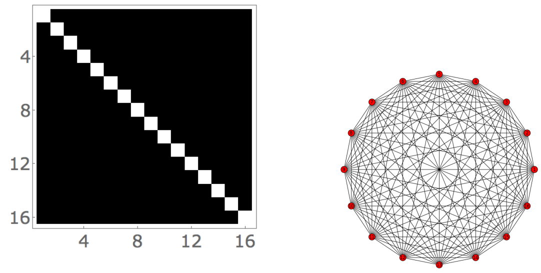
Figure 1. Sequence of steps of the MOD algorithm

here J is an all-ones matrix, I is an identity matrix, $B_d^{(p)} \in \mathbb{R}^{k \times k}$ are a diagonal block in the p^{th} step of the algorithm, $P_e \in \mathbb{R}^{k \times k}$ is the pivotal element matrix defined as $(P_e)_{ij} = \begin{cases} 1 & i = k, j = 1 \\ 0 & \text{otherwise} \end{cases}$

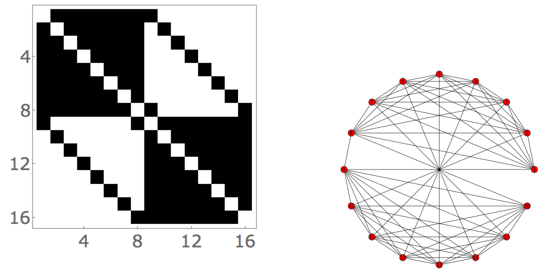
$B_u^{(p)}$and $B_l^{(p)}$ are respectively upper and lower counter-diagonal blocks in the p^{th} step of the algorithm and $k = \frac{n}{2^p}$.

1.2. Visualisation of steps of the MOD algorithm

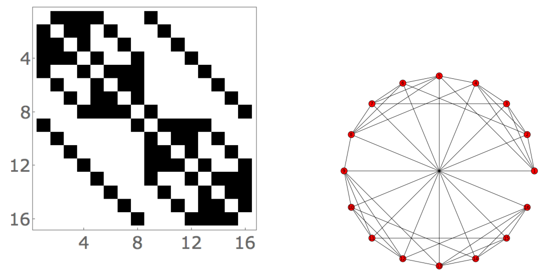
The figure below illustrates adjacency matrices generated in every step of the MOD algorithm (on the left) and the diagrams of graphs with a group of edges removed at every step (on the right).



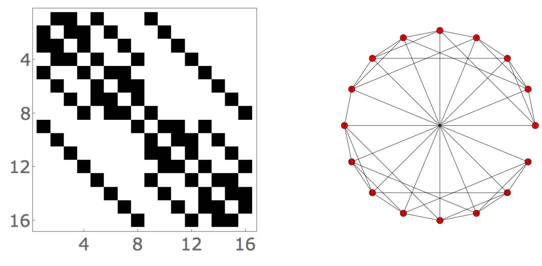
a) Initialization



b) Iteration 1



c) Iteration 2

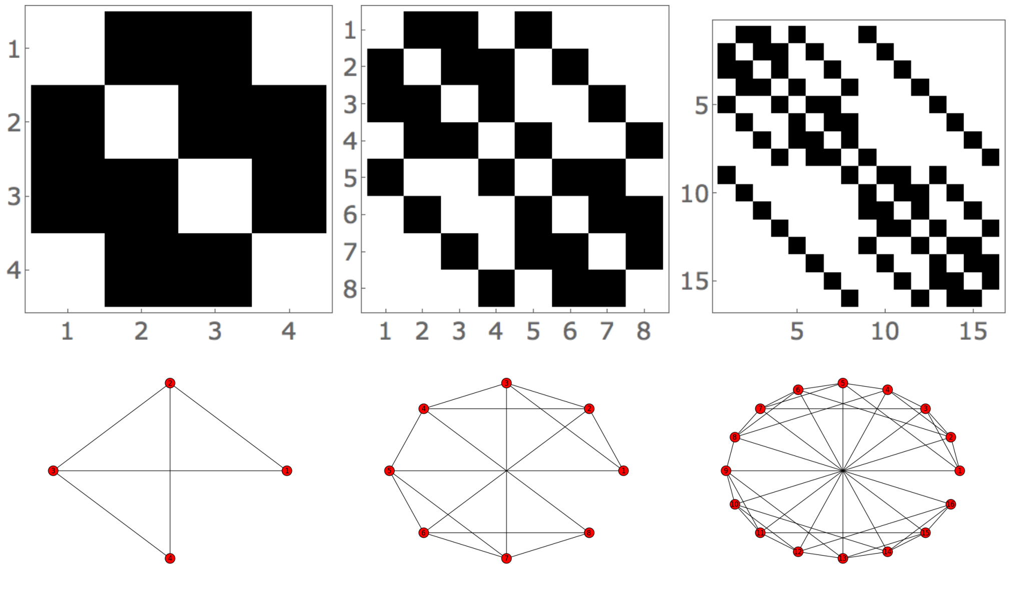


d) Iteration 3, final

Figure 2. Steps of the MOD algorithm generating MOD graph with 16 vertices

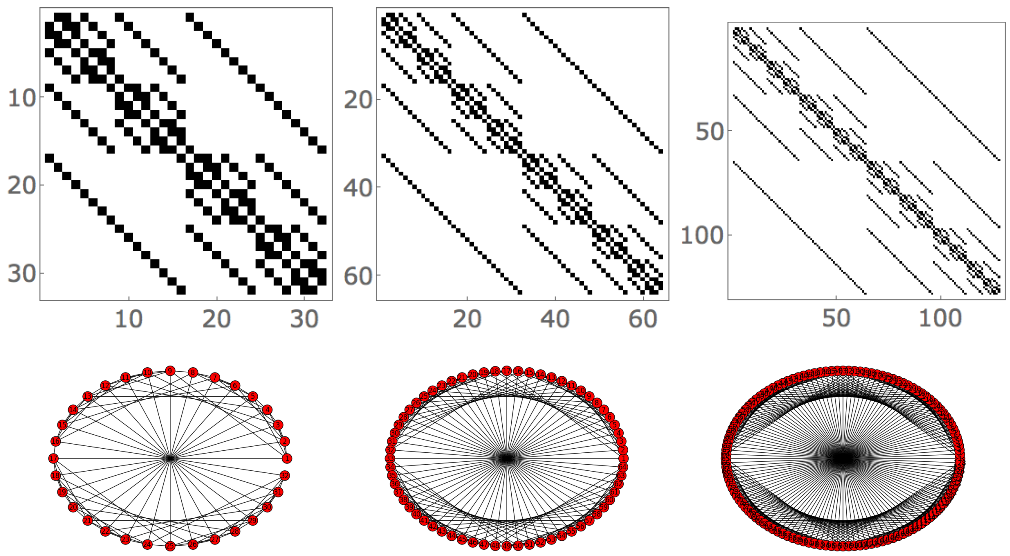
1.3. Visualisation of low order MOD graphs

The following figure exemplifies adjacency matrices generated by the MOD algorithm, along with corresponding graphs representation.



a) 4 vertex MOD **b)** 8 vertex MOD **c)** 16 vertex MOD

Figure 3. MOD graphs and their adjacency matrices for small $|N| = n = 2^m$ graphs with $2 \leq m \leq 4$



a) 32 vertex MOD **b)** 64 vertex MOD **c)** 128 vertex MOD

Figure 4. MOD graphs and their adjacency matrices for small $|N| = n = 2^m$ graphs with $5 \leq m \leq 7$

2. Properties of MOD graphs

In this section we collect results for several basic MOD graph properties such as the diameter, mean path length, size of the graph (*i.e.* number of edges), the ratio of MOD graph size to complete graph size of the same order, and the vertex degrees. It turns out that most of these properties can be derived analytically and are expressed by simple relationships. We also study the spectra of adjacency matrices and attempt to relate them to other graph properties.

2.1. Diameter and mean path length

Two very important properties of graphs are the diameter and mean path length. The diameter of a graph is the greatest distance between any two vertices in a graph, where distance between two vertices is defined as the shortest of all paths connecting these vertices.

The diameter of MOD graphs is found to be

$$d_{MOD}(n) = log_2 n = m, \qquad \text{for } 2 \leq m \leq 3 \tag{1}$$

$$d_{MOD}(n) = log_2 n - 1 = m - 1, \quad \text{for } m \geq 4 \tag{2}$$

The mean path length, l_G, is the average of distances between any two vertices in a graph, defined as follows

$$l_G = \frac{1}{n(n-1)} \sum_{i \neq j} d(v_i, v_j) = \frac{1}{n(n-1)} \sum_{i \neq j} (D)_{ij} \tag{3}$$

where $d(v_i, v_j)$ denotes distance between i^{th} and j^{th} vertex and D denotes a distance matrix corresponding to a graph G. The latter formula provides a prescription for computation of l_G.

The mean path length of the MOD graphs converges to half of the diameter, for large n.

$$\lim_{n \to \infty} \left(\frac{1}{n(n-1)} \sum_{i \neq j} (D)_{ij} \right) = \frac{1}{2} d_{MOD}(n) = \frac{1}{2}(m-1) \tag{4}$$

The figure below depicts a diameter and mean path of the MOD graphs as a function of graph order n.

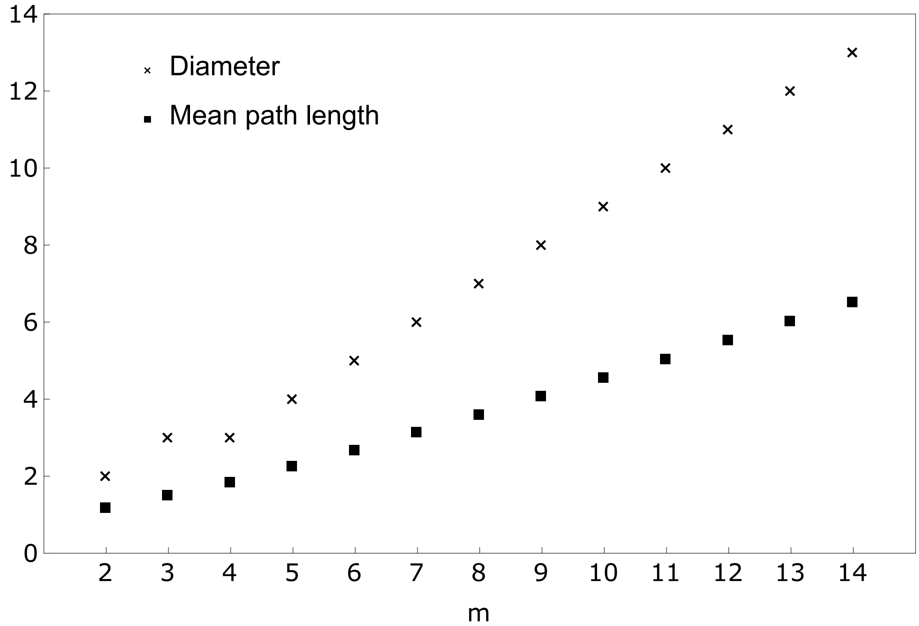


Figure 5. Diameter and mean path of the MOD graphs as a function of graph order n

2.2. Size of MOD graphs and their vertex degrees

Theorem 1. The size of a MOD graph of order n is given by:

$$\begin{aligned} |L| &= \frac{n}{2}(log_2 n + 1) - 1 \\ &= (m+1)2^{m-1} - 1 \end{aligned} \tag{5}$$

Where $n = 2^m$

Proof. We prove the formula by counting the edges removed at every step of the algorithm.

$$\begin{aligned} |L| &= \frac{1}{2}\left[n(n-1) - 2(\frac{n}{2})^2 + n + 2 - 2^2(\frac{n}{2^2})^2 + n + 2^2 + ... - 2^m(\frac{n}{2^m}) + n\right] \\ &= \frac{1}{2}\left[n(n-1) - n^2\sum_{i=1}^{m}\frac{1}{2^i} + nm + \sum_{i=1}^{m-1}2^i\right] = \frac{1}{2}\left[nm + 2n^2 - n - n^2\sum_{i=0}^{m}\frac{1}{2^i} + \sum_{i=0}^{m-1}2^i - 1\right] \\ &= \frac{1}{2}\left[nm + 2n^2 - n - \frac{n^2}{2^m}(2^{1+m} - 1) + 2^m - 2\right] = \frac{1}{2}\left[nm + 2n^2 - n - 2n^2 + \frac{n^2}{2^m} + 2^{1+m} - 2\right] \\ &= \frac{n}{2}(log_2 n + 1) - 1. \end{aligned}$$

□

The following figure shows how the size of the MOD graphs increases as a function of the graph order $n = 2^m$ grows from 4 to 16,384 (or m grows from 2 to 14).

