big data and cognitive computing

MDPI

Article

The Dataset for Optimal Circulant Topologies

Aleksandr Romanov

HSE University, Moscow 101000, Russia; a.romanov@hse.ru; Tel.: +7-495-772-95-90 (ext. 15140)

Abstract: This article presents software for the synthesis of circulant graphs and the dataset obtained. An algorithm and new methods, which increase the speed of finding optimal circulant topologies, are proposed. The results obtained confirm an increase in performance and a decrease in memory consumption compared to the previous implementation of the circulant topologies synthesis method. The developed software is designed to generate circulant topologies for the construction of networks-on-chip (NoCs) and multi-core systems reaching thousands of computing nodes. The developed software makes it possible to achieve high performance on an ordinary research workstation commensurate with similar solutions created for a supercomputer. The use cases of application of the created software for the analysis of routing algorithms in circulants and the regression analysis of the generated dataset of graph signatures to predict the characteristics of graphs of any size are described.

Keywords: circulant graph; circulant topologies dataset; routing algorithm; network-on-chip; supercomputing

Citation: Romanov, A. The Dataset for Optimal Circulant Topologies. *Big Data Cogn. Comput.* **2023**, *7*, 80. https://doi.org/10.3390/bdcc7020080

Academic Editors: Eric M. Lui, Antonio Concilio and Salvador García López

Received: 15 March 2023
Revised: 13 April 2023
Accepted: 18 April 2023
Published: 20 April 2023

1. Introduction

The development of computing systems suggests making solutions based on multicore processors. In this area, on-chip networks have been used as a replacement of system buses, the previous way of communication between IP cores. The main disadvantage of system buses is their inability to meet the bandwidth requirements of multicore systems-on-chip due to a decrease in operating speed at a high load when increasing the number of cores is needed [1]. This scalability problem is solved by communicating simultaneously over multiple connections between routers, which implement an NoC communication subsystem.

Thus, an important problem in the development of NoCs is the search for optimal topologies of the communication subsystem, since regular topologies, such as mesh, torus [2], and others [3] do not meet modern requirements for NoCs. An attempt to improve the basic characteristics of such topologies leads to large resource costs. Circulant topologies have better characteristics than classical regular topologies and better structural survivability, reliability, and connectivity; they also require fewer inter-processor exchanges when solving computational and system control problems. These features allow using them in networks with a large number of nodes.

A number of works are devoted to the use of circulant topologies for building communication networks. For example, a subset of circulant topologies known as Midimew [4] has been proposed for use in communication networks in high-performance supercomputers. In [5,6], circulants are also considered as a topology for supercomputing. In [7,8], it is argued that this allows an increase in network performance and a reduction in the mean path length (MPL) of packet transmission in comparison with a torus topology. Also, [9] proposed to use circulants in coding theory. Thus, circulants have potentially wide areas of application, and finding families of optimal circulants with a large number of nodes is an important task.

Circulants are regular topologies based on the Cayley graphs of a cyclic group. This type of topologies can be described as $C(N; s_1, \ldots, s_k)$, where N—number of nodes, k—number of generators s_i (graph dimension), and $k < N$ (Figure 1). It is this type

of description of circulant topologies (signature of the circulant) that is used in this article. Circulant topologies are also mentioned in the context of double fixed-step loop networks [10], fixed-step graphs [11], Chordal Rings [12], Mobius ladder [13,14], Midimew [4], low-dimensional regular lattice [15], double-loop networks [16], multi/multiple-loop networks [17,18], etc. The features of circulants are considered in detail in [19,20]. For practical tasks, it is necessary to develop regular descriptions of families of circulants with the desired properties using their signatures calculated by a formula from N. Thus, there are descriptions for two-dimensional optimal circulants [19,21], recursive [22,23] and multiplicative circulants [24], extremal circulants [25], etc. But potentially there are yet undiscovered or not formalized families of circulants that are waiting in the wings.

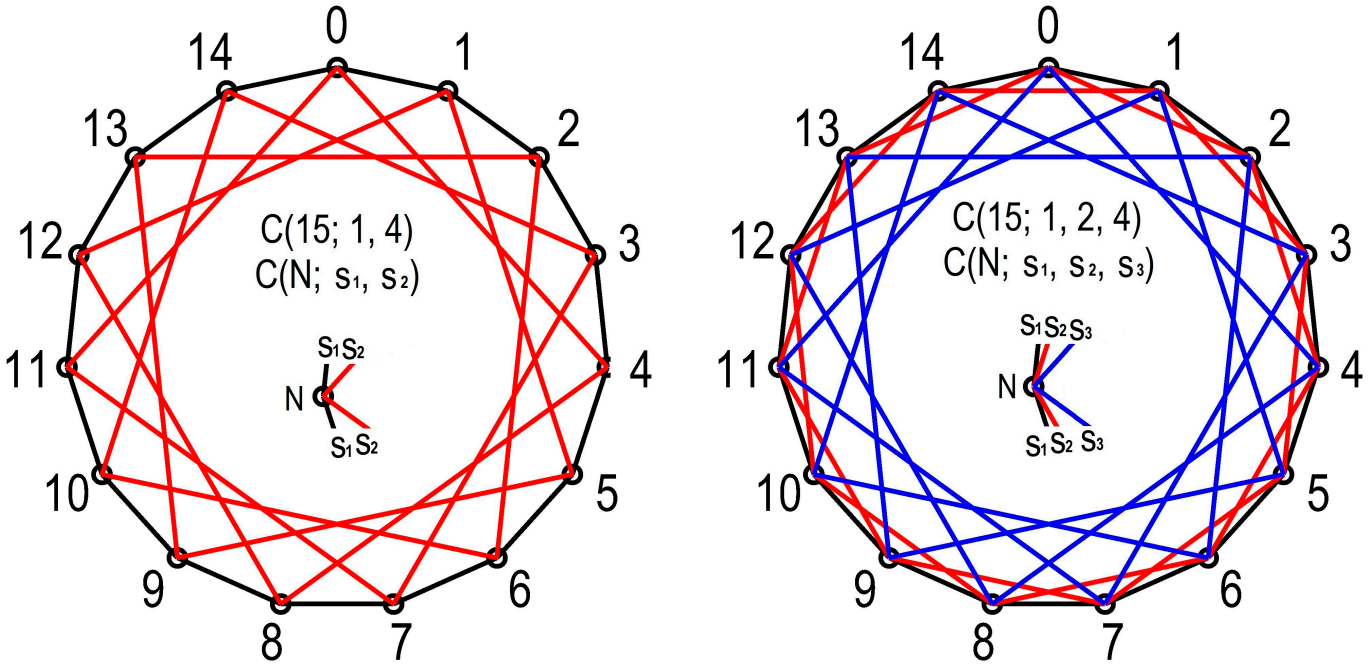

Figure 1. Topologies C(15; 1, 4) (**a**) and C(15; 1, 2, 4) (**b**).

