mathematics

MDPI

Article

A New Record of Graph Enumeration Enabled by Parallel Processing

Zhipeng Xu [1,2,3], Xiaolong Huang [3], Fabian Jimenez [4] and Yuefan Deng [3,*]

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong 510006, China; xuzhp9@mail2.sysu.edu.cn

[2] Guangdong Province Key Laboratory of Computational Science, Guangzhou, Guangdong 510275, China

[3] Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794, USA; xiaolong.huang@stonybrook.edu

[4] Department of Technological Operations and High Performance Computing, Empresa Publica Yachay, Urcuquí, Imbabura 100115, Ecuador; fjimenez@yachay.gob.ec

* Correspondence: yuefan.deng@stonybrook.edu

Received: 13 November 2019; Accepted: 5 December 2019; Published: 10 December 2019

check for updates

Abstract: Using three supercomputers, we broke a record set in 2011, in the enumeration of non-isomorphic regular graphs by expanding the sequence of A006820 in the Online Encyclopedia of Integer Sequences (OEIS), to achieve the number for 4-regular graphs of order 23 as 429,668,180,677,439, while discovering several regular graphs with minimum average shortest path lengths (ASPL) that can be used as interconnection networks for parallel computers. The enumeration of 4-regular graphs and the discovery of minimal-ASPL graphs are extremely time consuming. We accomplish them by adapting GENREG, a classical regular graph generator, to three supercomputers with thousands of processor cores.

Keywords: regular graph; parallel computing; dynamical scheduling

MSC: 05C30; 68R10

1. Introduction

The analysis of regular graphs for their properties, including eigen-spectra and automorphisms, is a fertile field for discovery and applications in algebraic graph theory [1–3]. Yet, there are many unsolved problems, e.g., Conway's 99-graph problem [4] and the 57-regular Moore graph [5]. For the analysis of interconnection networks, regularity is essential for its direct and useful relationship with the complexity of network implementation, and as such, many regular graphs including the Peterson graph, hypercube graph, and their extensions [6–11] are widely used to construct interconnection networks for parallel computers.

For 3-regular graphs of order n, Robinson and Wormald [12,13] presented all counting results for $n \leq 40$, while pointing out that enumeration for unlabeled k-regular graphs with $k > 3$ is an unsolved problem. Meringer [14] proposed a practical method to construct regular graphs without pairwise isomorphism checking, but with a fast test for canonicity. Brankmann [15,16] developed minibaum and snarkhunter for generating 3-regular graphs. "The House of Graphs" [17] and the Online Encyclopedia of Integer Sequences (OEIS) [18] databased the latest results for numbers of regular graphs. Kimberley [19] contributed many results (A068934) to these databases by a package called GENREG developed by Meringer [14]. In addition to its challenges of pure mathematics, the enumeration problem is the root of topics in reliability, artificial intelligence, reasoning, statistical physics [20], life sciences, chemistry [21], and even the search for the origins of life [22].

GENREG, efficient for small scale clusters due to its feature of task partition, approaches a hard wall of speedup for fine grained partitioning on large scale clusters, caused mainly by load imbalance. To obtain larger graphs, we extend GENREG for distributed clusters by using the message passing interface (MPI) [23]. Using the parallel GENREG we developed, we obtained the following results:

(1) filtered all 3-regular graphs up to order 32 with minimum average shortest path lengths (ASPL);
(2) discovered thousands of 4-regular graphs of order 32 with minimum ASPL;
(3) generated the exact counts of 4-regular graphs of order 23 by using the three supercomputer clusters located in the U.S., China, and Ecuador.

Among our results, the first and second were applied to the interconnection network research [24] for benchmarking the relationship of the graph ASPL to network performance latencies. Then, Zhang et al. [25] continued to analyze these candidate graphs and achieved optimal graphs with the properties of high throughput and symmetry. The third expands n from 22 to 23, for the first time, in the sequence A006820 [26] of OEIS, which is the number of connected 4-regular graphs of order n. Kimberley [26] used GENREG to enumerate the 4-regular graphs for up to the order 22 in 2011 [27]. This record for $n = 22$ remained unchallenged until our enumeration for $n = 23$, enabled by our parallel computing implementation to advance it a step.