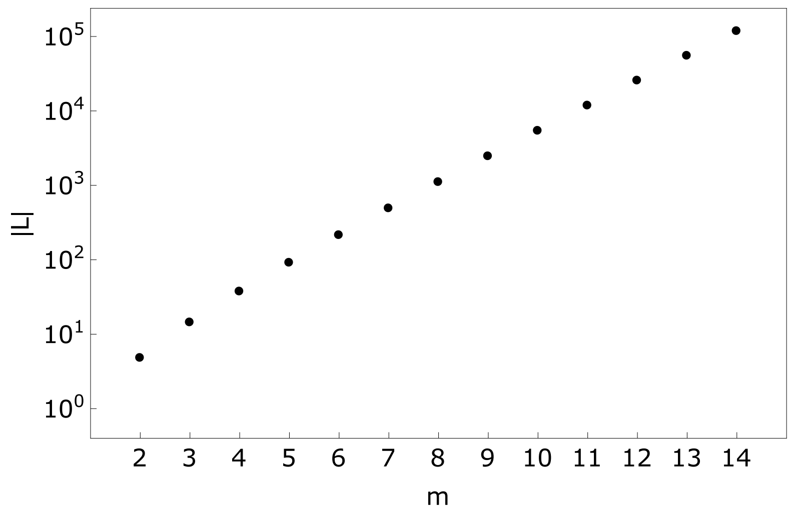


Figure 6. Size of MOD graphs for $2 \leq m \leq 14$

All the vertices of the MOD graph have a constant degree $m+1 = log_2 n + 1$, except for two vertices, numbered 1^{st} and n^{th} in adjacency matrices which are of degree $m = log_2 n$. This property can be expressed in the following way:

$$Au = (m, m+1, ..., m+1, m)^T. \tag{6}$$

where A is the adjacency matrix and $u^T = (1, \ldots, 1)$

2.3. Bisections of MOD graphs

Bisection is a subset of edges, that, when removed, separates a graph into two disconnected components of equal order (or almost equal for graphs of odd order). In the network theory, the quantity of interest is the bisection bandwidth, which can be derived from minimum bisection (a graph theoretic property) by multiplying each edge from the cut by its weight (or bandwidth in the network theory language). Here we assume that each edge has weight equal 1, which provides the best basis for topology comparisons. In this section we look at the minimum bisection of MOD graphs. We do not formally derive the formula for the number of edges in the minimum bisection, but instead arrive at the formula after a discussion to give the intuition of what is the minimum bisection of a MOD graph.

Let us observe, that every vertex v_k in a MOD graph of order n is directly connected to a vertex $v_{(k+n/2) \mod n}$. Furthermore every vertex v_k is connected to v_{k-1} and v_{k+1}, except for v_1 and v_n. This means that the minimum bisection (*i.e.* a bisection with the minimum number of edges) has to contain all the edges connecting the v_k with $v_{(k+n/2) \mod n}$ for $1 \leq k \leq n/2$ plus one edge connecting $v_{n/2}$ and $v_{n/2+1}$. This is presented in the following figure:

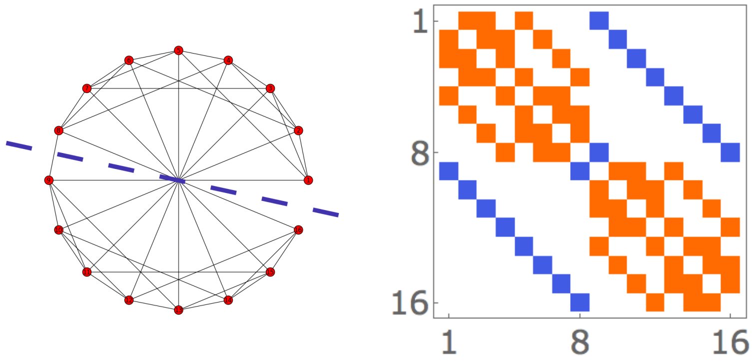


Figure 7. Minimum bisection of MOD graph of order 16 presented in graph (the cut in blue) and its adjacency matrix (entries in blue)

This leads us to an observation that the number of edges in the minimum bisection in a MOD graph is

$$|\min_{bisections} B(V_1, V_2)| = \frac{n}{2} + 1 = 2^{m-1} + 1 \tag{7}$$

Where V_1 and V_2 are two disjoint sets of vertices of equal size, whose sum is V.

In terms of graphs suited for topologies of interconnects, the higher the size of the minimum bisection, the better, meaning that the network based on such topology allows more data to flow from one half of the network to the other under uniform traffic pattern [18].

The following table summarises the minimum bisections and some other parameters of topologies that most resemble MOD graphs.

Table 1. Formulas for most important parameters including bisection width of hypercube, tori and MOD. Note that MOD has two vertices of degree m as depicted in (6)

Topology	Order	Degree	Diameter	Bisection
Hypercube	2^m	m	m	2^{m-1}
3D Torus	m^3	6	$3m/2$	$2m^2$
5D Torus	m^5	10	$5m/2$	$2m^4$
MOD	2^m	$m+1$	$m-1$	$2^{m-1}+1$

2.4. Edge ratio of MOD graphs

Definition 1. Edge ratio of a graph $G = (N, L)$ is a ratio of the size of a graph, $|L|$, to the size of a complete graph of the same order, i.e.:

$$L_{ratio} = \frac{2|L|}{|N|(|N|-1)}. \tag{8}$$

The edge ratio of the MOD graphs follows immediately from (5) and is given by:

$$L_{ratio} = \frac{nlog_2 n + 2n - 2}{n^2 - n} \tag{9}$$

$$= \frac{(m+2)2^m - 2}{2^m(2^m - 1)} \tag{10}$$

In the limit of large n an edge ratio of the MOD graph converges to 0, *i.e.*

$$\lim_{n\to\infty} L_{ratio} = 0 \tag{11}$$

2.5. Eigenvalues of the MOD graphs

Although the interpretation and meaning of *all* eigenvalues of adjacency matrices is yet unclear and subject to the study of spectral graph theory [17, 25], there are a few properties of MOD graphs which we understand and discuss here. The distribution of eigenvalues for MOD graphs with $2 \leq m \leq 14$ is represented below:

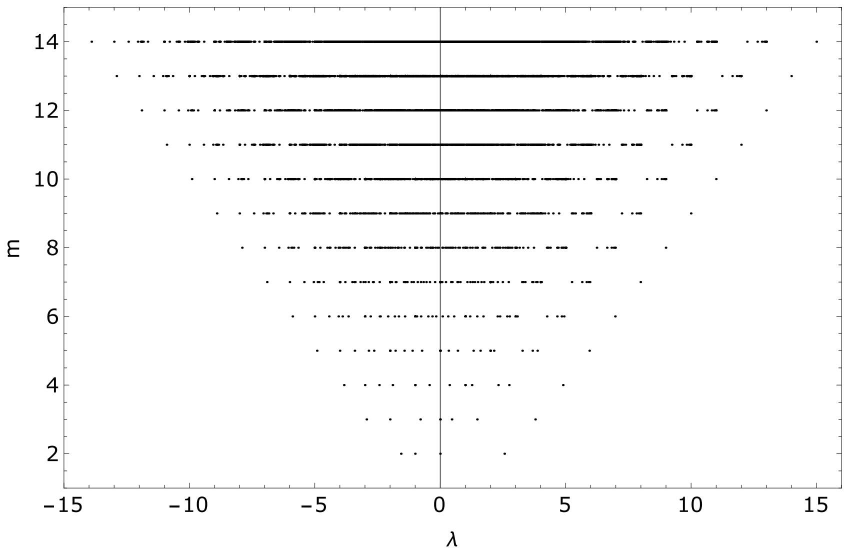


Figure 8. Distribution of eigenvalues for MOD graphs with $2 \leq m \leq 14$.

We denote the greatest eigenvalues as λ_1 and the least one as λ_n. We also observe that the average vertex degree is bounded by λ_1

$$\frac{1}{n}u^T A u = \frac{1}{n}\sum_{i,j}^{n}(A)_{ij} = 2\frac{|L|}{|N|} < \lambda_1 \tag{12}$$

and approaches λ_1 as the order of graph grows large. It follows trivially from Eq. 5 that

$$\lim_{n\to\infty} \lambda_1 = \lim_{n\to\infty} 2\frac{|L|}{|N|} = m + 1 \tag{13}$$

Furthermore, one sees that

$$\lambda_{n-1} = -m + 1 \tag{14}$$

$$\lambda_n \approx -m \tag{15}$$

We observe from Eq. 12 and Eq. 14 that for MOD graphs the greatest eigenvalue is equal to the spectral radius i.e. $\lambda_1 = \rho(A(G))$. The eigenvalues are bound as follows

$$-m < \lambda_i < m + 1, \text{ for all } i \in [1, n] \tag{16}$$

Finally, since: i) trace of the matrix is invariant with respect to similarity transformations, and, ii) the sum of all eigenvalues of an adjacency matrix of a graph with no self-loops is 0: $tr(A) = 0$.