The problem of searching for circulant topologies lies in its high computational complexity (it can be classified as an NP-hard problem), if solved by enumeration methods and breadth-first search methods [5,6,20]. At the same time, the rapid development of the theory of neural networks and machine learning methods [26] makes it possible to assume the prospect of using the signatures of already found topologies together with the methods of mathematical analysis for the problems of studying and predicting the characteristics of higher-order topologies, as well as for developing new families of circulant graphs that have useful properties for their application in practice. This requires large datasets and benchmarks, as well as software tools for their synthesis. This is the main contribution of this article. The example of the analysis of the obtained dataset using regression analysis methods given at the end of the article demonstrates the usefulness of this work and the prospects for applying the results obtained.

A list of key notations is presented in Table 1 for ease of reference.

Table 1. List of key notations.

Notation	Description
NoC	Network-on-Chip
$C(N; s_1, \ldots, s_k)$	Circulant graph signature
N	Number of nodes
k	Circulant graph dimension
$s_i, i = 1, \ldots, k$	Circulant graph generator
MPL	Mean Path Length
GCG	Greedy Circulants Generator
MAE	Mean Absolute Error
PPMCC	Pearson Product-Moment Correlation Coefficient

The remainder of this manuscript is organized as follows. In Section 2, we describe the problem of the synthesis of circulant topologies and its proposed solution. In Section 3, a detailed description of the developed software and its features is presented. Section 4 contains the evaluation of the obtained results, a comparison with concurrent solutions from other researchers, and the obtained dataset description. Section 5 consists of some examples of use cases for the developed dataset and software showing how it speeds up the search of new circulant topologies, how it can be used for routing algorithms analysis, and some formulas by regression analysis on the dataset obtained are also presented. The Section 6 finalizes this paper.

2. The Problem of the Synthesis of Circulant Topologies

The main problem that arises in the synthesis of optimal circulant topologies (as well as for irregular topologies [27,28]) is the search for the optimal topology configuration in accordance with the criteria for achieving a minimum diameter and MPL. The basic procedure, which guarantees the check of all possible variants of topologies, is an exhaustive search method. The basic ideas of the search procedure were first described in [20], but during development, they were deeply reworked and improved. The main steps of the proposed method are the steps of the search procedure implemented in the proposed Greedy Circulants Generator (GCG) software; they are supplemented by other methods and algorithmic solutions to speed up the search. Thus, at the macro level, the work of GCG consists of the following steps:

1. Iterations in which the topologies, being formed and denoted by indices, are performed (Figure 2).
2. For every received signature of the topology at every step of the iteration, the analysis of the obtained topology is carried out (Figure 3).
3. The parameters of the topology are calculated (Figure 4).
4. The topology is evaluated according to the introduced optimality criterion defined by objective function (Figure 4).

The main search procedure is improved by using the features of circulant topologies discussed below. Circulant graphs have a number of unique features, which make it possible to significantly improve and speed up the procedure of finding optimal topologies. The applied acceleration methods could be divided into several categories:

- Reducing the number of variants for enumerating possible parametric descriptions of topologies.
- Improving the speed of calculation of the diameter and MPL between the nodes.
- Reducing the forming time and memory cost of adjacency matrices, or abandoning them as a whole.

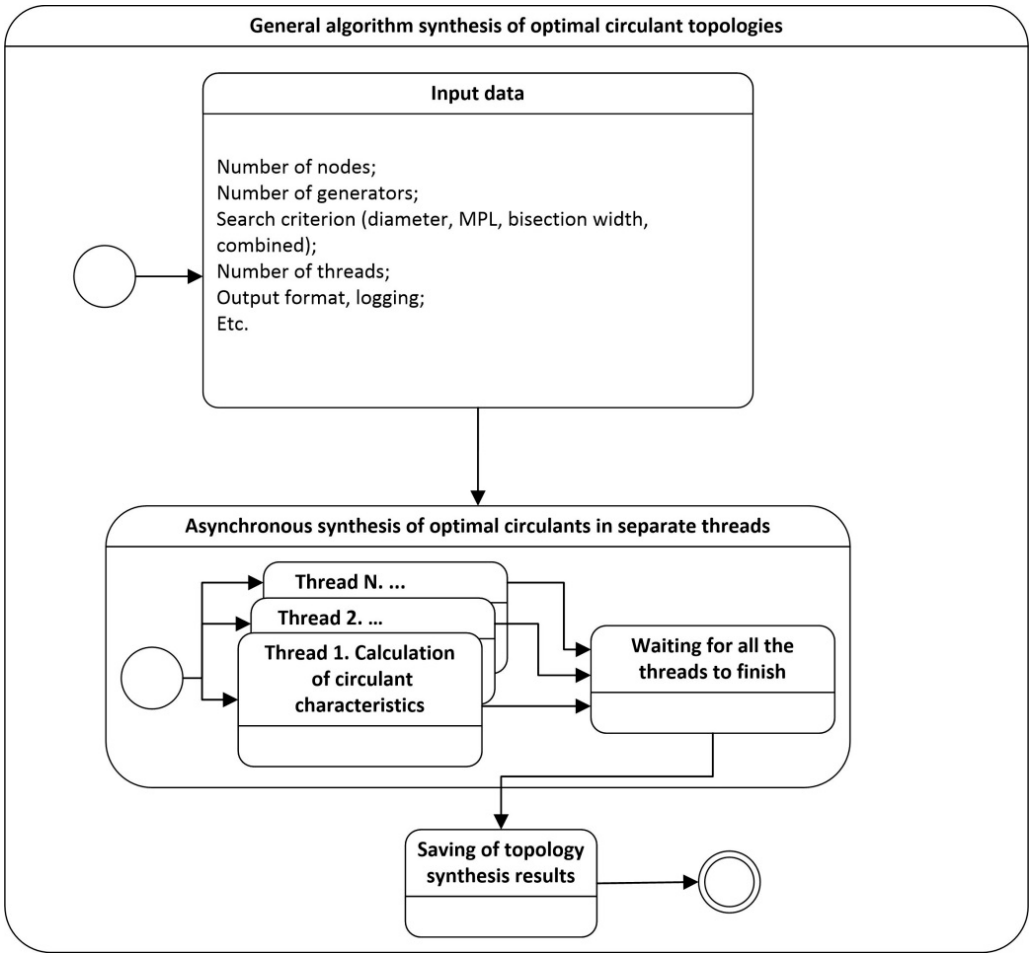

Figure 2. Generalized scheme of software for the synthesis of circulant topologies.