2. The Enumeration Framework and Results

2.1. The Enumeration Function

For enumerating the regular graphs, published packages such as minibaum [15], snarkhunter [16], and GENREG have their own strengths and weaknesses. GENREG is more general in covering the graph degrees than minibaum and snarkhunter, which only support 3-regular graphs.

In our parallel computing framework, we designate one node as the master, whose task involves adaptive scheduling and dispatching, and the rest as a team of workers. When our program starts, the workers send a message to the master to request a task, and the workers continue the requests until the list of tasks is exhausted. As usual, when the master sends a task to a worker, this task is marked as selected and becomes unavailable. At last, when the task pool empties, the master signals all workers to exit.

Our dynamical scheduling strategy keeps cores in the cluster busy for useful tasks to allow us an efficient search for graphs with specified parameters, e.g., diameters or eigenvalues, by inserting external serial programs. In addition to load balance, our parallel program reduces the communication cost to $N_{\text{task}} \times 2$, a lesser requirement of bandwidth because the message itself is the message count. If we use a dedicated thread for task scheduling, the scalability and limit of the maximum computer system can both shrink. Depending on the communication sub-system, the maximum scalable system our current approach can reach is approximately 3000 cores, due to communication congestions, eventually. We may improve the scalability of our program by a multi-level scheduling; particularly for the many-core systems.

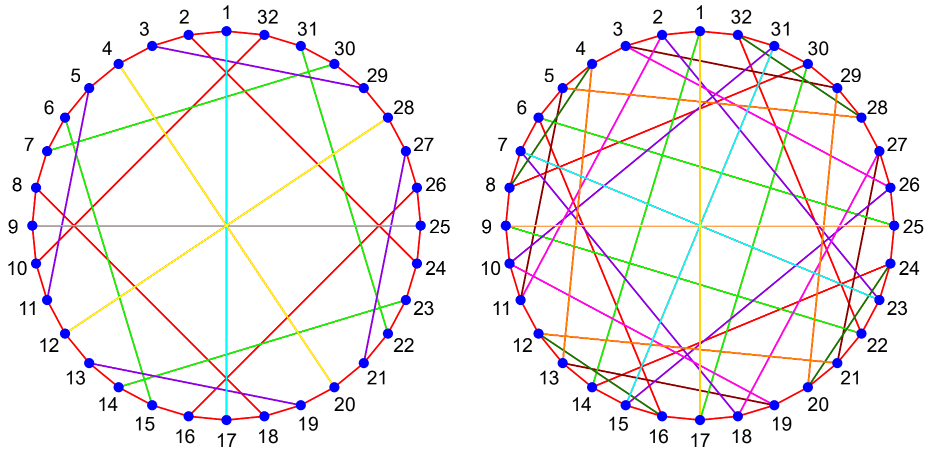
2.2. Search for a Regular Graph with Minimal ASPL

In the interconnection networks of supercomputers and data centers, regularity is a very significant feature because it is related to the complexity of the network configuration. For the topologies of regular graphs applied to the interconnection networks, it is highly desirable to obtain graphs with minimal ASPL because they help reduce communication latencies. Let $d(i,j)$ be the distance between vertex i and j. The ASPL is calculated as follows,

$$\text{ASPL} = \frac{1}{n(n-1)} \sum d(i,j). \tag{1}$$

Cerf [28] calculated and proved the lower bound of ASPL; hence, the optimality criterion is to find graphs with minimum ASPL. Usually and thus far, random or heuristic methods or intuitions resorted to searching for graphs with such desired properties for large networks [29–31]; Graph Golf [32], a competition of searching for graphs with the minimal diameters and ASPL, generated many graphs that are very common, but asymmetric. Based on this framework, we discovered a series of optimal graphs with all desired properties including minimum diameters, ASPL, high symmetry, and robustness [25], but with one disadvantage: smaller graphs. However, these graphs can be adopted in smaller clusters, or multiple modules of these clusters, or a system-on-chip directly and be expanded by using the Cartesian product [33].

Using this framework, we decomposed the search into 200,000 sub-tasks, which considered balancing computing time and the quantity of sub-tasks after a large number of experiments, and we completed the exhaustive search of all of the 3-regular graphs of order 32 with diameter four and minimal ASPL. The program was run on the Sunway-Bluelight supercomputer [34] with 80,000 cores for 72 h and discovered 56 graphs with the minimal ASPL after exhausting all 18,941,522,184,590 possible graphs predicted by Robinson et al. [12], who only enumerated them without finding the graphs with the desired properties including the minimum diameter and ASPL. Deng et al. [24] applied one (Figure 1a) of these graphs to construct a Beowulf cluster [35], and their benchmark results showed that the graphs with the minimal ASPL outperformed other mainstream topologies.