$$\sum_{i=1}^{n} \lambda_i = 0 \tag{17}$$

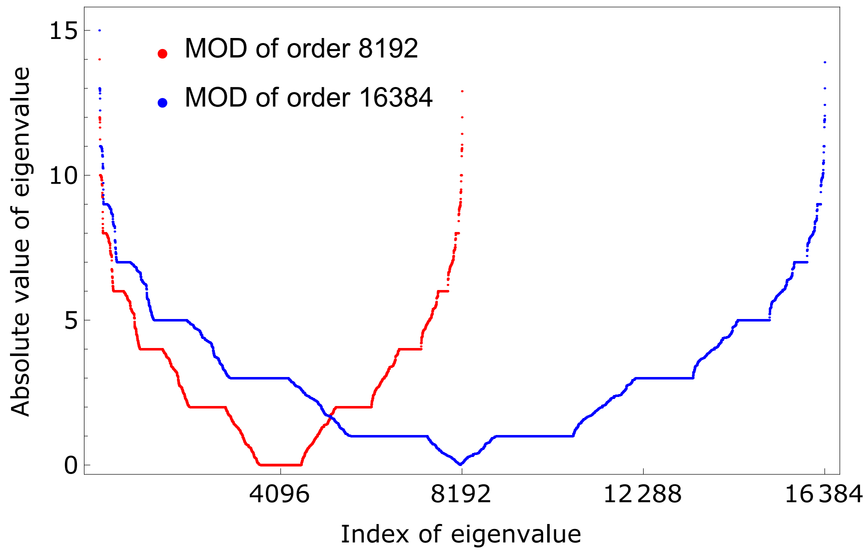
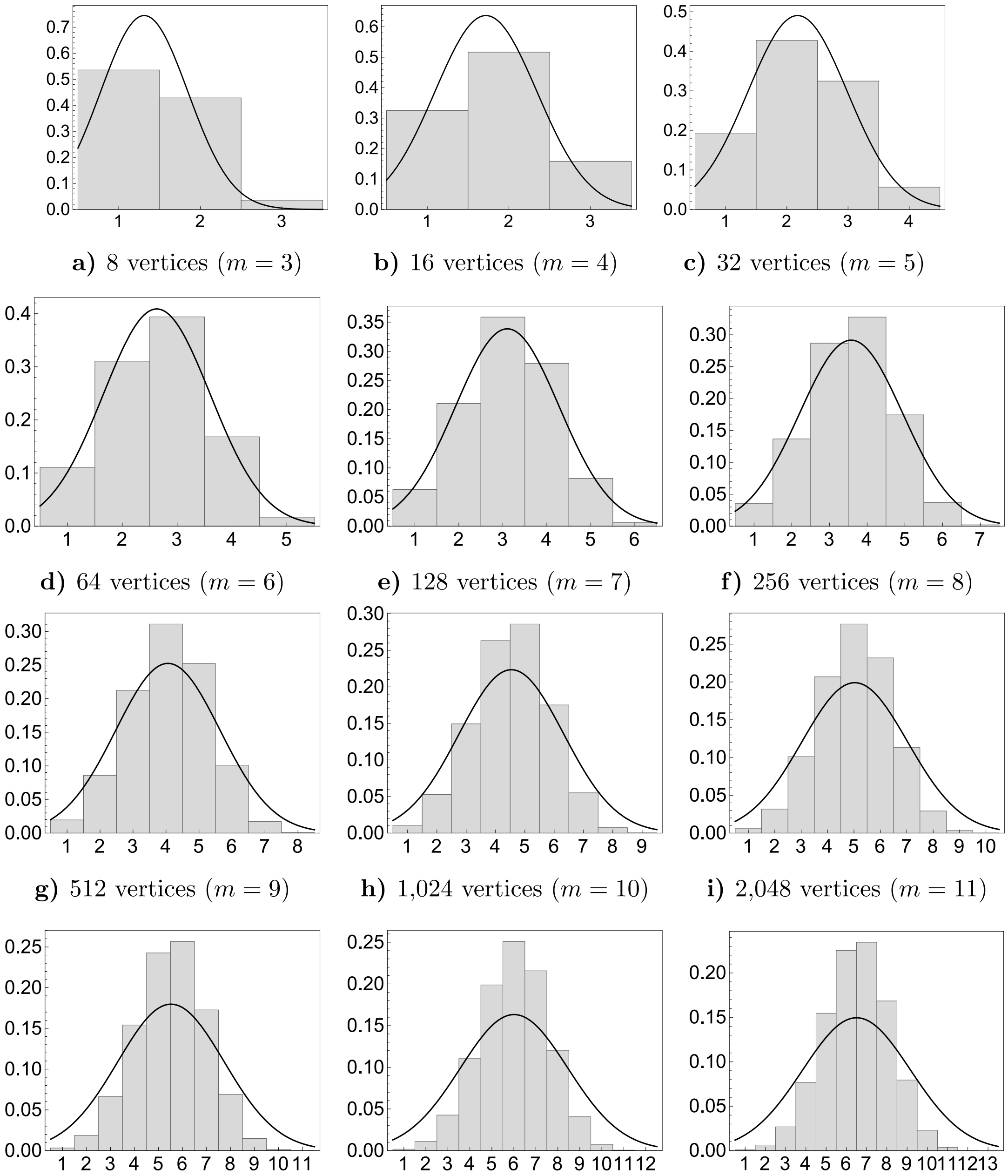


Figure 9. Absolute values of eigenvalues for the MOD graphs of order $n = 2^{13}$ (red) and $n = 2^{14}$ (blue). NB For $j \geq \frac{n}{2}$, $\lambda_j \leq 0$

The Figure 7 above represents the absolute values of all eigenvalues of the MOD graphs of order $n = 2^{13}$ (red) and $n = 2^{14}$ (blue). It is clearly seen that there are regions of the spectra where all eigenvalues assume natural number value, and are highly degenerate. For m odd, the integer eigenvalues are even, for m even, they are odd. Another fact that can be observed from the plots, is that very nearly half of eigenvalues are negative and half positive. The minimum of the absolute value of eigenvalues is non-degenerate for even values of m and degenerate for odd values of m and in both cases falls at or is centred (respectively) at $n/2$. For odd values of m the minimum of the absolute value of eigenvalues is equal zero, and for even m is slightly above zero.

2.6. Distance distribution of MOD graphs

The histogram of all distances expresses the discrete probability distribution of a distance between any two vertices in a graph. This distribution is an essential signature of a graph. It relates to the mean path length and the maximum distance, *i.e.* the graph diameter. In the context of practical supercomputer interconnect topologies distance distribution translates to network latency between the given pair of ports in the network. There are important implications of distribution of distances on the graph traversal, which in terms of network topologies translate to flow control and routing algorithms.



a) 8 vertices ($m = 3$) **b)** 16 vertices ($m = 4$) **c)** 32 vertices ($m = 5$)

d) 64 vertices ($m = 6$) **e)** 128 vertices ($m = 7$) **f)** 256 vertices ($m = 8$)

g) 512 vertices ($m = 9$) **h)** 1,024 vertices ($m = 10$) **i)** 2,048 vertices ($m = 11$)

j) 4,096 vertices ($m = 12$) **k)** 8,192 vertices ($m = 13$) **l)** 16,384 vertices ($m = 14$)

Figure 10. Distribution of distances in MOD graphs with $4 \leq m \leq 14$

Figure 10 depicts histograms of distances in MOD graphs of order $n = 2^m$, for $4 \leq m \leq 14$ overlaid with probability density functions of normally distributed random variable with the same mean value and variance as respective samples. Although at first it might seem that distances are normally distributed - they are not. The histograms of distances in MOD graphs have heavier tails than normal distribution.

A normal distribution (and centralised distributions) results in very predictable distances, meaning that the expected value of path length is equal (or close) to mean path length. Right skewed distributions (those with positive skewness or third standardised moment) have majority of short paths, a desirable characteristic, which may give a performance boost to a routing algorithm, however the vertices which are far away from each other will have negative impact on global communication *e.g. gather and scatter*. Left skewed distributions (with negative skewness) might intuitively seem undesirable. However, for topologies with a very short diameter, such as the dragonfly topology, skewness is not a decisive feature which might disqualify the topology from being a very effective solution.

The most important final point here is that irrespective of the underlying topology, and how good the distance distribution might seem, the ultimate criterion is a communication pattern *within* the algorithm, and how well it matches the hardwired supercomputer network topology. This topic deserves entirely separate treatment. Of course, some common stencils and known implemented algorithms might map very well on standard supercomputing interconnect topologies (see Section 5).

3. Graphs resulting from halted MOD algorithm: arrested MOD graphs

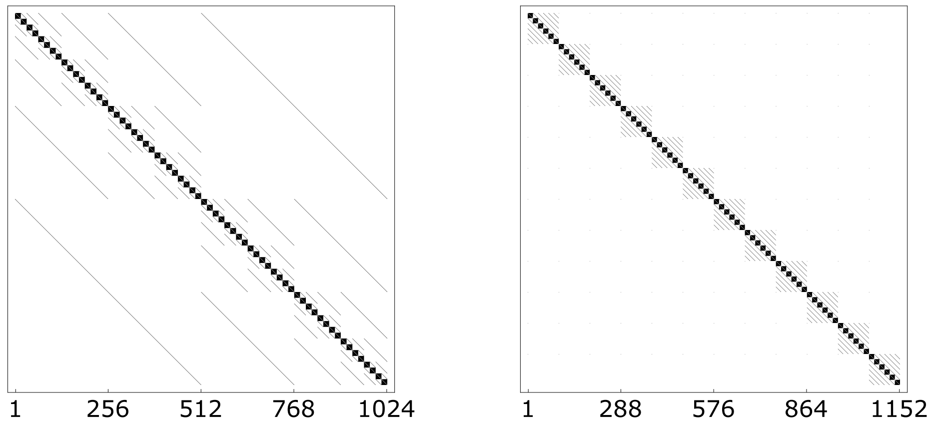
One very attractive feature of the graph creation algorithms presented here, and especially the MOD algorithm, is an ability to halt the iterative process at any step $0 \leq c \leq m-1$ to create what we define as **arrested MOD graphs**, or **aMOD**, in short.

We introduce notation to describe arrested MOD graph of order $n = 2^m$ and halted at c^{th} iteration as $aMOD(m, c)$, for example, if $m = 10$ and $c = 6$, we write: $aMOD(10, 6)$ and it describes the arrested MOD graph of order $n = 2^{10} = 1024$ and halted at the 6^{th} iteration resulting in all-to-all fully connected blocks of $2^{m-c} = 16$ vertices.

This process leads to creating 2^c complete sub-graphs of order 2^{m-c}, or in the language of graph theory 2^c of 2^{m-c}-*vertex cliques*.

Why is this important? It turns out that the most efficient supercomputer interconnect topology (currently) on the market is the dragonfly topology [13, 19, 21], exemplified by the Cray XC30 Aries topology[5]. It is complete at back-plane level, or at what we would attribute in the aMOD context (in our aMOD topology) to $2^{m-c} = 16$. Please note that for the sake of simplification we consider Aries NIC as the end-node (vertex) in a dragonfly topology. This is partially justifiable since all connections between Aries and CPUs run as printed circuit lines, and are not realised as copper or optic links. The adjacency matrices of the $aMOD(10, 6)$ graph and the dragonfly graph of order $n = 1152$ are presented below.

[5]see Table 2



a) Adjacency matrix of $aMOD(10,6)$ graph, and

b) .. the dragonfly graph of order $n = 1152$.

Figure 11. Comparison of adjacency matrices of aMOD and dragonfly

N.B. The plot of dragonfly contains the whole set of separate connections (dots on our representation of the adjacency matrix) which are too tiny to be resolved in this picture. In contrast, aMOD has connections which run in off-diagonal arrangement and are more clearly visible. Further discussion of similarities and differences is presented in Section 5.

Figure 10 below illustrates how the MOD algorithm for the MOD graph of order $n = 1024$ or $m = 10$, which is halted at every iteration $c = 0, ..., 9$

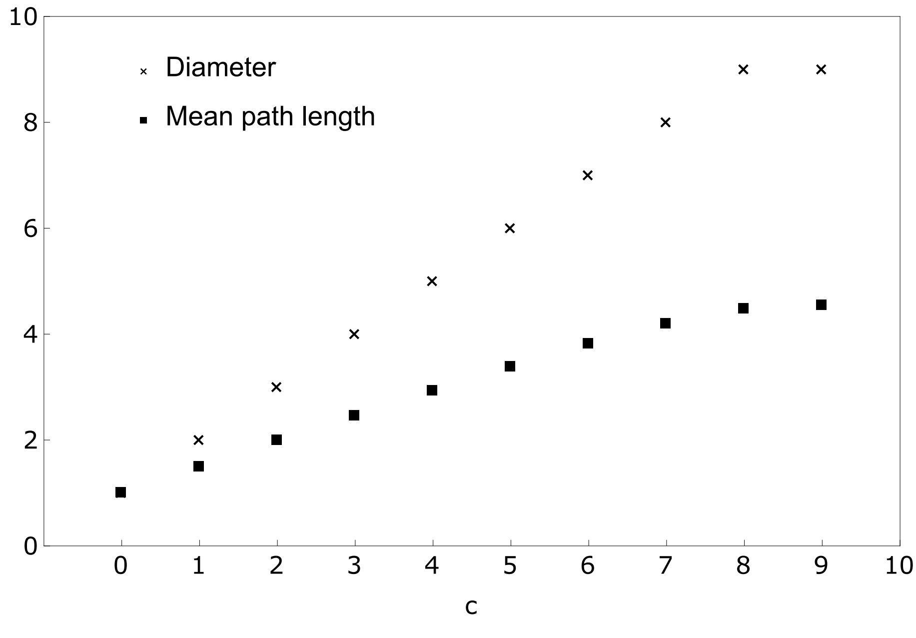


Figure 12. Diameter and mean path length of the aMOD(m,c) graph vs. c; m=10, c-iteration step

The arrested algorithm results in one complete graph ($c = 0$), seven aMOD graphs for $c = 1, 2, ..., 8$, and a saturated MOD algorithm, MOD graph for $c = 9$.
Please note that the adjacency matrices and aMOD graphs for $m = 4$ and $c = 0, 1, 2, 3$ were represented in Figure 2.