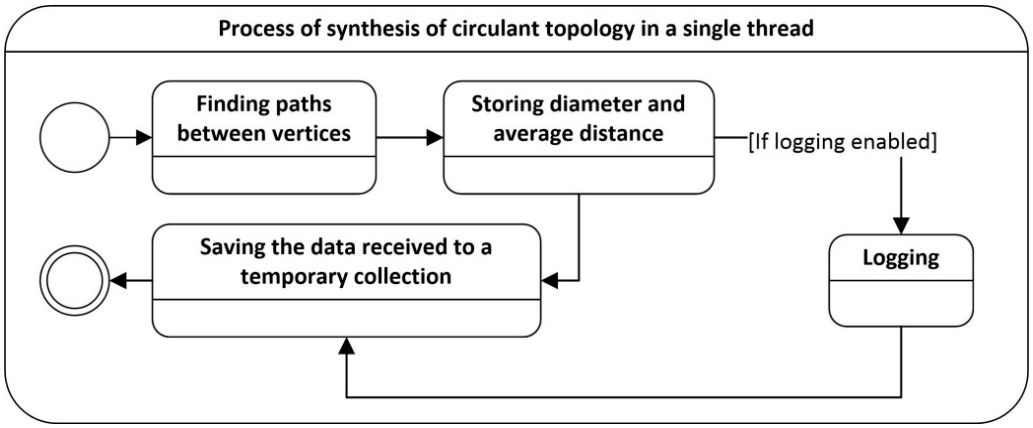

Figure 3. General scheme of software execution in single thread.

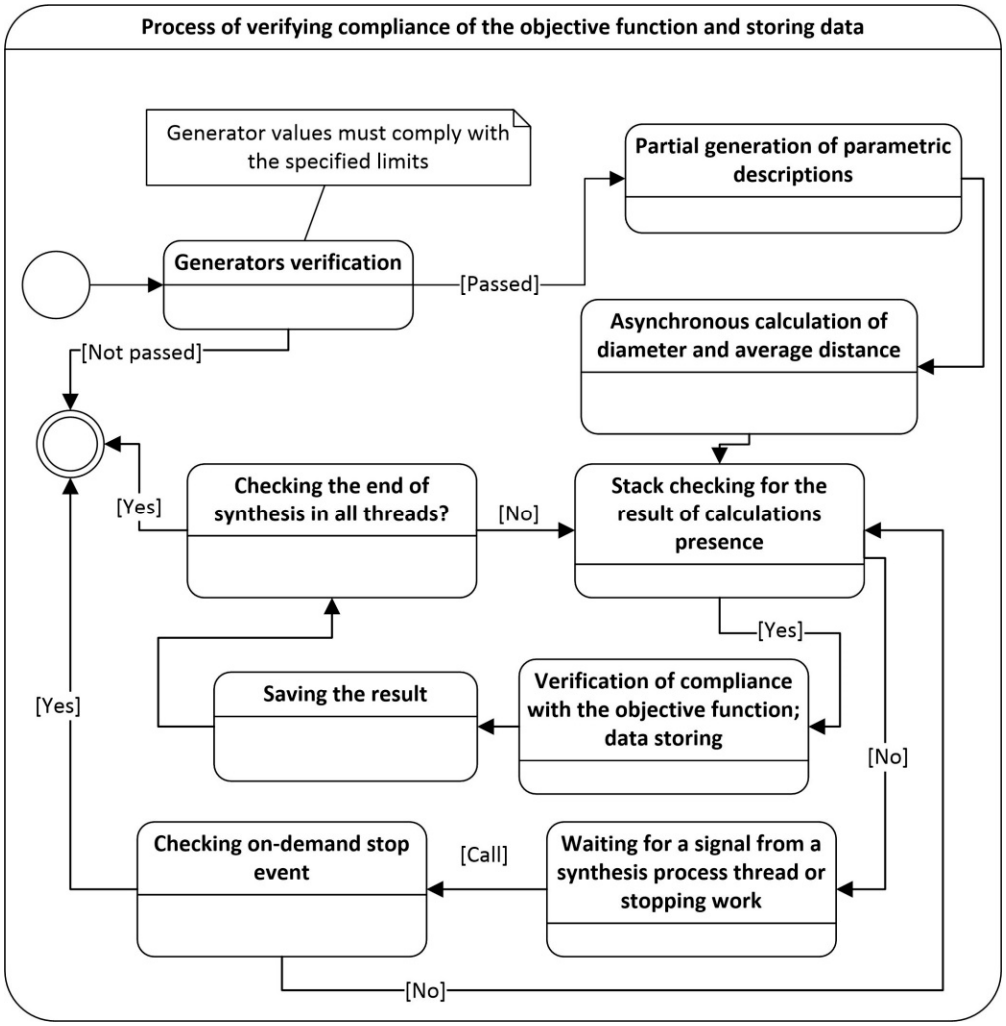

Figure 4. Description of software in multithread mode.

Considering Features of Circulant Graphs to Reduce the Number of Variants in an Exhaustive Search Algorithm

Let us consider the methods of search procedure acceleration applied. Denote the forming topologies as $(1 \leq s_i < N) \in S$ [29], where N—number of nodes, k—number of generators, $i = 1, \ldots, k$. If the value of the generator is greater than N, then the obtained parametric descriptions of the topology will be isomorphic to the descriptions $C(N; s_1\%N, s_2\%N, \ldots, s_k\%N)$, where $s_i \geq N$. By discarding isomorphic graph descriptions, $(N-1)\cdot k$ variants of possible topologies remain.

Another feature of circulant graphs is that circulant topologies with different signatures (for example, C(15; 3, 2) and C(15; 2, 3)) are the same [19]. To eliminate such isomorphism, condition $1 \leq s_1 < s_2 < \ldots < s_k < N$ is used (values of generators are arranged in ascending order; moreover, values of generators cannot be the same), which leads to a reduction in unnecessary repeated checks of topologies.

Another way towards reducing the number of searches is to use an isomorphism of circulant topologies of type $C(N; s_1, \ldots, s_i, \ldots, s_k)$ and $C(N; s_1, \ldots, N - s_i, \ldots, s_k)$ [19]. This feature allows us to bound iterating procedure over the values of generators $s_k < N/2 + N\%2$ (Figure 4).

Using the features of circulant topologies, it is possible to significantly reduce the number of iterations in an exhaustive search algorithm; the number of iterations in general $\approx (N/4 + 1/2)^k$, which leads to a reduction in the time for topologies of up to 10,000 times.

For most regular topologies, such as mesh, hypercube [30], and torus, the diameter and MPL can be calculated using formulas which depend on the number of nodes [3]. Circulant topologies are defined by graph order and generators; therefore, there are no universal formulas for the direct calculation of the diameter and MPL. In most cases, we can only estimate the maximum lower and upper boundaries of these parameters [19], which do not allow calculating metric data for specific instances of graphs. For only some classes of circulants, such as two-dimensional circulants, there are some formulas for finding the optimal topologies [10,21]. In a general case, the only way to find the diameter and MPL for the most generated topologies is the same as for those usual irregular topologies (Figure 4). For some families of circulants, there are fast algorithms for calculating some characteristics (for example, for ring two-dimensional circulants [16–18,31]) which can be used in the developed software. But for a general case, one can use Dijkstra's algorithm [32], the A* algorithm [33], and others. It is enough to find all the paths from only one random vertex to other vertices because circulant topologies can be described as circulant matrices of the Toeplitz matrix subspecies [34]. The diameter of the graph can be found as the largest path between the vertices, and the MPL as the arithmetic average of all paths [35].

