A new record of enumeration of regular graphs by parallel processing

Zhipeng Xu[a,b], Xiaolong Huang[b], Fabian Jimenez[c], Yuefan Deng[b,*]

[a]School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong 510275, P.R. China
[b]Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794, USA
[c]Technologies, Empresa Pblica Yachay, Centro de Emprendimiento Innopolis 100115, Ecuador

Abstract

We broke a record, set 8 years ago, in the enumeration of non-isomorphic regular graphs by expanding the sequence of A006820 in OEIS, to achieve the number for 4-regular graphs of order 23 as 429,668,180,677,439, while discovering serval optimal regular graphs with minimum average shortest path lengths (ASPL). The number of 4-regular graphs and the optimal graphs, extremely time-consuming to calculate, result from a method we adapt from GENREG, a classical regular graph generator, to fit for supercomputers' strengths of using thousands of processor cores.

Keywords: Regular graph, parallel computing, dynamical scheduling, OEIS
2010 MSC: 05C30, 68R10

*Corresponding author
 Email addresses: `zhipeng.xu@stonybrook.edu` (Zhipeng Xu), `xiaolong.huang@stonybrook.edu` (Xiaolong Huang), `fjimenez@yachay.gob.ec` (Fabian Jimenez), `yuefan.deng@stonybrook.edu` (Yuefan Deng)

1. Introduction

Analysis of regular graphs for their properties, including eigen-spectra and automorphisms, is a fertile field for discovering and applications in algebraic graph theory [1]. Yet, there are many unsolved problems, e.g., the Conways 99-graph problem [2] and the 57-regular Moore graph [3]. For analysis of interconnection networks, regularity is an essential feature for its direct and useful relationship to the complexity of network implementation, and as such, many regular graphs including the Peterson graph, hypercube graph, and their extensions [4, 5, 6, 7, 8, 9] are widely used to construct interconnection networks.

For 3-regular graphs of order n, Robinson and Wormald [10, 11] presented all of counting results for $n \leq 40$, while pointing out that enumeration for unlabeled k-regular graphs with $k > 3$ is an unsolved problem. Meringer [12] proposed a practical method to construct regular graphs without checking the isomorphism but with a fast test for canonicity. Brankmann [13, 14] developed minibaum and snarkhunter for generating 3-regular graphs. The House of Graphs [15] and the OEIS [16] databased the latest results for counting the regular graphs. Kimberley [17] contributed many results (A068934) to these databases by a package called GENREG developed by Meringer [12].

The 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. For obtaining larger graphs, we extend the GENREG for distributed clusters by using the message passing interface (MPI) [18]. Using the parallel GENREG we developed, we have obtained the following results:

1) filtered all 3-regular graphs up to order 32 with minimum ASPLs;

2) discovered thousands of 4-regular graphs of order 32 with minimum ASPLs;

3) generated the exact counts of 4-regular graphs of order 23 by using the three supercomputer clusters located in the US, China, and Ecuador.

Among our results, the first and second have been applied to the interconnection network research [19] for benchmark relationship of the graph ASPLs to network performance latencies. The third expands n from 22 to 23, for the first time, in the sequence A006820 [20] of OEIS, which is the number of connected 4-regular graphs of order n. Kimberley [20] used the GENREG to enumerate the 4-regular graphs for up to the order 22 in 2011 [21]. This record they set then remains unchallenged until our enumeration for $n = 23$, enabled by our high-performance computing algorithms to advance it a step. In addition to its challenges of pure mathematics, the graph counting problem is the root of topics in reliability, artificial intelligence, reasoning, and statistical physics [22].

2. The enumeration framework and results

2.1. The enumeration function

For enumerating the regular graphs, published packages such as minibaum [13], snarkhunter [14], and GENREG have their own strengths and weaknesses, while GENREG is more general in covering the graph degrees than minibaum and snarkhunter that can only support the 3-regular graphs.

In our parallel computing framework, we designate one node as the master, whose task involves 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 exhausts. 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 the workers to exit.

Our dynamical scheduling strategy makes cores in the cluster busy for useful tasks to allow us 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 $N_{\text{task}} \times 2$, less bandwidth because the message itself is the message count. If we use a dedicated thread for task scheduling, we shrink the scalability and limit the maximum computer system we can reach. Depending on the communication sub-system, the maximum scalable system our current approach can reach is approximately 3,000 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 ASPLs 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 [23] calculated and proved lower bound of ASPL, hence the goal of optimal graph is to find graphs with minimum ASPLs. Usually and thus far, random or heuristic methods or intuitions were resorted to searching for graphs with such desired properties for large networks; Graph Golf [24], a competition of searching for graphs with the minimal diameters and ASPLs, generated many graphs that are very commonly, but asymmetrically. However, we still hope to discover graphs with all desired properties including minimum diameters, ASPLs, symmetry, and robustness but with one disadvantage: smaller graphs. However, these graphs can be adopted in smaller clusters or multiple modules of these clusters or system-on-chip.