The mean path length is about half of the graph diameter, and what is clearly seen, the interesting property of the arrested aMOD graphs - the diameter of the aMOD(m,c) graph is:

$$d_{aMOD(m,c)} = c + 1 \quad 1 \leq c \leq m - 2 \tag{18}$$

$$d_{aMOD(m,c)} = c \quad c = m - 1 \tag{19}$$

The topologies of aMOD graphs could be utilised in the future highly connected and very dense systems, where lot more nodes could be tightly interconnected in all-to-all manner. There is, of course, a price to pay for all-to-all communication: the number of links increases and this adds substantial costs to the entire system cost. However, we believe that the future, ultra-dense systems with on-chip interconnects will allow this type of topology to be implemented. Figure 11 below illustrates the exponential decrease of the size of the aMOD graph as a function of iterations (translated to fully connected blocks (*cliques*) of sub-systems).

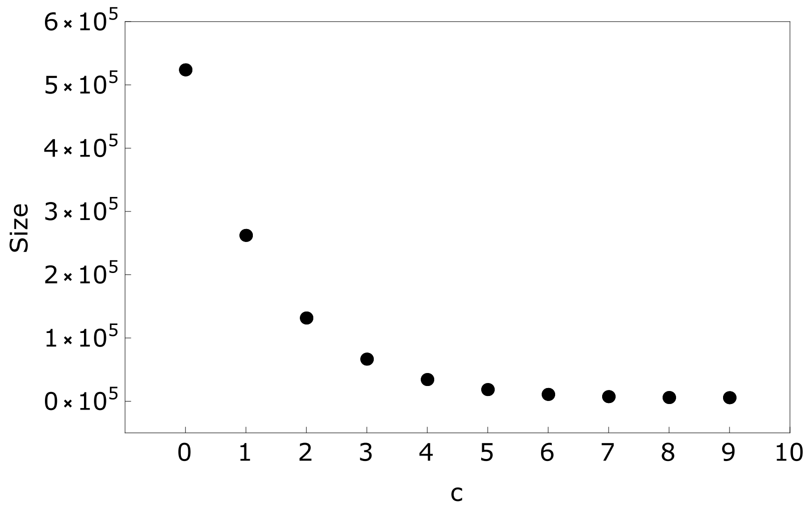


Figure 13. Size of the aMOD(m,c) graph vs. c; m=10, c is the iteration step

4. How all this fits to real Supercomputer interconnect topologies?

The most powerful supercomputer in the world in June 2015 [2], Tianhe-2 has more than 3,000,000 cores. It is well recognised that with limits imposed on increases of clock speed of the new, more powerful systems, the most obvious way to greater computational performance is by increase of parallelism, or in other words, the number of separate computational units: cores in processors or accelerators and a number of nodes in a system. This prospect immediately leads to incredible increase of complexity of possible interconnect networks. Again, the importance of communication, data placement (to minimise communication) and the requirement for robust and well performing network topologies has been well documented [9, 10, 23].

Table 2. Comparison of basic graph properties of MOD and aMOD graphs with graph topologies used in the world's leading supercomputer systems and for recently proposed SlimFly [14] for three different graph orders.
*Note that MOD and aMOD graphs have 2 vertices, whose degree is lower by 1 than that of the remaining vertices - see (6).
**Dragonfly topology as implemented in Cascade has 20 edges per Aries within an electrical group (a two-cabinet arrangement) plus at least one link connecting to a different electrical group

Topology	Graph Order/ Dimensions	Order	Size	Deg.	Diam.	Mean path length	Bisec.
MOD	**1,024**	**1,024**	**5,631**	**11***	**9**	**4.55**	**513**
aMOD	**1,024;16**	**1,024**	**10,815**	**22***	**7**	**3.82**	**513**
Dragonfly	1,152	1,152	11,586	≥21**	5	4.31	12
SlimFly	q = 23	1,058	18,515	35	2	1.97	6,095
3D torus	$8 \times 8 \times 16$	1,024	3,072	6	16	8.01	128
5D torus	$4 \times 4 \times 4 \times 4 \times 4$	1,024	5,120	10	10	5.00	512
Hypercube	10D	1,024	5,120	10	10	5.00	512
TOFU	$3 \times 4 \times 8 \times 2 \times 3 \times 2$	1,152	5,760	10	10	5.38	288
MOD	**2,048**	**2,048**	**12,287**	**12***	**10**	**5.03**	**1,025**
aMOD	**2,048;16**	**2,048**	**22,655**	**23***	**8**	**4.29**	**1,025**
aMOD	**2,048;32**	**2,048**	**37,951**	**38***	**7**	**3.86**	**1,025**
Dragonfly	2,400	2,400	24,300	≥21**	5	4.43	156
SlimFly	q = 31	1,922	45,167	47	2	1.98	14,927
3D torus	$8 \times 16 \times 16$	2,048	6,144	6	20	10.00	256
5D torus	$4 \times 4 \times 4 \times 4 \times 8$	2,048	10,240	10	12	6.00	512
Hypercube	11D	2,048	11,264	10	11	5.50	1,024
TOFU	$5 \times 5 \times 8 \times 2 \times 3 \times 2$	2,400	12,000	10	11	6.07	600
MOD	**4,096**	**4,096**	**26,623**	**13***	**11**	**5.52**	**2,049**
aMOD	**4,096;16**	**4,096**	**47,359**	**24***	**9**	**4.78**	**2,049**
aMOD	**4,096;32**	**4,096**	**77,951**	**39***	**8**	**4.34**	**2,049**
aMOD	**4,096;128**	**4,096**	**270,367**	**133***	**6**	**3.41**	**2,049**
Dragonfly	4,032	4,032	41,181	≥21**	5	4.48	441
SlimFly	q = 37	2,738	75,295	55	2	1.98	25,382
3D torus	$16 \times 16 \times 16$	4,096	12,288	6	24	12.00	512
5D torus	$4 \times 4 \times 4 \times 8 \times 8$	4,096	20,480	10	14	7.00	1,024
Hypercube	12D	4,096	24,576	10	12	6.00	2,048
TOFU	$6 \times 7 \times 8 \times 2 \times 3 \times 2$	4,032	20,160	10	13	6.88	1008

In Table 2 we compare our MOD and aMOD graph topologies with the most common topologies implemented in the largest and most powerful supercomputers in the world: 3D torus exemplified by Gemini interconnect in Cray XE6 and XK7 [3, 4], 5D torus implemented in IBM BlueGene/Q [5], hypercube partially implemented in SGI Pleiades Ice X at NASA [6, 16]; and Fujitsu K-computer with TOFU proprietary interconnect [12]. Dragonfly network is exemplified by Cascade/Aries technology implemented in Cray XC 30 [19]. Currently the largest Cray XC 30 in the world is Piz Daint at Swiss National Supercomputer Center [7]. Its theoretical peak performance R_{peak} is close to 7.8 PFLOPS and actual peak performance R_{max} is 6.2 PFLOPS. Piz Daint has 5,272 nodes, but each node has one Aries NIC and there are four nodes per Aries. Hence there are 1,318 Aries switches, or in our representation of dragonfly topology, 1,318 vertices in the graph representing Piz Daint supercomputer. It means that a graph representing the 6^{th} most powerful supercomputer in the world according to June 2015 Top500 list [2] has the mean path length ~ 4.3.

We have included arrested aMOD(10,6) graph which is the most similar to dragonfly in terms of clique size and graph size. It is interesting to note that our aMOD(10,6) and aMOD(11,7) graphs have less links (smaller size) than the corresponding dragonfly graphs and also slightly better mean path length. This should perhaps translate to slightly better computational performance, if aMOD topologies were tested, subject to network bandwidth and efficiency of routing algorithms comparable to Cascade systems. This superiority is regained by dragonfly graph topology for a network of order 4,096 or higher. Note, that the the two largest Cray XC series systems ever built: Piz Daint at CSCS and Shaheen II at KAUST [8] have about 1,536 Aries routers, which are treated in our analysis as the vertices of the Cray's implementation of the dragonfly topology interconnect. Hence the largest currently implemented dragonfly topology with 1,536 vertices are well within the orders of graphs analysed and presented in Table 2.

For comparison we have also included SlimFly [14] topology which has a very small diameter of 2 and attracted a lot of attention lately. From our analysis, it is clear that SlimFly does not scale well since its size is based on prime numbers. There are no HPC systems built with interconnect based on SlimFly topology yet. Of course SlimFly significantly outperforms both MOD and aMOD graphs with respect to diameter and minimum bisection, but this comes at the price of very high degree and size (see Table 3).

The second very important observation is that all other interconnect topologies perform rather poorly if we compare them on just two scores: diameter and mean path length for the same or very similar order. On the other hand, they do have substantially smaller size (less edges) - which normally translates to cheaper system. However - that is not a fair comparison with dragonfly topology of Cray XC 30 - as we noted previously we do not count all the cores of the four nodes per Aries - the more adequate comparison between various supercomputers would need to include exact core count, routing technology and total bisectional bandwidth.

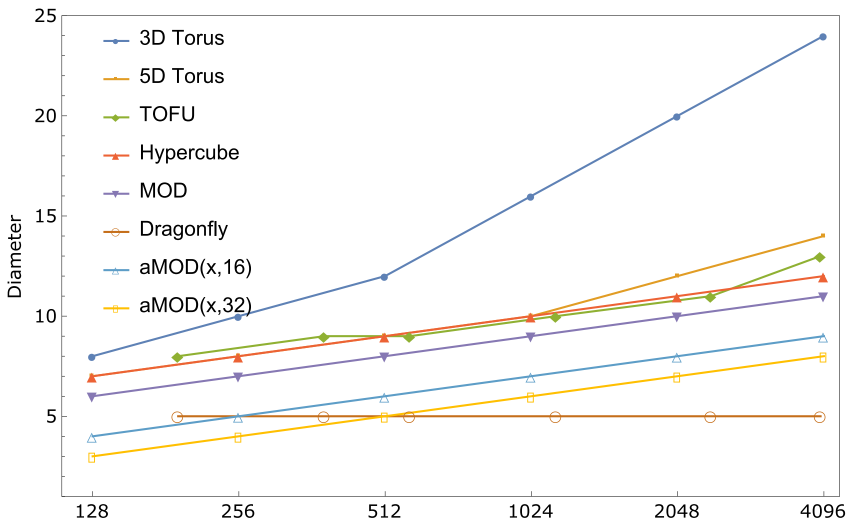


Figure 14. Diameters of MOD graphs and of graph topologies used in the world's leading supercomputer systems as function of graph order

Figure 13 above illustrates the diameter of a graph as a function of its order. This is an important feature in practical interconnect design considerations, since a graph diameter is the upper bound for all distances in a graph. The smaller the diameter, the better, especially for the systems designed for big data, where a program has no control over the placement and locality of data. In 3D torus the diameter grows quite fast and we would expect this topology to be good for a special class of well structured problems with good data locality and very close communication among nearest neighbours, such as regular 3D mesh problems. The diameters of 5D torus and hypercube overlap in the plot since they are the same for graph orders up to 1,024. For larger orders 5D torus diameter is larger than that of hypercube. TOFU performs slightly better than 5D torus and hypercube only for graphs of orders between 512 and 1,024, outside of this range it has larger diameter than both 5D torus and hypercube. Dragonfly has a constant diameter of 5, which is an important property of that graph topology. MOD graphs exhibit moderate growth of diameter as a function of the graph order. MOD graphs' diameters are uniformly better (by 1) than the diameters of hypercube (see Formula 4).