3. Software Description

The developed software (placed along with the dataset [36]) finds the features of circulant topologies, such as diameter, MPL, and number of connections, and (on the ground of the data obtained) selects the optimal features by the criterion of minimizing the diameter and MPL. The software solution is a console application with the ability to configure input parameters by changing the configuration XML file.

The result is saved as files with parametric descriptions of the optimal topologies in the .csv and .bin formats, which contain extended information on the synthesis results (including the total calculation time).

3.1. Software Architecture

The software solution is implemented in the C# programming language and includes several components:

- Library for the synthesis of optimal circulant topologies.
- Console application for interacting with the library.
- Console application for evaluating synthesis time and validating results.

The main component is a library for the synthesis of optimal circulant topologies. It contains the basic functionality associated with the synthesis of circulant topologies. The library is easily portable to other projects including those developed in other languages. The second most important component is the console application for interacting with the library, which is responsible for calling the methods contained in it. The application for evaluating the synthesis and validation time of the results is designed to analyze the limit of the number of nodes for a single search configuration that will be executed in a reasonable time.

3.2. Description of the Features of the Software

The software interface is implemented as a console application. The input parameters, which the program should accept, are set in two files: a file with the extension .config and a separate XML file. The .config file is the manifest of the application and it is essentially also an XML file. It is supplemented by two parameters in the appSettings section: *FirstTaskPath*—path to the first file with the "task", and *LastStatePath*—path to the file saving the current progress in the synthesis.

The second file (directly the "task" for synthesis) contains a description of the following parameters:

- *nodesDescription* describes the number of nodes within which the topologies will be synthesized; this parameter can be specified as a range or can be listed with a comma.

- *dimension* sets the number of generators which will take part in the compilation of a parametric description of circulant topologies and is used for synthesis.
- *threadCount* is responsible for the number of threads allocated for synthesis; the application dynamically monitors the commands passed to it by the user or the program that launched the application (for example, the application can be paused or stopped); so, at least two threads are allocated for working: one for the commands, the other for synthesis. Other parameters which are responsible for the format of the output data include: logging of the synthesis process, saving intermediate results of the synthesis with signatures of non-optimal topologies, user identifier, and storing the information.

The application has a mechanism for maintaining progress. When the stop command is called, an XML file will automatically be created; the path to it is specified in the variable *LastStatePath*. This file has the following structure:

- *NodesDescription* is responsible for checking the progress save file; if, in the current task, the value of the *nodesDescription* parameter from the first file (its structure is described above) completely coincides with *NodesDescription*, the synthesis progress will be restored from the breakpoint; otherwise, this file will be ignored and overwritten when the application continues to work.
- *NotCheckedConfig* contains the vector of generators of the circulant topology from which it will be necessary to begin the search.
- *GoodConfigs* are the best topologies found at the time when the program has been stopped; with the continuation of the work, these topologies will be considered optimal by default.
- *Dimension* is determined by a given number of generators of the circulant topology.
- *Diameter* is the minimum diameter of the topology at the time when the program has been stopped.
- *AverageDiameter* is the minimum MPL at the time when the program has stopped working.
- *CurrentNodesCount* is the current number of nodes from which the task is to be continued.

4. Illustrative Examples

Currently, the developed software uses the enumeration of all topologies by their generators.

In this algorithm, the input parameter *nodesCount* is used to indicate the number of nodes; *config* is an array of generators in which one generator s_i is incremented, and the other generators (to the right of it) take the values $s_i + j - i$, where $i < j$; i, j—current positions of generators in the parametric description.

The next step in the synthesis algorithm is to find the minimum distances between one node and the rest of them. There are no accelerated algorithms for finding the diameter and MPL (as, for example, the one described in [16–18,31]), which could be applied to any circulant topologies. When the initial task starts, it is possible to choose one of several algorithms in the general case (Dijkstra's algorithm [32] or one of the modifications based on it).

Dijkstra's algorithm allows finding the exact distance between one node and the others, but with a large number of nodes, the size of the RAM used to store the adjacency matrix increases significantly. Partially, memory costs can be reduced by storing adjacency matrices called sparse [37], which helps in the case of topologies with a large number of nodes, including irregular ones. For circulant topologies, due to their features [19,38], a more effective solution is to completely get rid of the use of adjacency matrices as such.

The generators variable is an array of generators of the topology, d—array of minimum distances between the source node and the node with a given index.

With a small number of nodes and generators, Dijkstra's algorithm shows better results in the synthesis time compared to the modified algorithm. With a large number of nodes (from 5000 upwards) in the circulant topology, the achieved synthesis rate is noticeably lower than in the modified algorithm (Figures 5 and 6).

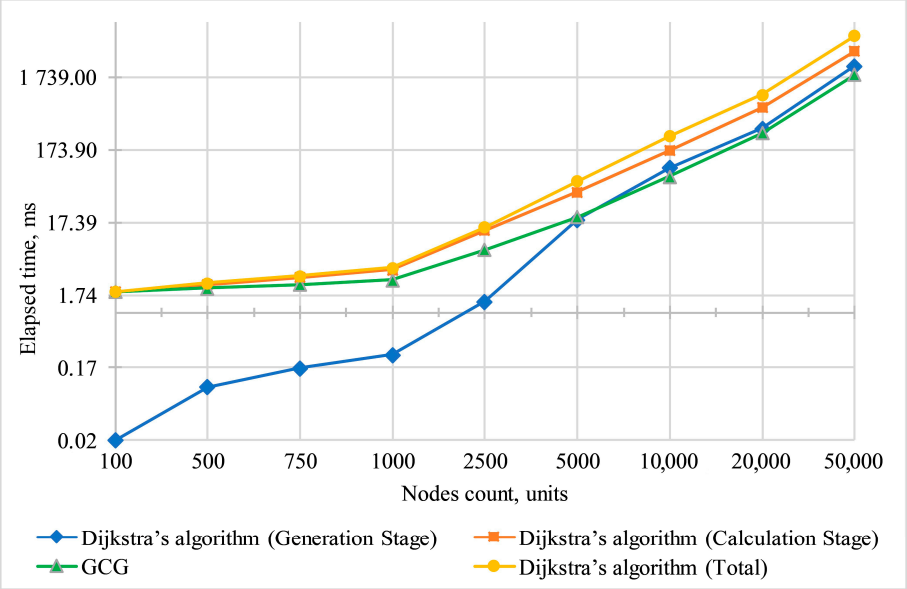

Figure 5. Diagram of speed of searching for the diameter and MPL for one topology when using the original Dijkstra's algorithm and GCG.