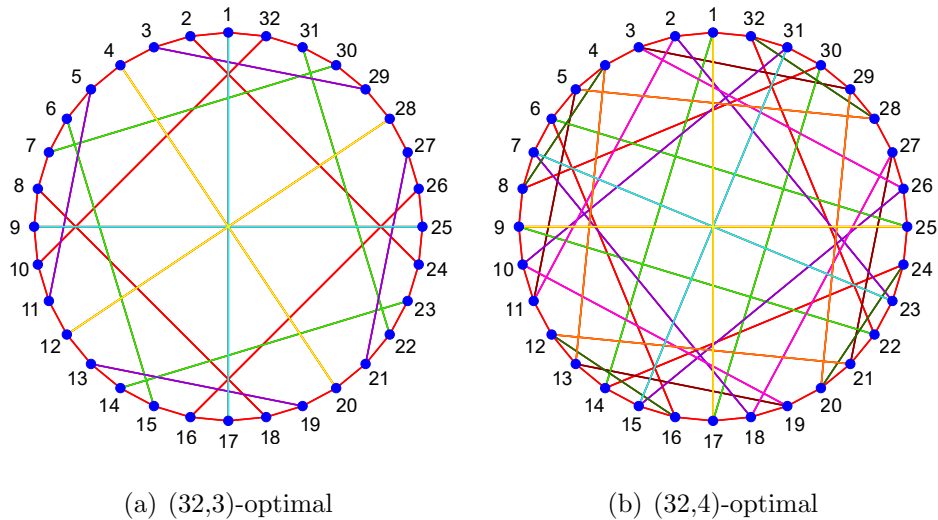(a) (32,3)-optimal   (b) (32,4)-optimal

Figure 1: The optimal graphs used by a group [19] constructing a cluster.

Using this framework, we decomposed the search into 200,000 sub-tasks and completed the exhaustive search of all of 3-regular graphs of order 32 with diameter 4 and minimal ASPL. The program running on the Sunway-Bluelight supercomputer [25] with 80,000 cores for 72 hours and discovered

5

55 graphs with the minimal ASPLs after exhausting all 18,941,522,184,590 possible graphs predicted by Robinson et al. [10] who only enumerated them without finding the graphs with desired properties including minimum diameters and ASPLs. Deng et al. [19] applied one (Fig. 1 (a)) of these graphs to construct a Beowulf [26] cluster, and their benchmark results show that the graphs with the minimal ASPL outpeform other classical topologies.

Fig. 2 shows the distribution of generated graph numbers for all cases, which look like Gaussian distribution with the long tail containing hundreds of millions of graphs and the frequency fluctuates by 4 orders of magnitude. Clearly, the tasks of enumerations vary greatly from graph to graph, and our dynamic task scheduling has eliminated substantial waiting time due to load imbalance.
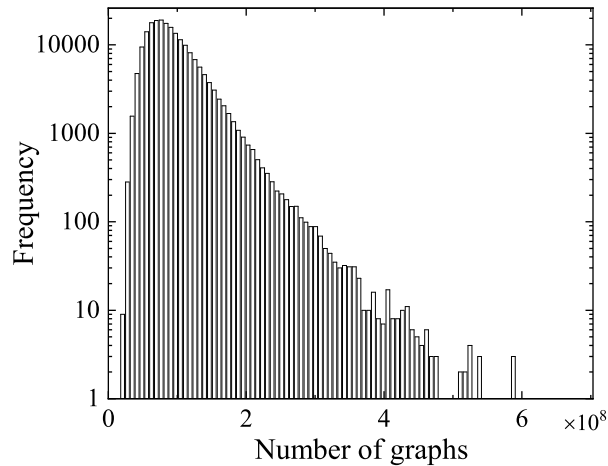

Figure 2: Distribution of loading for each part.

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

6

Table 1: OEIS A006820

Order	Quartics
5	1
6	1
7	2
8	6
9	16
10	59
11	265
12	1,544
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**

$girth \leq 5$. We can still use our software to search for regular graphs with minimum ASPL, as shown in Fig. 1 (b), without confirmation of exhaustiveness. Therefore, we verified the results in Tab. 1 after decomposing the problem into 50,000 sub-tasks and setting the split-level as 12. We manage 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 $n = 23$ is estimated to cost nearly 100 core-years. We ganged three supercomputers, the SeaWulf at Stony Brook University, the Tianhe-1 in Southern China, and the IBM

Table 2: Cost on three clusters

Cluster	Graphs processed	Computing Time (core years)	Speed (graphs per second)
Seawulf	3.7129e+14	66.12	1.7805e+05
Tianhe-1	6.9510e+13	19.53	1.1285e+05
IBM Quinde	2.3599e+13	13.25	5.6469e+04

Quinde in Ecuador, for completing the task. As shown in Tab. 2, the Seawulf with Intel Xeon Gold 6148 processors has the highest efficiency for searching 180,000 graphs per second on a single core, while the Tianhe-1 with Intel Xeon X5670 processors and the IBM Quinde with Power8 processors contributed the rest.