Figure 14 below shows mean path lengths as functions of the order of graphs. Mean path length of 3D torus, 5D torus and hypercube is always half of their diameter, which follows from the periodic boundary conditions of torus (hypercube is a special case of a torus), therefore a rate of growth of mean path length for those three topologies is exactly the same as that of their diameters. Similar behaviour can be observed for TOFU, however its mean path length is slightly more a half of its diameter. MOD graphs have lower diameter than all other graph topologies considered here, bar dragonfly, and the mean path length is expressed by Eq. 3 (in the limit of large graph order). Dragonfly shows a different behaviour. As the order of Dragonfly grows, so does the mean path length, approaching the constant diameter of 5 asymptotically (see Figure 14 and Table 1).

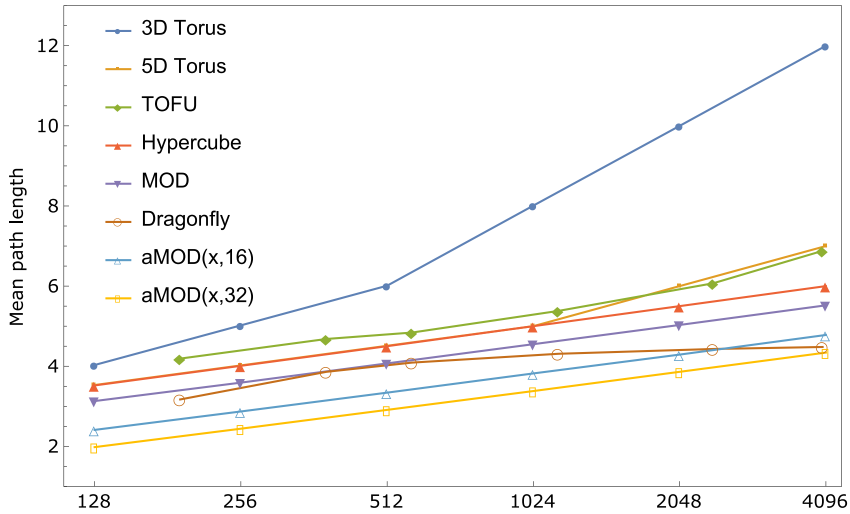


Figure 15. Mean path lengths of MOD graphs and graph topologies used in the world's leading supercomputer systems as function of graph order

The sizes of graphs representing typical supercomputer interconnects are shown in Figure 14. Sizes of graphs grow exponentially, as their orders grow. For dragonfly, the lowest diameter and mean path length is achieved with additional cost of rapid growth of size. MOD and hypercube have similar rate of growth of their size. MOD graphs have more edges though. 5D torus and TOFU both have similar size as the order grows, since they have the same constant vertex degrees. Finally 3D torus whose diameter grows most rapidly, has the slowest rate of size growth.

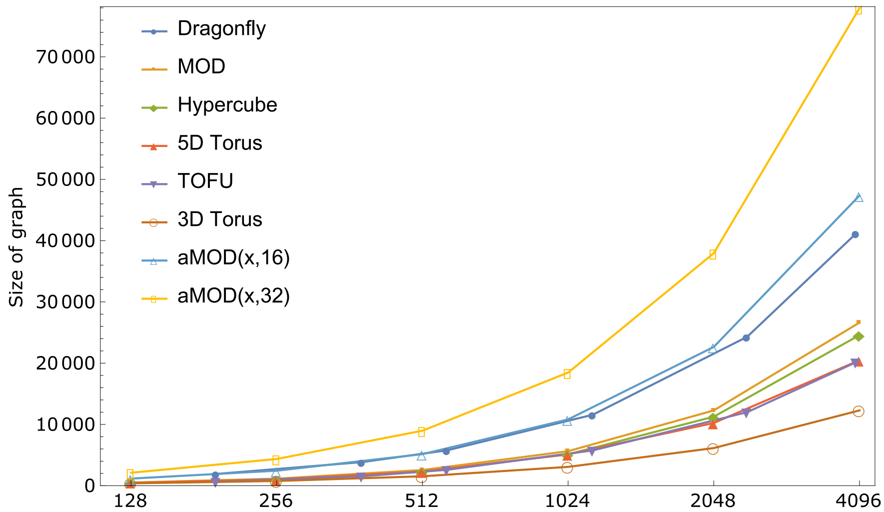


Figure 16. Sizes of MOD graphs and graph topologies used in the world's leading supercomputer systems as function of graph order

Part II

In Part I we presented MOD and arrested MOD graphs resulting from removal of square blocks of edges at each iteration and substituting them with a diagonal matrix with one pivotal element along the main diagonal. Here, we follow similar edge removal strategy, but instead of square shapes we remove triangular shapes from the adjacency matrix, which leads to the final matrix which has two Sierpiński gaskets aligned along the main diagonal. It will be shown below that a graph belonging to this new class of graphs *is not* a Sierpiński graph, since it is the adjacency matrix which has a structure of a Sierpiński gasket, and not a graph described by this matrix. We call this new class of graphs Sierpiński-Michalewicz-Orłowski-Deng (SMOD) graphs.

5. Sierpiński-Michalewicz-Orłowski-Deng (SMOD) Algorithm

Again, as in Section 1, we consider undirected, unweighted graphs. This time, for simplicity of algorithm construction, we restrict the order of SMOD graphs to odd numbers expressed as $|N| = n = 2^m + 1, m \in \mathbb{N}_+$.

The SMOD algorithm is also based on edge removal. The algorithm starts with a complete graph and edges are removed with every iteration, leaving $3^{\log_2(n-1)} = 3^m$ edges upon finishing. The algorithm finishes in $m - 1$ steps.

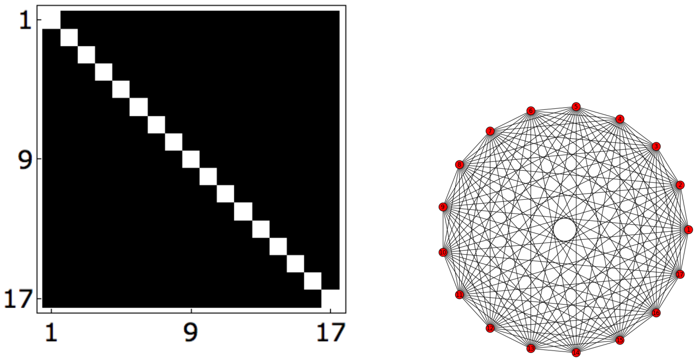
5.1. The SMOD algorithm

The SMOD algorithm has the following steps:

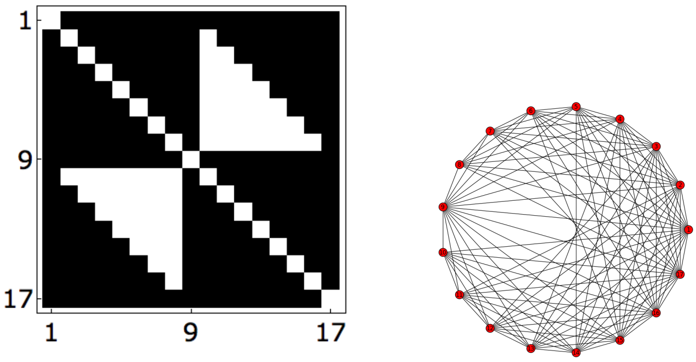
1. Start with an adjacency matrix of a complete graph of order $n + 1 = 2^m + 1$ for $m \geq 2$.
2. The diagonal divides the matrix into two triangular sections: upper-right triangular one (UR) and, lower-left triangular one (LL). We follow exactly Sierpiński iterative process within each of the triangles, with the following qualification: the triangular shapes (matrices) are not equilateral but isosceles right angled triangles.
3. At p^{th} step we subtract $2 \cdot 3^{p-1}$ triangular shapes.
4. Repeat steps 2 and 3 for remaining regions.
5. Stop when triangles reduce to single entry.

5.2. Visualisation of steps of the SMOD algorithm

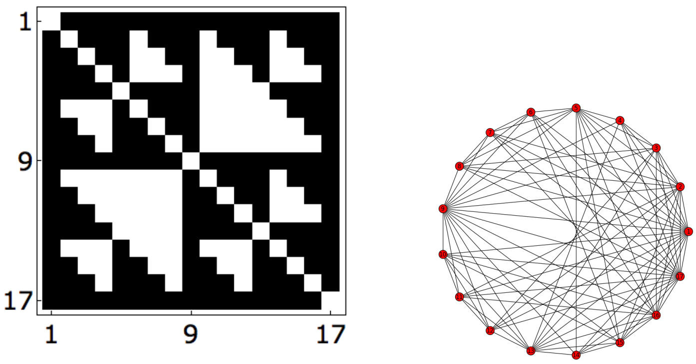
The figure below illustrates adjacency matrices generated in every step of the SMOD algorithm and the diagrams of graphs with a group of edges removed at every step.



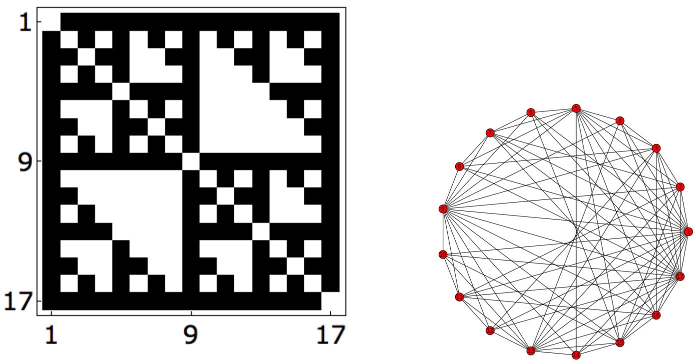
a) Initialization



b) Iteration 1



c) Iteration 2

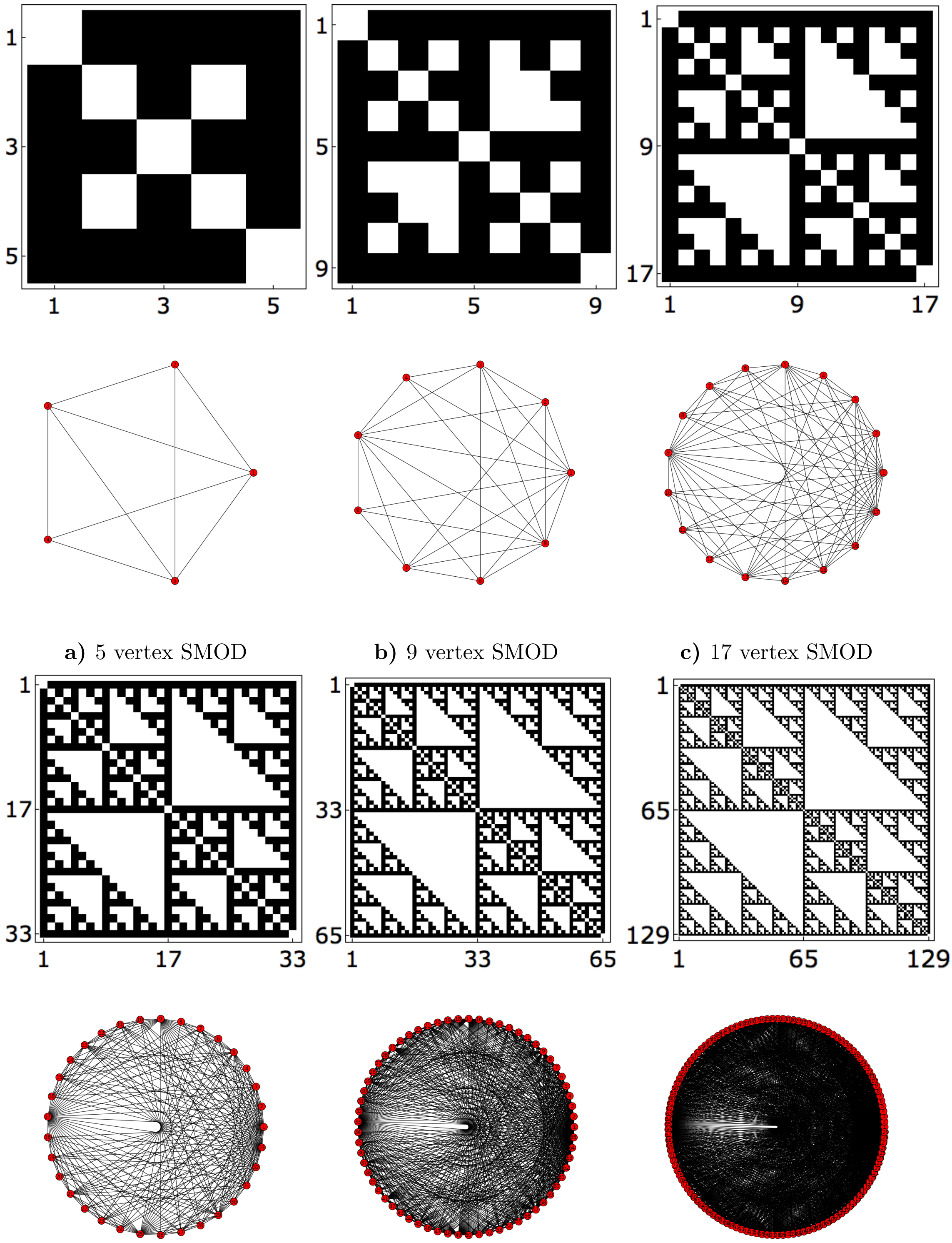


d) Final iteration 3

Figure 17. Steps of SMOD algorithm generating a graph with 17 vertices

5.3. Visualisations of SMOD graphs of small size

Figure 16 illustrates adjacency matrices generated by the SMOD algorithm and corresponding graphs representation for SMOD graphs with $2 \leq m \leq 7$.