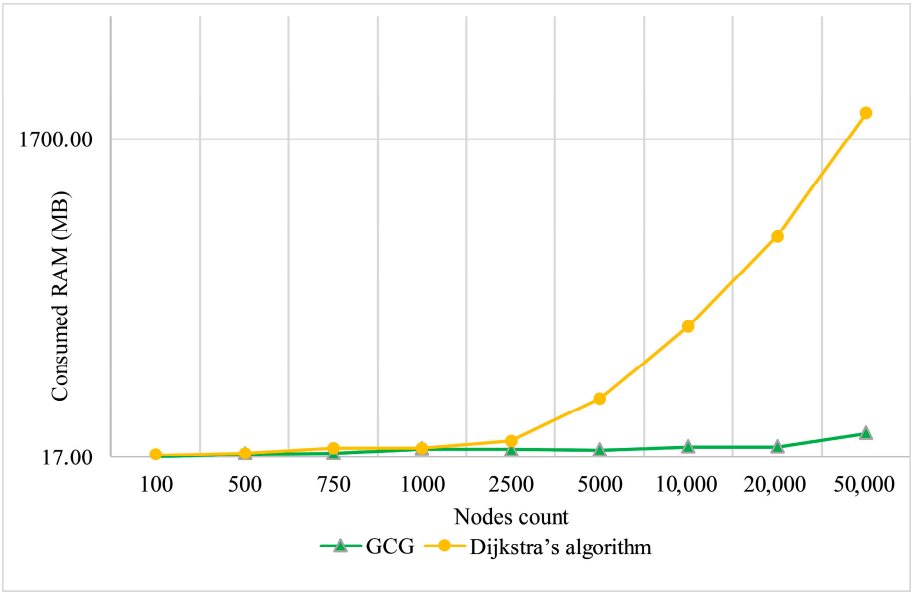

Figure 6. Diagram of memory consumption when using the original Dijkstra's algorithm and GCG.

4.1. Evaluation of the Results

To calculate the consumed RAM and the time it takes to find the diameter and MPL for one topology, a single-threaded mode was used. To reduce the error in the results obtained, the resulting arithmetic mean values are given. For each parametric description, a separate process was launched to correctly evaluate the consumed memory with a forced call to the garbage collector. The process was launched at the following configuration: 6-core Intel i7-8750H processor with a set frequency of 3.8 GHz with disabled hyper-threading (HT) and 16 GB of RAM. To estimate the approximate synthesis time of the optimal circulant topology, the number of iterations was obtained using the algorithm and the arithmetic mean of the time of searching for the diameter and MPL for one topology. Table 2 shows the estimated synthesis time for two algorithms: the unmodified Dijkstra's algorithm [32] and the optimized algorithm for circulant topologies.

Table 2. Estimated time of synthesis of three-dimensional circulant topologies.

Number of Nodes	Number of Iterations	Estimated Time (Dijkstra's Algorithm)	Estimated Time (GCG)	Acceleration
100	1224	2 ± 0.5 s	2 ± 0.5 s	100%
500	31,124	1 min 19 ± 10 s	1 min 8 ± 15 s	116%
750	70,124	3 min 45 ± 30 s	2 min 49 ± 20 s	133%
1000	124,749	8 min 39 ± 60 s	5 min 51 ± 30 s	148%
2500	780,624	3 h 15 ± 10 min	1 h 16 ± 20 min	203%
5000	3,123,749	56 ± 3 h	18 ± 2 h	312%
10,000	12,497,499	39 ± 3 days	11 ± 1 days	358%
20,000	49,994,999	1 year 7 ± 1 month	6 ± 1 month	340%
50,000	312,487,499	65 ± 3 years	18 ± 1 year	345%

The new GCG algorithm allows for an acceleration of more than three times for 5000 nodes. For more nodes, the search time is already unsatisfactory. To speed up the search, multithreading and cluster systems may be used so that the synthesis of circulant topologies could be parallelized.

Compared to the single-threaded mode of operation, the increase in productivity and the consumption of RAM are directly proportional to the number of logical cores that the system employs. It should be mentioned that the configuration of the system and its multiprocessing does not practically affect performance, and all stages are performed almost independently of each other, since the parallelization of the flows associated with the generation, calculation, and comparison of the results was performed.

4.2. Comparison with Other Solutions

The effectiveness of the proposed software solution can be demonstrated in comparison with the developments of other authors. For example, in [5,6], the results of the study of circulant topologies are presented for their application in the construction of high-performance clusters. The key point is that to search for circulants, the authors used their own algorithm based on a modified exhaustive search and ran it on a powerful SeaWulf supercomputer at Stony Brook University. At the same time, the circulant graphs considered in the articles do not exceed in their characteristic graphs, which can be obtained using GCG on just one computer.

4.3. Validation of the Topologies Generated

The correctness of the work of the developed software was validated by checking the characteristics of the found optimal circulants for compliance with the estimates of the maximum lower and upper boundaries of the circulant topologies diameter and MPL [19].

The results of the software proposed in this work coincide with the signatures of circulants found by other authors. In the Acknowledgments section of work [5], researchers thanked the author of the present work for the comments and recommendations made to them through e-mail correspondence and while discussing the preprint of their work.

The topologies generated by the developed software and their characteristics were used to test the calculations in preprint [18] and the formulas for estimating the diameter of $C_n(1,\ s)$ graphs presented in the preprint [31]. Due to this, an inaccuracy in the calculations was found.

4.4. Dataset

Using the developed software, a dataset [36] of circulant signatures was created for $k = 2 \ldots 10$. The dataset is divided into sections for each k. Each section contains catalogs with optimal circulants and ring optimal circulants. Entries in the dataset are .csv files containing variants of circulant signatures for a given N, which varies from 8 to 1000. The dataset is constantly replenished and updated due to the continuous operation of the GCG software on a separate high-performance server.

The resulting signatures of optimal circulants can be examined to identify regular dependencies between graph characteristics (diameter, MPL, generators values) and their node count, number of generators, etc. Also, the dataset can be used to train machine learning models [26] to predict the characteristics of graphs. Presenting the generated topologies in a dataset format opens up tremendous opportunities for studying optimal circulant graphs, as well as for searching for new families and formulas for estimating their characteristics. The search for new families of circulant graphs is an urgent and open problem. Until now, such datasets did not exist; there were analytical descriptions for a number of families of graphs only [19,24,25,29].

5. Use Case

5.1. New Topologies Search Speedup

The developed software and algorithm make it possible to accelerate the synthesis of optimal topologies for circulant topologies in comparison with the usual exhaustive search by a thousand times. The obtained optimal circulant topologies can be used to design NoCs or cluster systems with the best structural characteristics and to solve one of the fundamental scientific problems in the field of computer systems: the construction of communication structures and algorithms for exchanging data in networks on next-generation chips. Thus, this solution is a powerful tool for developers of new topologies and methods of communication in multicore systems; it can also be used by mathematicians to study the features of various families of circulant graphs and to search for new families. Despite the fact that the GCG algorithm has a lower speed compared with the implementation based on genetic algorithms [39], it guarantees that all signatures of optimal circulant graphs exist for given conditions. Thus, the proposed software is excellent for verifying the results of other algorithms based on bio-inspired optimization algorithms and analytical and hybrid methods.