(**a**) (32,3)-optimal

(**b**) (32,4)-optimal

Figure 1. The graphs with minimal average shortest path lengths (ASPL) used in, for example, a Beowulf cluster [24].

Figure 2 shows the distribution of generated graph numbers for all cases, which look like a Gaussian distribution with the long tail containing hundreds of millions of graphs, and the frequency fluctuates by four orders of magnitude. Clearly, the tasks of enumerations vary greatly from graph to graph, and our dynamic task scheduling eliminated the substantial waiting time due to load imbalance.

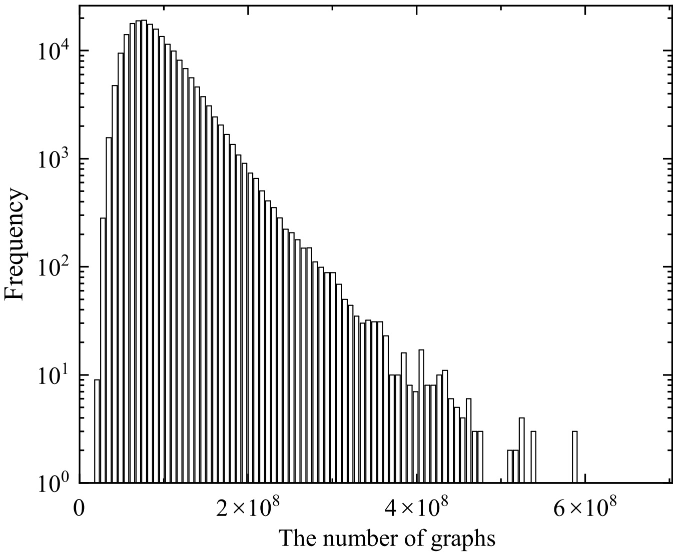


Figure 2. The number of graphs and frequency.

2.3. Graph Counting for (23,4)-Regular Graphs

When we search for the minimum ASPL graphs among all possible 4-regular graphs of order 32, there is no enumeration result for this scale when $girth \leq 5$. We can still use our software to search for regular graphs with minimum ASPL, as shown in Figure 1b, without confirmation of exhaustiveness. Therefore, we verified the results in Table 1 after decomposing the problem into 50,000 sub-tasks and setting the split level as 12. We managed to increase the n of the sequence A006820 from 22 to 23 and confirmed the number of 4-regular graphs of order 23 as 429,668,180,677,439 by using our parallel GENREG with the same parameters.

Our work to obtain the new enumeration for $n = 23$ was estimated to cost nearly 100 core-years. We ganged three supercomputers, the SeaWulf at Stony Brook University, while the Tianhe-1 with Intel Xeon X5670 processors and the IBM Quinde 1 with Power8 processors can process 113,000 and 56,469 graphs per second per core, respectively. As shown in Table 2, the SeaWulf with Intel Xeon Gold 6148 processors had the highest efficiency for searching 178,000 graphs per second per core, while the Tianhe-1 with Intel Xeon X5670 processors and the IBM Quinde 1 with Power8 processors contributed the rest.

In fact, the result of graph counting for (23,4)-regular graphs was obtained by the strategic and opportunistic use of the fragmented and shared computing resources. In addition to the policy of a fair and efficient share for most supercomputers, the Tianhe-1 would terminate tasks that last long for many cores because of maintenance, which prevented us from running long tasks. The external scheduling system we developed helped overcome the limitations of computing resources while facilitating optimal utilization of the occasionally available cores.

Table 1. Online Encyclopedia of Integer Sequences (OEIS) A006820 with our record-breaking new result for $n = 23$ highlighted in bold.

Order n	Quartics
5	1
6	1
7	2
8	6
9	16
10	59
11	265
12	1544
13	10,778
14	88,168
15	805,491
16	8,037,418
17	86,221,634
18	985,870,522
19	11,946,487,647
20	152,808,063,181
21	2,056,692,014,474
22	28,566,273,166,527
23	**429,668,180,677,439**

Table 2. Cost on three clusters.