a) 5 vertex SMOD **b)** 9 vertex SMOD **c)** 17 vertex SMOD

d) 33 vertex SMOD **e)** 65 vertex SMOD **f)** 129 vertex SMOD

Figure 18. SMOD graphs with $2 \leq m \leq 7$

6. Properties of SMOD graphs

This section outlines basic properties of SMOD graphs such as the relationship between order and the size of SMOD graphs, their diameter and mean path length as well as distance and vertex degree distributions. Most of these properties can be easily expressed analytically. We also look into the spectra of SMOD graphs and attempt to understand what information can be extracted from them.

6.1. Size of SMOD graphs

Theorem 2. If $n = 2^m + 1$ is the order of a SMOD graph, then its size is given by

$$|L| = 3^m = 3^{\log_2(n-1)} \tag{20}$$

Proof. Counting the edges removed at each step of the SMOD algorithm one gets the following formula

$$\begin{aligned} |L| &= \frac{1}{2}\left[n(n-1) - \sum_{k=1}^{m-1} 3^{k-1}\left(\frac{n-1}{2^k} - 1\right)\left(\frac{n-1}{2^k}\right)\right] \\ &= \frac{1}{2}\left[(2^m+1)2^m - \sum_{k=1}^{m-1} 3^{k-1}\left(\frac{2^m}{2^k} - 1\right)\left(\frac{2^m}{2^k}\right)\right] \end{aligned} \tag{21}$$

By mathematical induction let's show that

$$\frac{1}{2}\left[(2^m+1)2^m - \sum_{k=1}^{m-1} 3^{k-1}\left(\frac{2^m}{2^k} - 1\right)\left(\frac{2^m}{2^k}\right)\right] = 3^m \tag{22}$$

The simplest SMOD graph has order $n = 5$, which corresponds to $m = 2$. The formula holds for such m as

$$\frac{1}{2}\big[(2^2+1)2^2 - 2\big] = \frac{1}{2}(20-2) = 9 = 3^2 \tag{23}$$

Let's assume that the formula holds for some $m > 2$. We shall show that it also holds for $m+1$. Before we proceed, let's note that

$$\begin{aligned} &(2^{m+1}+1)2^{m+1} = 2\big[1+2+3+\cdots+2^{m+1}\big] \\ &= 2\big[1+2+3+\cdots+2^m+(2^m+1)+\cdots+(2^m+2^m)\big] \\ &= 2\big[\sum_{k=1}^{2^m} k + 2^m 2^m + \sum_{k=1}^{2^m} k\big] = 2\big[2\sum_{k=1}^{2^m} k + 4^m\big] \\ &= 2(2^m+1)2^m + 2\cdot 4^m \end{aligned} \tag{24}$$

and that

$$\begin{aligned} &\left(\frac{2^{m+1}}{2^k} - 1\right)\left(\frac{2^{m+1}}{2^k}\right) = \big(2^{m-k+1}-1\big)2^{m-k+1} \\ &= 2\big[1+2+3+\cdots+(2^{m-k}-1)+2^{m-k}+(2^{m-k}+1)+\cdots+(2^{m-k}+2^{m-k}-1)\big] \\ &= 2\big[\sum_{k=1}^{2^{m-k}-1} k + 2^{m-k}2^{m-k} + \sum_{k=1}^{2^{m-k}-1} k\big] = 2(2^{m-k}-1)2^{m-k} + 2\cdot 4^{m-k} \end{aligned} \tag{25}$$

Applying the inductive step and using (23) and (24) we get

$$
\begin{aligned}
&\frac{1}{2}\left[(2^{m+1}+1)2^{m+1} - \sum_{k=1}^{m} 3^{k-1}\left(\frac{2^{m+1}}{2^k}-1\right)\left(\frac{2^{m+1}}{2^k}\right)\right] \\
&= \frac{1}{2}\left[2(2^m+1)2^m + 2\cdot 4^m - 2\sum_{k=1}^{m} 3^{k-1}[(2^{m-k}-1)2^{m-k}+4^{m-k}]\right] \\
&= (2^m+1)2^m - \sum_{k=1}^{m-1} 3^{k-1}(2^{m-k}-1)2^{m-k} + 4^m - \frac{4^m}{3}\sum_{k=1}^{m-1}\left(\frac{3}{4}\right)^k \\
&= 2\cdot 3^m + 4^m - \frac{4^m}{3}\sum_{k=1}^{m-1}\left(\frac{3}{4}\right)^k = 3^{m+1} - 3^m + 4^m - \frac{4^m}{3}\sum_{k=1}^{m-1}\left(\frac{3}{4}\right)^k \\
&= 3^{m+1} - 3^m + 4^m - \frac{4^m}{3}\left(\frac{3/4-(3/4)^{m+1}}{1-3/4}\right) \\
&= 3^{m+1} - 3^m + 4^m - \frac{4^m}{3}\left(3-\frac{3^{m+1}}{4^m}\right) = 3^{m+1} - 3^m + 4^m - 4^m + 3^m \\
&= 3^{m+1}
\end{aligned}
\tag{26}
$$

By the induction assumption the formula holds for $m+1$, hence it holds for any $m \geq 2$. □

The figure below shows how the size of SMOD graphs changes as a function of m, where $n = 2^m + 1$ as n grows from 5 to 1,025 (or $2 \leq m \leq 10$)

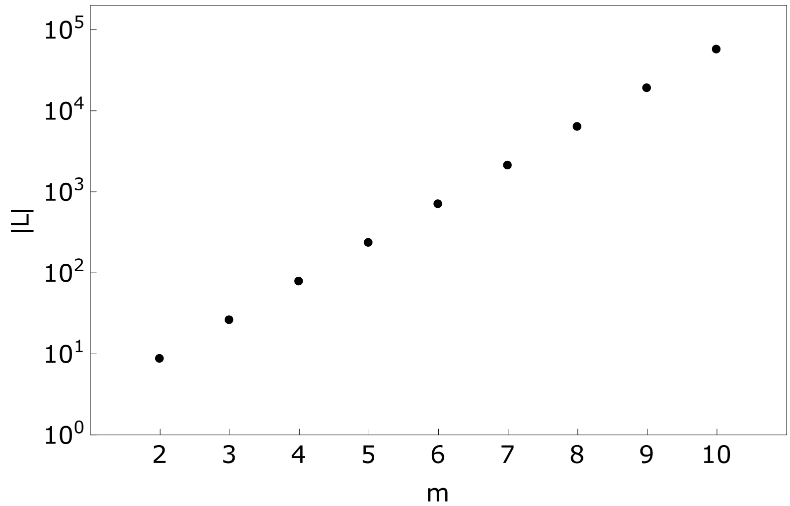


Figure 19. Sizes of SMOD graph with $2 \leq m \leq 10$

6.2. Diameter, mean path length and distances

Every SMOD graph of order n contains three S_n (n-star) subgraphs. Those three subgraphs are seen in the SMOD graphs' adjacency matrices as fully filled (except of the diagonal element) rows (or columns) 1, $\frac{n-1}{2}$ and n. This trivially corresponds to vertices that are connected to all other vertices. The S_n subgraphs are the minimum spanning trees of SMOD graphs. Since their diameter is 2 this implies that every SMOD graph has diameter 2.

An interesting conclusion follows from the above observation:

Theorem 3. There are no mutually orthogonal vectors in the column (or row) space of an adjacency matrix of a SMOD graph.

Proof. An SMOD graph of order n has exactly three S_n n-star subgraphs, which are also the minimum spanning trees. This means that the diameter of an SMOD graph of order n is 2 and thus the distance is at most 2, hence there always exists a walk of length 2 between any two vertices i and j. By Lemma 3 in [25] for any undirected graph the number of walks of length k between vertices i and j is equal $(A^k)_{ij}$, thus, since for all pairs of vertices i and j in SMOD graph there is always a walk corresponding to a distance ≤ 2 hence $(A^2)_{ij} = (A^T A)_{ij} > 0$. Therefore the scalar product of any two columns of an adjacency matrix of an SMOD is non-zero, and thus no pair of vectors from column space is orthogonal. □

The following Figure depicts how fast the mean path length approaches the diameter. This means that as the order of the graph grows, so does the number of pairs of vertices that are 2 hops away with respect those that are directly connected.

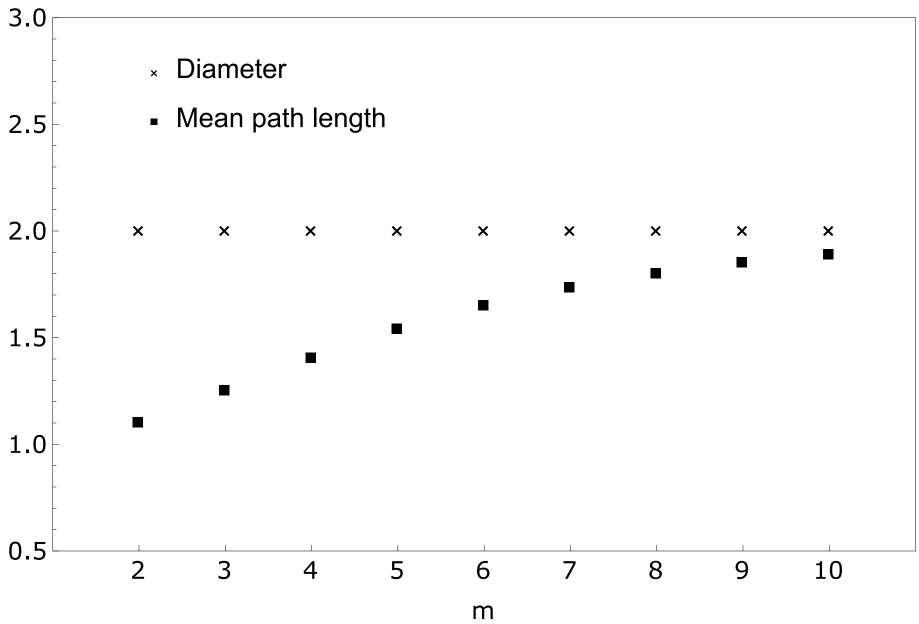


Figure 20. Graph diameters and mean path lengths of SMOD graphs

Since the diameter of any SMOD graph is 2, the distance between any two vertices can be either 1 or 2. This means that if two vertices i and j are not connected, then they are 2 hops away. In terms of the distance matrix, this says that wherever there is an off-diagonal 0 in the adjacency matrix there is 2 in the distance matrix. By this argument we see that the following formula holds for the distance matrix.