This work is intended for a narrow range of tasks but, nevertheless, the project is open; it is constantly evolving both towards improving the speed of the algorithm, the quality of the source code, and adding the ability to search for regular graphs with topologies of other types.

5.2. The Analysis of Routing Algorithms for Circulant Topologies

An important task is not only the generation of new topologies, but also the development of routing algorithms in them. There are many approaches to routing in circulants, for example, those described in [29,40–44]. However, they are often developed for the general case of routing and do not take into account deadlocks, livelocks, starvation, and faults [45,46]. The developed software allows generating complete maps of routes possible for specific circulant signatures, as well as implementing different routing strategies that can be used to calculate target metrics when searching for optimal graphs. i.e., it is possible to search for circulant topologies that are optimal for a particular routing algorithm.

In work [47], a routing method well suited for use in circulants is proposed: the centers of the hierarchy around which the distance means the distance along a certain virtual coordinate are preliminarily marked in the graph. This allows fault-tolerant routing based on performing calculations on coordinate vectors. It was also found experimentally that there were circulant signatures in which the neighborhoods of the centers of the hierarchy were completely filled with nodes. Such circulants are optimal for use with the routing algorithm proposed. The developed software allows searching for such topologies.

5.3. Regression Analysis

The signatures of circulants from the generated dataset were subjected to regression analysis, which made it possible to obtain approximation formulas using linear regression [48]; thanks to the latter, it is possible to predict the parameters of circulants depending on their characteristics. So, for example, Figures 7–9 show graphs for the obtained formulas

for calculating MPL from N for $k = 3,\ 4,\ 5$ (the most promising circulant graphs dimensions for NoCs design), respectively:

$$\mathrm{MPL}_{circ}\ (N,\ k = 3) = (0.313 \cdot N - 2.355)^{\frac{1}{3}}, \mathrm{MAE} = 0.02,\ \mathrm{PPMCC} = 1.00 \quad (1)$$

$$\mathrm{MPL}_{circ}\ (N,\ k = 4) = (0.505 \cdot N - 8.023)^{\frac{1}{4}}, \mathrm{MAE} = 0.05,\ \mathrm{PPMCC} = 0.98 \quad (2)$$

$$\mathrm{MPL}_{circ}\ (N,\ k = 5) = (0.955 \cdot N - 33.896)^{\frac{1}{5}}, \mathrm{MAE} = 0.04,\ \mathrm{PPMCC} = 0.99 \quad (3)$$

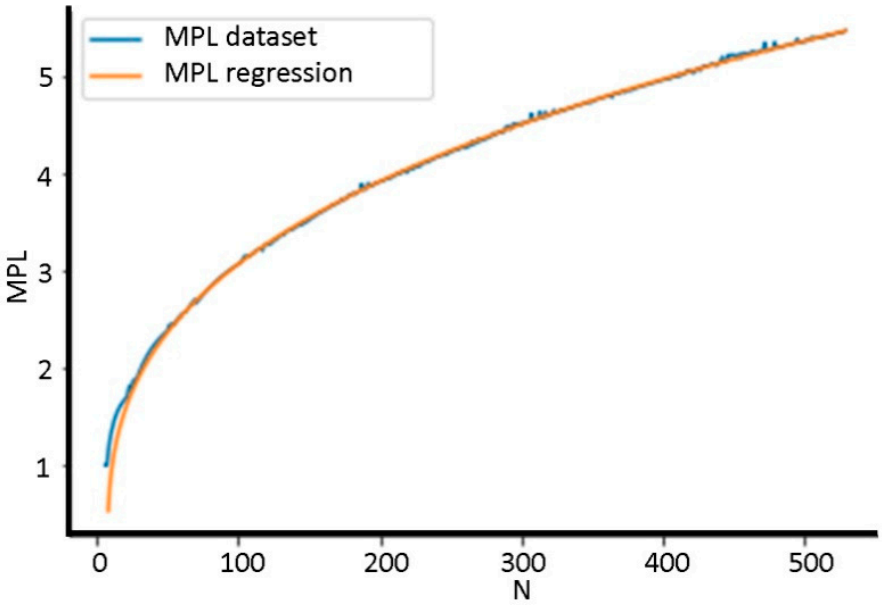

Figure 7. Dependence of MPL on N for $k = 3$.

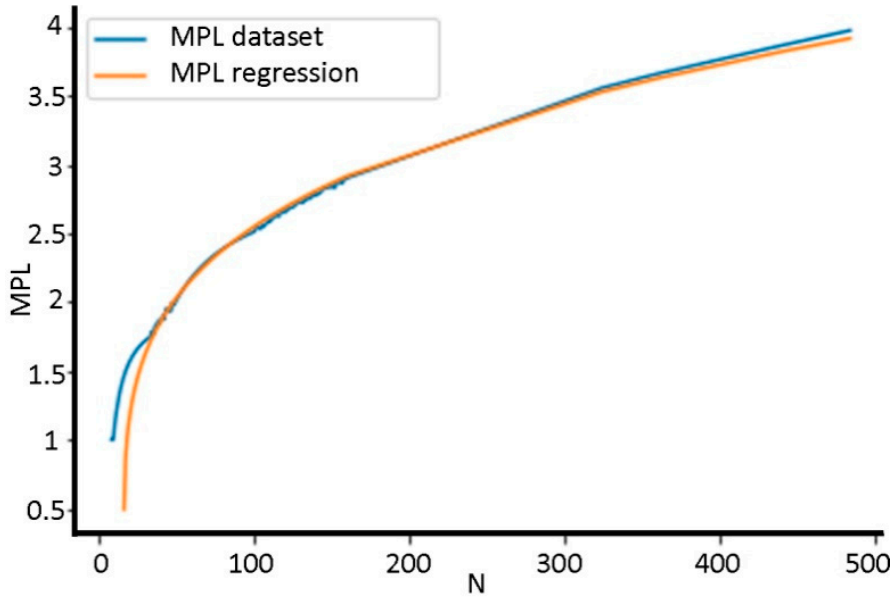

Figure 8. Dependence of MPL on N for $k = 4$.