Cluster	Total Processed (10^{12} Graphs)	Computing Time (Core Years)	Core Speed (10^3 Graphs/s)
SeaWulf	371.13	66.12	178
Tianhe-1	69.51	19.53	113
IBM Quinde 1	23.60	13.25	56

3. Conclusions

Our parallel method adapting GENREG enabled us to complete the search and enumeration on systems of 3000 processor cores. For the first time, using this new approach, we discovered several graphs of order 32 with minimal ASPL and found the enumeration count for 4-regular graphs of order 23, gaining confidence in the graph theory community that high performance computing can help solve otherwise intractable problems.

Author Contributions: Z.X. and X.H. designed the algorithms and completed the search; F.J. provided a portion of the hardware resources and participated in several stages of discussions; Y.D. guided the project and carried out the manuscript revisions and finalization.

Funding: The research of Z.X. is partially supported by the Special Project on High-Performance Computing of the National Key R&D Program under No.2016YFB0200604.

Acknowledgments: The authors thank Stony Brook Research Computing and Cyberinfrastructure, and the Institute for Advanced Computational Science at Stony Brook University for access to the high-performance SeaWulf computing system, which was made possible by a $1.4M National Science Foundation grant (#1531492). Also, they thank the National Supercomputing Centers in Jinan and Changsha in China, for computing resources, and M. Meringer of the German Aerospace Center, for technical support regarding GENREG and beneficial suggestions for the manuscript via e-mails.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Godsil, C.; Royle, G. *Algebraic Graph Theory*; Springer: New York, NY, USA, 2001. [CrossRef]
2. Jäntschi, L. The Eigenproblem Translated for Alignment of Molecules. *Symmetry* **2019**, *11*, 1027. [CrossRef]