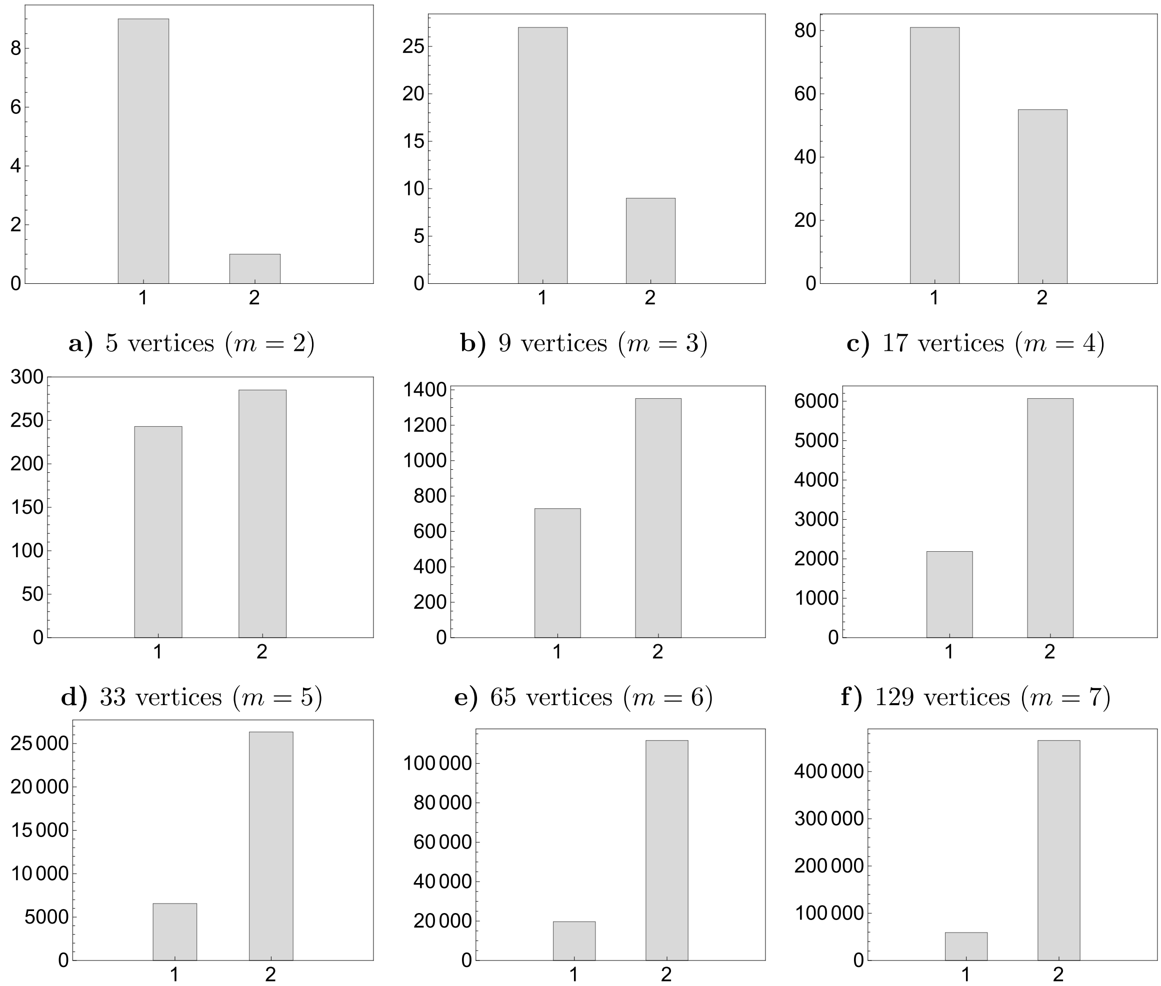
$$D = 2J - 2I - A \tag{27}$$

where J is all-ones matrix.

Using the argument leading to Eq. 26 and Theorem 2 one sees that the number of pairs of vertices, the distance between which is 2 is expressed by the following formula

$$\sum_{i \neq j} d(v_i, v_j)\Big|_2 = \frac{1}{2}(2^m + 1)2^m - 3^m = \frac{1}{2}n(n-1) - 3^{\log_2(n-1)} \tag{28}$$

The following figure shows the distance distribution in SMOD graphs.



a) 5 vertices ($m = 2$) **b)** 9 vertices ($m = 3$) **c)** 17 vertices ($m = 4$)

d) 33 vertices ($m = 5$) **e)** 65 vertices ($m = 6$) **f)** 129 vertices ($m = 7$)

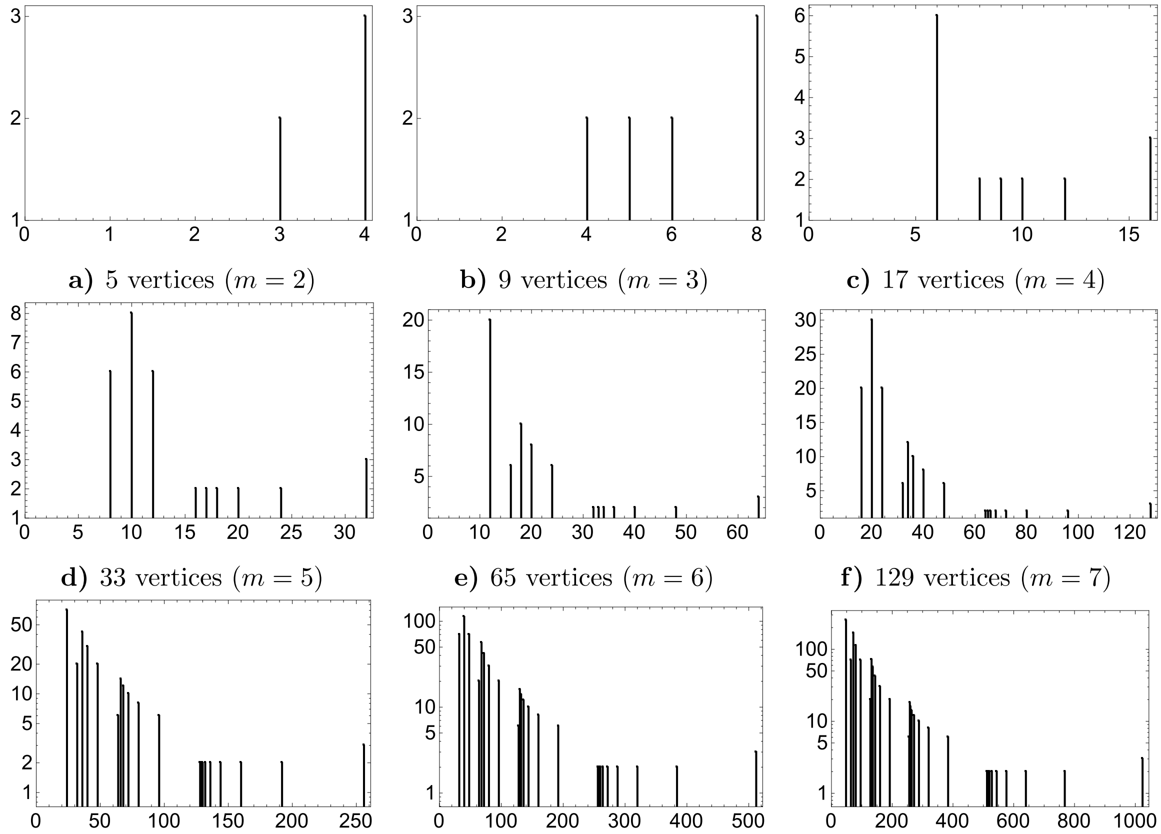
g) 257 vertices ($m = 8$) **h)** 513 ($m = 9$) **i)** 1,025 vertices ($m = 10$)

Figure 21. Distribution of path lengths in SMOD graphs with $2 \leq m \leq 10$

The figure depicts what can also be seen from Eq. 27: for $m \geq 5$ the pairs of vertices 2 hops away outnumber directly connected pairs.

6.3. Vertex degrees and their distribution

In general, a low diameter of a graph implies existence of vertices of high degree. This is the case with SMOD graphs, which have diameter 2 regardless of the graph order, but at the same time three vertices have degree $n - 1 = 2^m$ and large number of vertices are of relatively high degree.



a) 5 vertices ($m = 2$) **b)** 9 vertices ($m = 3$) **c)** 17 vertices ($m = 4$)

d) 33 vertices ($m = 5$) **e)** 65 vertices ($m = 6$) **f)** 129 vertices ($m = 7$)

g) 257 vertices ($m = 8$) **h)** 513 vertices ($m = 9$) **i)** 1,025 vertices ($m = 10$)

Figure 22. Vertex degree *vs.* number of vertices of that degree in SMOD graphs with $2 \leq m \leq 10$

6.4. Eigenvalues of SMOD graphs

Here again, we denote the greatest eigenvalue by λ_1 and the least eigenvalue by λ_n. As with MOD graphs, SMOD graphs have the the greatest eigenvalue equal to the spectral radius, that is $\lambda_1 = \rho(A(G))$.

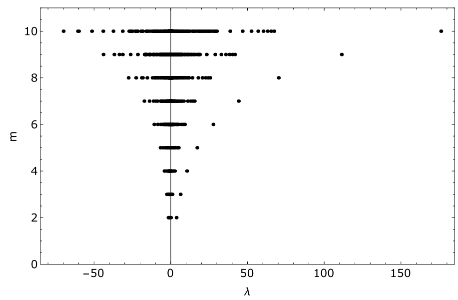


Figure 23. Distribution of eigenvalues for SMOD graphs with $2 \leq m \leq 10$.

Figure 21 above represents the distribution of eigenvalues for SMOD graphs of order $n = 2^m + 1$ for $2 \leq m \leq 10$. There is a noticeable gap between λ_1 and λ_2. Detailed study of exact interpretation of the SMOD graph eigenvalues is outside the scope of this paper.

6.5. Edge ratio of SMOD graphs

Regardless of the order, SMOD graphs have diameter 2. From Eqs. 19 and 27 and the definition of the mean path length of an undirected graph it follows that the mean path length of a SMOD graph approaches 2 for large m. It is also seen in Figure 18. A question we raise here is how does SMOD graph size is compared with complete graph size. First, we compute sizes of all graphs of order $n = 2^m + 1$, for $2 \leq m \leq 10$. The results are depicted in the figure below.

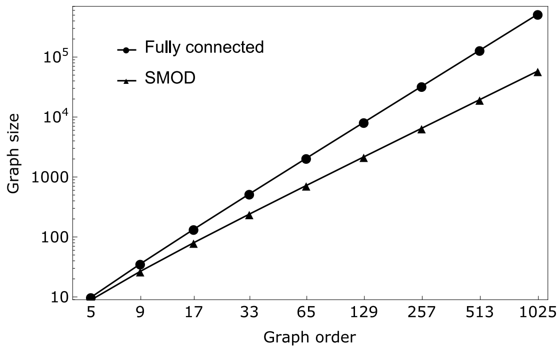


Figure 24. Comparison of graph sizes for SMOD graphs and complete graphs

Figure 23 below illustrates edge ratio of SMOD graphs which we define as a size ratio of a SMOD graph to a complete graph. The tick marks in the figure correspond to consecutive values of m, where $n = 2^m + 1$. For $m = 2$, the ratio is 0.9 and it drops to 0.1 for $m = 10$. The edge ratio converges exponentially to 0 as m grows large.

This is expressed analytically in the following manner

$$\lim_{m\to\infty} L_{ratio} = \lim_{m\to\infty} \frac{3^m}{\frac{1}{2}(2^m+1)2^m} = 0 \tag{29}$$

Table 3 on the following page summarises the order, size, diameter and mean path length of SMOD graphs for $2 \leq m \leq 10$.