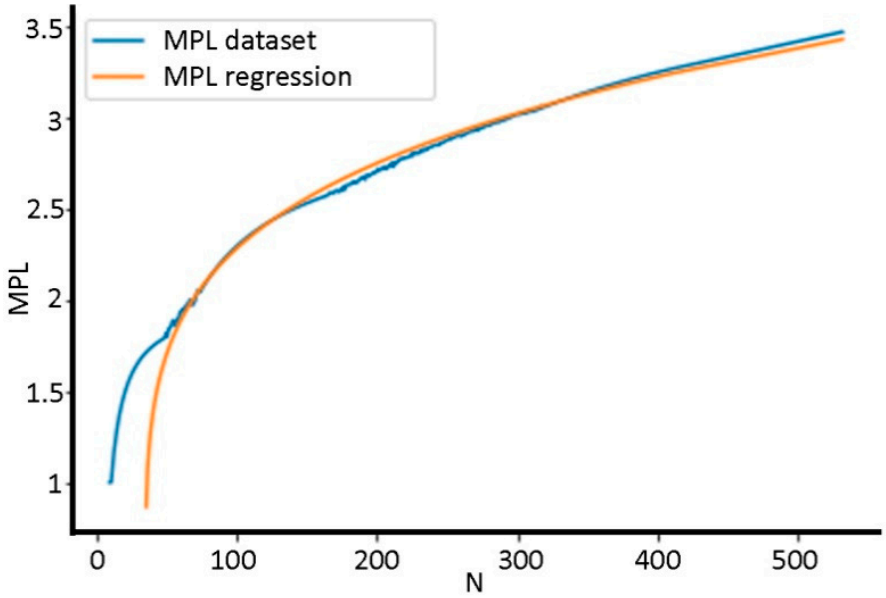

Figure 9. Dependence of MPL on N for $k = 5$.

The calculated values of the mean absolute error (MAE; no more than 0.05) and Pearson product-moment correlation coefficient (PPMCC; no less than 0.98) allow us to state that even such simple formulas accurately describe the dependence of MPL on N and can be used in practice. Further analysis of the dataset using more advanced regression analysis methods [49] and machine learning methods [26] will provide more accurate estimates for various parameters of optimal circulant graphs.

6. Conclusions

Thus, we presented a software solution for the synthesis of circulant topologies with any number of generators and the order of the graph reaching thousands of nodes. The existing algorithm for the synthesis of circulant topologies was improved so that the search for a topology with a minimum diameter and an MPL between nodes could be sped up. The software supports multi-threaded calculations and makes it possible to synthesize optimal circulant topologies in an acceptable time, as well as to collect statistics on other circulant topologies. The existing application logic was reworked and the main part (responsible for the synthesis of circulant topologies) was moved to a separate library. The infrastructure for expanding the software solution through the use of parallel computing was implemented, which makes it possible to further accelerate the application and, accordingly, to increase the order of graphs that can be synthesized using this software development. Compared with the basic Dijkstra's algorithm, the computation speed was increased by approximately 35,000 times and 3.5 times compared with the modified version, and a linear dependence of the consumed RAM on the number of nodes per thread (unlike the quadratic dependence in the basic version of Dijkstra's algorithm) was achieved.

Using a large dataset of optimal circulant topologies opens up good opportunities for further research on these topologies, as well as for the identification of the dependencies between their parameters and characteristics and for developing new routing algorithms for them. The obtained optimal circulant topologies can be used to check the correctness of the generated topologies and also methods for the accelerated calculation of the characteristics of circulant topologies obtained by other authors.

Funding: The results of this research were obtained within the RSF grant (project No. 22-29-00979).

Data Availability Statement: The software and dataset of this research are available from [36] and https://github.com/RomeoMe5/circulantGraphs (accessed on 12 Aril 2023).

Acknowledgments: The author acknowledges Glukhikh A.Y. and Kotov F.I. for doing some calculations and the support provided.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Lu, R. Fast Methods for Designing Circulant Network Topology with High Connectivity and Survivability. *J. Cloud Comput.* **2016**, *5*, 5. [CrossRef]
2. Benmessaoud Gabis, A.; Koudil, M. NoC Routing Protocols—Objective-Based Classification. *J. Syst. Archit.* **2016**, *66–67*, 14–32. [CrossRef]
3. Baby, N.; Mathew, S.; Abraham, S.; Ravindranath, S.; Sanju, V. Network on Chip Simulator: Design, Implementation and Comparison of Mesh, Torus and RiCoBiT Topologies. In Proceedings of the 2016 2nd International Conference on Next Generation Computing Technologies, NGCT 2016, Dehradun, India, 14–16 October 2016; pp. 46–50. [CrossRef]
4. Lau, F.C.M.; Chen, G. Optimal Layouts of Midimew Networks. *IEEE Trans. Parallel Distrib. Syst.* **1996**, *7*, 954–961. [CrossRef]
5. Huang, X.; Ramos, A.F.; Deng, Y. Optimal circulant graphs as low-latency network topologies. *J. Supercomput.* **2022**, *78*, 13491–13510. [CrossRef]
6. Deng, Y.; Guo, M.; Ramos, A.F.; Huang, X.; Xu, Z.; Liu, W. Optimal low-latency network topologies for cluster performance enhancement. *J. Supercomput.* **2020**, *76*, 9558–9584. [CrossRef]
7. Puente, V.; Izu, C.; Beivide, R.; Gregorio, J.A.; Vallejo, F.; Prellezo, J.M. The Adaptive Bubble Router. *J. Parallel Distrib. Comput.* **2001**, *61*, 1180–1208. [CrossRef]
8. Yang, Y.; Funahashi, A.; Jouraku, A.; Nishi, H.; Amano, H.; Sueyoshi, T. Recursive Diagonal Torus: An Interconnection Network for Massively Parallel Computers. *IEEE Trans. Parallel Distrib. Syst.* **2001**, *12*, 701–715. [CrossRef]
9. Beivide, R.; Martínez, C.; Izu, C.; Gutierrez, J.; Gregorio, J.Á.; Miguel-Alonso, J. Chordal Topologies for Interconnection Networks. In *High Performance Computing, Proceedings of the 5th International Symposium, ISHPC 2003, Tokyo, Japan, 20–22 October 2003*; Springer: Berlin, Germany, 2003; Volume 2858. [CrossRef]
10. Atajan, T.; Otsuka, N.; Yong, X. Counting the Number of Spanning Trees in a Class of Double Fixed-Step Loop Networks. *Appl. Math. Lett.* **2010**, *23*, 291–298. [CrossRef]
11. Liestman, A.L.; Opatrny, J.; Zaragozá, M. Network Properties of Double and Triple Fixed Step Graphs. *Int. J. Found. Comput. Sci.* **1998**, *09*, 57–76. [CrossRef]
12. Attrah, N.H.; Abdul-Majeed, G.H.; Abdullah, M.Z. Implementation of Chordal Ring Network Topology to Enhance the Performance of Wireless Broadband Network. *EUREKA. Phys. Eng.* **2021**, *2021*, 11–18. [CrossRef]
13. Bulut, A.; Hacioglu, I. Asymptotic Energy of Connected Cubic Circulant Graphs. *AKCE Int. J. Graphs Comb.* **2021**, *18*, 25–28. [CrossRef]
14. Azeem, M.; Imran, M.; Nadeem, M. Sharp Bounds on Partition Dimension of Hexagonal Möbius Ladder. *J. King Saud Univ.—Sci.* **2022**, *34*, 101779. [CrossRef]
15. Newman, M.E.J. The Structure and Function of Complex Networks. *SIAM Rev.* **2003**, *45*, 167–256. [CrossRef]
16. Chen, B.; Xiao, W.; Parhami, B. Diameter Formulas for a Class of Undirected Double-Loop Networks. *J. Interconnect. Networks* **2005**, *6*, 1–15. [CrossRef]
17. Zerovnik, J.; Pisanski, T. Computing the Diameter in Multiple-Loop Networks. *J. Algorithms* **1993**, *14*, 226–243. [CrossRef]
18. Loudiki, L.; Kchikech, M.; Essaky, E.H. Diameter Formulas for a Class of Undirected Multi-Loop Networks. *arXiv* **2022**. [CrossRef]
19. Monakhova, E.A. A Survey on Undirected Circulant Graphs. *Discret. Math. Algorithms Appl.* **2012**, *4*, 1250002. [CrossRef]
20. Romanov, A.Y.; Romanova, I.I.; Glukhikh, A.Y. Development of a Universal Adaptive Fast Algorithm for the Synthesis of Circulant Topologies for Networks-on-Chip Implementations. In Proceedings of the 2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO), Kyiv, Ukraine, 24–26 April 2018; IEEE: New York, NY, USA, 2018; pp. 110–115. [CrossRef]
21. Herrada, E.; Balcazar, J.L.; Arruabarrena, A. Optimal distance networks of low degree for parallel computers. *IEEE Trans. Comput.* **1991**, *40*, 1109–1124. [CrossRef]
22. Park, J.-H.; Chwa, K.-Y. Recursive Circulant: A New Topology for Multicomputer Networks. In Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN), Kanazawa, Japan, 14–16 December 1994; IEEE Computer Society Press: New York, NY, USA, 1994; pp. 73–80. [CrossRef]
23. Cheng, D.-W.; Yao, K.-H.; Hsieh, S.-Y. Constructing Independent Spanning Trees on Generalized Recursive Circulant Graphs. *IEEE Access* **2021**, *9*, 74028–74037. [CrossRef]
24. Stojmenovic, I. Multiplicative circulant networks. Topological properties and communication algorithms. *Discr. Appl. Math.* **1997**, *77*, 281–305. [CrossRef]
25. Lewis, R.R. Analysis and Construction of Extremal Circulant and Other Abelian Cayley Graphs. The Open University: Hong Kong, China, 2021. [CrossRef]
26. Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; Leskovec, J. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *Adv. Neural Inf. Process Syst.* **2020**, *33*, 22118–22133. [CrossRef]