3. Joiţa, D.M.; Jäntschi, L. Extending the Characteristic Polynomial for Characterization of C20 Fullerene Congeners. *Mathematics* **2017**, *5*, 84. [CrossRef]
4. Conway, J. Five $1,000 Problems (Update 2017). In *On-Line Encyclopedia of Integer Sequences*. Available online: https://oeis.org/A248380/a248380.pdf (accessed on 6 August 2019).
5. Hoffman, A.J.; Singleton, R.R. On Moore Graphs with Diameters 2 and 3. *IBM J. Res. Dev.* **1960**, *4*, 497–504. [CrossRef]
6. Das, S.; Banerjee, A. Hyper Petersen network: yet another hypercube-like topology. In Proceedings of the 1992 Fourth Symposium on the Frontiers of Massively Parallel Computation, McLean, VA, USA, 19–21 October 1992; IEEE Computer Society Press: Washington, DC, USA, 1992. [CrossRef]
7. Ohring, S.; Das, S. Folded Petersen cube networks: New competitors for the hypercubes. In Proceedings of the 1993 5th IEEE Symposium on Parallel and Distributed Processing, Dallas, TX, USA, 1–4 December 1993; IEEE Computer Society Press: Washington, DC, USA, 1993. [CrossRef]
8. Ohring, S.; Das, S.K. The Folded Petersen Network : A New Communication-Efficient Multiprocessor Topology. In Proceedings of the 1993 International Conference on Parallel Processing—ICPP 93, Syracuse, NY, USA, 16–20 August 1993; Volume 1. [CrossRef]
9. Seo, J.H. Three-dimensional Petersen-torus network: A fixed-degree network for massively parallel computers. *J. Supercomput.* **2011**, *64*, 987–1007. [CrossRef]
10. Seo, J.H.; Kim, J.S.; Chang, H.J.; Lee, H.O. The hierarchical Petersen network: a new interconnection network with fixed degree. *J. Supercomput.* **2017**, *74*, 1636–1654. [CrossRef]
11. Seo, J.H.; Lee, H.; suk Jang, M. Petersen-Torus Networks for Multicomputer Systems. In Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management, Gyeongju, Korea, 2–4 September 2008. [CrossRef]
12. Robinson, R.W.; Wormald, N.C. Numbers of cubic graphs. *J. Graph Theory* **1983**, *7*, 463–467. [CrossRef]
13. Robinson, R.W. Counting cubic graphs. *J. Graph Theory* **1977**, *1*, 285–286. [CrossRef]
14. Meringer, M. Fast generation of regular graphs and construction of cages. *J. Graph Theory* **1999**, *30*, 137–146. [CrossRef]
15. Brinkmann, G. Fast generation of cubic graphs. *J. Graph Theory* **1996**, *23*, 139–149. [CrossRef]
16. Brinkmann, G.; Goedgebeur, J. Generation of Cubic Graphs and Snarks with Large Girth. *J. Graph Theory* **2017**, *86*, 255–272. [CrossRef]
17. Brinkmann, G.; Coolsaet, K.; Goedgebeur, J.; Mélot, H. House of Graphs: A database of interesting graphs. *Discret. Appl. Math.* **2013**, *161*, 311–314. [CrossRef]
18. OEIS Foundation Inc. The On-Line Encyclopaedia of Integer Sequences. Available online: http://oeis.org/ (accessed on 6 August 2019).
19. OEIS Foundation Inc. A068934 in the On-Line Encyclopaedia of Integer Sequences. Available online: http://oeis.org/wiki/User:Jason_Kimberley/A068934 (accessed on 6 August 2019).
20. Vadhan, S.P. The Complexity of Counting in Sparse, Regular, and Planar Graphs. *SIAM J. Comput.* **2001**, *31*, 398–427. [CrossRef]
21. Meringer, M. Structure Enumeration and Sampling. In *Handbook of Chemoinformatics Algorithms*; Chapman and Hall/CRC: London, UK, 2010; pp. 233–267. [CrossRef]
22. Meringer, M.; Cleaves, H.J. Exploring astrobiology using in silico molecular structure generation. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2017**, *375*, 20160344. [CrossRef]
23. Gropp, W.; Lusk, E.; Skjellum, A. (Eds.) *Using MPI: Portable Parallel Programming with the Message Passing Interface*; MIT Press Ltd.: Cambridge, MA, USA, 2014.
24. Deng, Y.; Guo, M.; Ramos, A.F.; Huang, X.; Xu, Z.; Liu, W. Optimal Low-Latency Network Topologies for Cluster Performance Enhancement. *arXiv* **2019**, arXiv:1904.00513v1.
25. Zhang, Y.; Huang, X.; Xu, Z.; Deng, Y. A Structured Table of Graphs with Symmetries and Other Special Properties. *arXiv* **2019**, arXiv:1910.13539v3.
26. OEIS Foundation Inc. A006820 in the On-Line Encyclopaedia of Integer Sequences. Available online: https://oeis.org/A006820 (accessed on 6 August 2019).
27. Larrión, F.; Pizaña, M.; Villarroel-Flores, R. On Self-clique Shoal Graphs. *Discrete Appl. Math.* **2016**, *205*, 86–100. [CrossRef]
28. Cerf, V.G.; Cowan, D.D.; Mullin, R.C.; Stanton, R.G. A partial census of trivalent generalized Moore networks. In *Combinatorial Mathematics III*; Springer: Berlin/Heidelberg, Germany, 1975; pp. 1–27. [CrossRef]

29. Kitasuka, T.; Iida, M. A heuristic method of generating diameter 3 graphs for order/degree problem (invited paper). In Proceedings of the 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS), Nara, Japan, 31 August–2 September 2016. [CrossRef]
30. Mizuno, R.; Ishida, Y. Constructing large-scale low-latency network from small optimal networks. In Proceedings of the 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS), Nara, Japan, 31 August–2 September 2016. [CrossRef]
31. Shimizu, N.; Mori, R. Average shortest path length of graphs of diameter 3. In Proceedings of the 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS), Nara, Japan, 31 August–2 September 2016; pp. 1–6. [CrossRef]
32. Koibuchi, M.; Fujiwara, I.; Fujita, S.; Nakano, K.; T. Uno, T.I.; Kawarabayashi, K. Graph Golf: The Order/degree Problem Competition. Available online: http://research.nii.ac.jp/graphgolf/ (accessed on 6 August 2019).
33. Xu, Z.; Deng, Y. Optimal Routing for a Family of Scalable Interconnection Networks. In Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis (SC'19), Denver, CO, USA, 17–22 November 2019.
34. The Sunway Blue Light in Top 500 List (June 2018). Available online: https://www.top500.org/system/177447 (accessed on 6 August 2019).
35. Gropp, W.; Lusk, E.L.; Sterling, T. (Eds.) *Beowulf Cluster Computing with Linux (Scientific and Engineering Computation)*; The MIT Press: Cambridge, MA, USA, 2003.