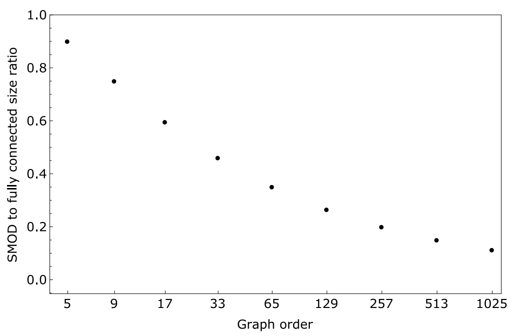


Figure 25. Size ratio Comparison of SMOD and fully connected graphs

Table 3. Comparison of SMOD, SlimFly [14] and the complete graphs. SlimFly is compared wherever it can attain similar order. See Table 2 for comparison of SlimFly with other graphs

Graph	Order	Size	Diameter	Mean path length	Bisection bandwidth
SMOD	**5**	**9**	**2**	**1.10**	5
Complete	5	10	1	1.00	6
SMOD	**9**	**27**	**2**	**1.25**	14
Complete	9	36	1	1.00	20
SMOD	**17**	**81**	**2**	**1.40**	41
SlimFly	18	45	2	1.70	15
Complete	17	136	1	1.00	72
SMOD	**33**	**243**	**2**	**1.54**	130
Complete	33	528	1	1.00	272
SMOD	**65**	**729**	**2**	**1.64**	276
SlimFly	50	539	2	1.86	65
Complete	65	2,080	1	1.00	1,056
SMOD	**129**	**2,187**	**2**	**1.74**	794
Complete	129	8,256	1	1.00	4,158
SMOD	**257**	**6,561**	**2**	**1.80**	2,317
SlimFly	242	2,057	2	1.93	671
Complete	257	32,896	1	1.00	16,500
SMOD	**513**	**19,683**	**2**	**1.85**	7,068
Complete	513	131,328	1	1.00	65,736
SMOD	**1,025**	**59,049**	**2**	**1.89**	20,262
SlimFly	1058	18,515	2	1.97	6,095
Complete	1,025	524,800	1	1.00	262,446

Conclusions

We present a novel method and algorithms for generating new classes of MOD and SMOD graphs. We stress that our primary objective in the present work is to propose fast and effective algorithms for creating sparse adjacency matrices by iterative removal of entire groups of edges. This translates to creating well connected graphs of arbitrarily large order but of reasonably small size and also having other properties that render some of them to be interesting candidates for supercomputer interconnect topologies. However, we do not claim that this process will lead to "optimal" interconnect network topologies - irrespective of the definition of "optimal". The central object for this method is the adjacency matrix, and we derive the graphs, and all their properties from this matrix. The starting graph is always the complete graph. We construct graphs of arbitrary order $n = 2^m$ or $n = 2^m + 1$, for MOD and SMOD graphs, respectively, and gradually, by algorithmic removal of edges, arrive at the final graph. We also propose a halted

MOD algorithm which leads to graphs $aMOD(m,c)$ with cliques of order 2^{m-c}. We compare the main graph properties such as diameter, mean path length, histogram of distances and size to all typical graph topologies represented in the most important supercomputers that are currently on the market. It turns out that our MOD graphs are better than almost all of these supercomputer interconnect topologies, with the exception of dragonfly topology, for graphs of some large orders.

The algorithm was designed to introduce a systematic and simple way of removing as large number of edges as possible at each iterative step, starting from the complete graph, and at the same time to preserve good graph connectivity, and arrive at final graphs of reasonably small size. The method is fast, almost all important graph properties can easily be derived from the combinatorics of the algorithm.

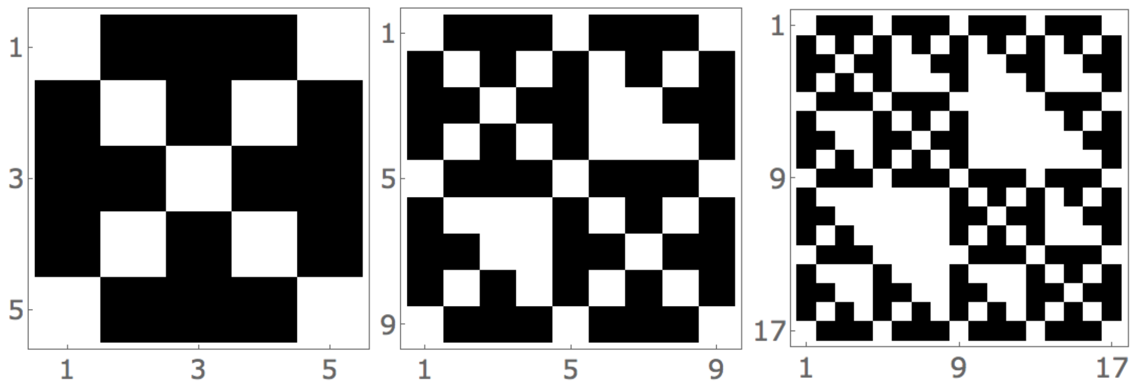
Clearly, besides the two algorithms described here, there may be great number of similar schemes leading to other graphs of similarly interesting, and perhaps even "better", properties.

Here are some examples of the other schemes that have not been investigated here. We plan to investigate these alternatives and report detailed results in the future.

1. A trivial modification of MOD algorithm described in Fig. 1 would be to substitute P_e at the first iteration step with the symmetric matrix $P_s \in \mathbb{R}^{n/2\times n/2}$ defined as

   $$(P_s)_{ij} = \begin{cases} 1 & i = n/2, j = 1 \\ 1 & i = 1, j = n/2 \\ 0 & \text{otherwise} \end{cases}$$

   this will lead to a *regular* graph *i.e.* a graph with all vertices of the same degree. Eq. 6 for such a variant of MOD graph would read: $Au = (m+1, m+1, ..., m+1, m+1)^T$. This graph will have a size bigger by 1 than our MOD graph, *i.e.* $|L| = \frac{n}{2}(log_2 n + 1) = (m+1)2^{m-1}$.
2. Modification of SMOD algorithm to remove 3^{p-1} extra edges - "the centres of star graphs" at each p^{th} iteration. This is illustrated below in Fig. 24 (a)-(c) and should be compared with Fig. 16 (a)-(c). Three most important implications of such a new augmented algorithm are: i) this will also lead to graphs of diameter 2, ii) the graphs will have smaller size, iii) all vertices with of the highest order will be removed (see Fig. 20 (a)-(i)). All these characteristics make such modified SMOD graphs better candidates for interconnects in Big Data era (up to some reasonable size $n = 2^m + 1$).



a) 5 vertex modified SMOD **b)** 9 vertex modified SMOD **c)** 17 vertex modified SMOD

Figure 26. Adjacency matrices of modified SMOD graphs with lowered maximum vertex degrees

3. Combination of MOD and SMOD algorithms. This process will start with several steps performed with one of the algorithms, and than switch to the other one. Clearly this will lead to graphs with "off-diagonal" parts further away from the main diagonal akin to the first graph, with all blocks adjacent to the main diagonal having a character of the second graph family. It is also clear that such combined graphs are also *embedded* graphs of one in to the other graph class.

A cursory glance at Figs. 3, 16 and 24 leads us to an immediate conclusion that each MOD, SMOD or modified SMOD graph is composed of *self-similar* graphs of all smaller orders. Again, this can be cast in the statement that these graph families are *embeddings* of graphs of smaller order. The most remarkable property of the SMOD graphs is that irrespective of the graph order, the diameter is constant and equals 2. However, the size of the graph, or the total number of edges, is about 10% of the size of a complete graph of the same order and there are vertices of rather high order. These characteristics may prove useful in considering SMOD graphs as possible candidates for interconnect topologies for Big Data computer architectures, provided the vertex order does not exceed technological capabilities, *i.e.* the number of ports in a typical switch. One may however speculate that if such topology is implemented as printed circuit paths or in optical switch - the high radix may not be a serious obstacle anymore, considering substantial reduction in the number of required links. It would be also interesting to investigate further how well MOD and SMOD graphs topologies match communication patterns in typical computational algorithms. We plan to devote a separate study to this open problem.

References

1. http://www.infoworld.com/d/computer-hardware/china-building-100-petaflop-supercomputer-206072. Last accessed: 2014.08.05.
2. http://www.top500.org/lists/2014/06/. Last accessed: 2015.10.05.
3. http://www.cray.com/Products/Computing/XK7/Technology.aspx. Last accessed: 2014.08.05.
4. https://bluewaters.ncsa.illinois.edu/user-guide. Last accessed: 2014.08.05.
5. https://computing.llnl.gov/tutorials/bgq/#Networks. Last accessed: 2014.08.05.
6. http://www.nas.nasa.gov/hecc/resources/pleiades.html. Last accessed: 2014.08.05.
7. http://www.cscs.ch/computers/pizdaint/index.html. Last accessed: 2014.08.05.
8. http://www.top500.org/system/178515. Last accessed: 2015.10.05.
9. Report on the workshop on extreme-scale solvers: Transitions to future architectures. Technical report, Office of Advanced Scientific Computing Research, U.S. Department of Energy, March 2012. http://science.energy.gov/ascr/news-and-resources/program-documents/, Last accessed: 2014.08.05.
10. On advanced scientific computing research exascale mathematics working group (emwg) report for 2014. Technical report, The Department of Energy (DOE) Office of Science Pro-

gram, 2014. `https://collab.mcs.anl.gov/display/examath/Exascale+Mathematics+Home`, Last accessed: 2014.08.05.

11. N. R. Adiga et al. BlueGene/L torus interconnection network. *IBM Journal of Research and Development*, 45(2.3):265–276, 2005.
12. Yuichiro Ajima, Shinji Sumimoto, and Toshiyuki Shimizu. Tofu: A 6d mesh/torus interconnect for exascale computers. *Computer*, 11(42):36–40, 2009.
13. Richard F Barrett, Courtenay T Vaughan, Simon D Hammond, and D Roweth. Application explorations for future interconnects. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*, pages 1717–1724. IEEE, 2013.
14. Maciej Besta and Torsten Hoefler. Slim fly: a cost effective low-diameter network topology. In *High Performance Computing, Networking, Storage and Analysis, SC14: International Conference for*, pages 348–359. IEEE, 2014.
15. RB Borie, R Gary Parker, and CA Tovey. Recursively constructed graphs. *Handbook of Graph Theory*, pages 99–125, 2004.
16. J. P. Brown. Hypercubes. *Practical Computing, Vol. 5 No. 4*, pages 97–99, 1982.
17. Dragoš M Cvetković, Peter Rowlinson, and Slobodan Simic. *Eigenspaces of graphs*. Number 66. Cambridge University Press, 1997.
18. William James Dally and Brian Patrick Towles. *Principles and practices of interconnection networks*. Elsevier, 2004.
19. Greg Faanes, Abdulla Bataineh, Duncan Roweth, Edwin Froese, Bob Alverson, Tim Johnson, Joe Kopnick, Mike Higgins, James Reinhard, et al. Cray cascade: a scalable hpc system based on a dragonfly network. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, page 103. IEEE Computer Society Press, 2012.
20. M. Kan. China is building a 100-petaflop supercomputer. *IDG News Service, infoworld.com (2012-10-31)*.
21. John Kim, Wiliam J Dally, Steve Scott, and Dennis Abts. Technology-driven, highly-scalable dragonfly topology. In *ACM SIGARCH Computer Architecture News*, volume 36, pages 77–88. IEEE Computer Society, 2008.
22. F Thomson Leighton. *Introduction to parallel algorithms and architectures: Arrays· trees· hypercubes*. Elsevier, 2014.
23. Matt Knepley Mark F. Adams, Jed Brown. Low-communication techniques for extreme-scale multilevel solvers. `https://collab.mcs.anl.gov/display/examath/Submitted+Papers`. Last accessed: 2014.08.05.
24. Richard Otter. The number of trees. *Annals of Mathematics*, pages 583–599, 1948.
25. Piet Van Mieghem. *Graph spectra for complex networks*. Cambridge University Press, 2010.