27. Manevich, R.; Polishuk, L.; Cidon, I.; Kolodny, A. Designing Single-Cycle Long Links in Hierarchical NoCs. *Microprocess. Microsyst.* **2014**, *38*, 814–825. [CrossRef]
28. Ogras, U.Y.; Marculescu, R.; Lee, H.G.; Chang, N. Communication Architecture Optimization: Making the Shortest Path Shorter in Regular Networks-on-Chip. In Proceedings of the Design, Automation and Test in Europe, Munich, Germany, 6–10 March 2006. [CrossRef]
29. Elspas, B.; Turner, J. Graphs with Circulant Adjacency Matrices. *J. Comb. Theory* **1970**, *9*, 3. [CrossRef]
30. Saldaña, M.; Shannon, L.; Chow, P. The Routability of Multiprocessor Network Topologies in FPGAs. In Proceedings of the Internation Symposium on Field Programmable Gate Arrays—FPGA'06, Monterey, CA, USA, 22–24 February 2006; ACM Press: New York, NY, USA, 2006; p. 232. [CrossRef]
31. Loudiki, L.; Kchikech, M.; Essaky, E.H. A New Approach for Computing the Distance and the Diameter in Circulant Graphs. *arXiv* **2022**. [CrossRef]
32. Dijkstra, E.W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
33. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
34. Abbott, S.; Bottcher, A.; Grudsky, S.M. Toeplitz Matrices, Asymptotic Linear Algebra and Functional Analysis. *Math. Gaz.* **2000**, *84*, 572. [CrossRef]
35. Boesch, F.; Tindell, R. Circulants and Their Connectivities. *J. Graph Theory* **1984**, *8*, 487–499. [CrossRef]
36. Optimal Circulant Topologies Dataset. Available online: https://doi.org/10.5281/zenodo.7265637 (accessed on 3 March 2023).
37. Tamimi, A.A. Comparison Studies for Different Shortest Path Algorithms. *Int. J. Comput. Technol.* **2015**, *14*, 5979–5986. [CrossRef]
38. Romanov, A.Y. Development of Routing Algorithms in Networks-on-Chip Based on Ring Circulant Topologies. *Heliyon* **2019**, *5*, e01516. [CrossRef]
39. Monakhova, E.A.; Monakhov, O.G.; Mukhoed, E.V. Genetic Construction of Optimal Circulant Network Designs. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin, Germany, 1999; Volume 1596, pp. 215–233. [CrossRef]
40. Robic, B. *Optimal Routing in 2-Jump Circulant Networks*; Univ. Cambridge: Cambridge, UK, 1996. [CrossRef]
41. Chen, B.-X.; Meng, J.-X.; Xiao, W.-J. A constant time optimal routing algorithm for undirected double-loop networks. *Lect. Notes Comp. Sci.* **2005**, *3794*, 308–316. [CrossRef]
42. Gomez, D.; Gutierrez, J.; Ibeas, A.; Martinez, C.; Beivide, R. On Finding a Shortest Path in Circulant Graphs with Two Jumps. *Lect. Notes Comp. Sci.* **2005**, *3595*, 777–786. [CrossRef]
43. Dobravec, T.; Zerovnik, J.; Robic, B. An Optimal Message Routing Algorithm for Circulant Networks. *J. Syst. Arch.* **2006**, *52*, 298–306. [CrossRef]
44. Perez-Roses, H.; Bras-Amoros, M.; Serradilla-Merinero, J.M. Greedy Routing in Circulant Networks. *Graphs Comb.* **2022**, *8*, 86. [CrossRef]
45. Das, S.; Karfa, S.; Biswas, S. Formal Modeling of Network-on-Chip Using CFSM and its Application in Detecting Deadlock. *IEEE Trans. Very Large Scale Integr. Syst.* **2020**, *28*, 1016–1029. [CrossRef]
46. Nadeem, M.; Imran, M.; Siddiqui, H.M.; Azeem, M. Fault Tolerance Designs of Interconnection Networks. *Peer-to-Peer Netw. Appl.* **2023**, *1*, 1–10. [CrossRef]
47. Romanov, A.; Myachin, N.; Sukhov, A. Fault-Tolerant Routing in Networks-on-Chip Using Self-Organizing Routing Algorithms. In Proceedings of the IECON 2021—47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, ON, Canada, 13–16 October 2021; pp. 1–6. [CrossRef]
48. Montgomery, D.C.; Peck, E.A.; Vining, G.G. *Introduction to Linear Regression Analysis*, 6th ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2021.
49. Freedman, D. *Statistical Models: Theory and Practice*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2009. [CrossRef]