3. Conclusions

Our parallel method adapting the GENREG enables us to complete the research and enumeration on systems of 3000 processor cores. For the first time, using this new approach, we discovered several optimal graphs of order 32 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.

Acknowledgements

8

for computing resources and, M. Meringer of German Aerospace Center, for technical support of GENREG via e-mails.

References

[1] C. Godsil, G. Royle, Algebraic Graph Theory, Springer New York, 2001. `doi:10.1007/978-1-4613-0163-9`.

[2] J. Conway, Five $1,000 problems (update 2017), On-Line Encyclopedia of Integer Sequences (2019).

[3] A. J. Hoffman, R. R. Singleton, On moore graphs with diameters 2 and 3, IBM Journal of Research and Development 4 (5) (1960) 497–504. `doi:10.1147/rd.45.0497`.

[4] S. Das, A. Banerjee, Hyper petersen network: yet another hypercube-like topology, in: [Proceedings 1992] The Fourth Symposium on the Frontiers of Massively Parallel Computation, IEEE Comput. Soc. Press, 1992. `doi:10.1109/fmpc.1992.234949`.

[5] S. Ohring, S. Das, Folded petersen cube networks: new competitors for the hypercubes, in: Proceedings of 1993 5th IEEE Symposium on Parallel and Distributed Processing, IEEE Comput. Soc. Press, 1993. `doi:10.1109/spdp.1993.395482`.

[6] S. Ohring, S. K. Das, The folded petersen network : A new communication-efficient multiprocessor topology, in: 1993 International Conference on Parallel Processing - ICPP 93 Vol1, IEEE, 1993. `doi:10.1109/icpp.1993.175`.

[7] J.-H. Seo, Three-dimensional petersen-torus network: a fixed-degree network for massively parallel computers, The Journal of Supercomputing 64 (3) (2011) 987–1007. `doi:10.1007/s11227-011-0716-z`.

[8] J.-H. Seo, J.-S. Kim, H. J. Chang, H.-O. Lee, The hierarchical petersen network: a new interconnection network with fixed degree, The Journal of Supercomputing 74 (4) (2017) 1636–1654. `doi:10.1007/s11227-017-2186-4`.

[9] J.-H. Seo, H. Lee, M. suk Jang, Petersen-torus networks for multi-computer systems, in: 2008 Fourth International Conference on Networked Computing and Advanced Information Management, IEEE, 2008. `doi:10.1109/ncm.2008.47`.

[10] R. W. Robinson, N. C. Wormald, Numbers of cubic graphs, Journal of Graph Theory 7 (4) (1983) 463–467. `doi:10.1002/jgt.3190070412`.

[11] R. W. Robinson, Counting cubic graphs, Journal of Graph Theory 1 (3) (1977) 285–286. `doi:10.1002/jgt.3190010310`.

[12] M. Meringer, Fast generation of regular graphs and construction of cages, Journal of Graph Theory 30 (2) (1999) 137–146. `doi:10.1002/(sici)1097-0118(199902)30:2<137::aid-jgt7>3.0.co;2-g`.

[13] G. Brinkmann, Fast generation of cubic graphs, Journal of Graph Theory 23 (2) (1996) 139–149. `doi:10.1002/(sici)1097-0118(199610)23:2<139::aid-jgt5>3.0.co;2-u`.

[14] G. Brinkmann, J. Goedgebeur, Generation of cubic graphs and snarks

with large girth, Journal of Graph Theory 86 (2) (2017) 255–272. `doi:10.1002/jgt.22125`.

[15] G. Brinkmann, K. Coolsaet, J. Goedgebeur, H. Mélot, House of graphs: A database of interesting graphs, Discrete Applied Mathematics 161 (1-2) (2013) 311–314. `doi:10.1016/j.dam.2012.07.018`.

[16] OEIS Foundation Inc, The on-line encyclopaedia of integer sequences (2019).
URL `http://oeis.org`

[17] OEIS Foundation Inc, A068934 in the on-line encyclopaedia of integer sequences (2019).
URL `http://oeis.org/wiki/User:Jason_Kimberley/A068934`

[18] W. Gropp, E. Lusk, A. Skjellum (Eds.), Using MPI: Portable Parallel Programming with the Message Passing Interface, MIT Press Ltd, 2014.

[19] Y. Deng, M. Guo, A. F. Ramos, X. Huang, Z. Xu, W. Liu, Optimal low-latency network topologies for cluster performance enhancement, arxiv (2019). `arXiv:http://arxiv.org/abs/1904.00513v1`.

[20] OEIS Foundation Inc, A006820 in the on-line encyclopaedia of integer sequences (2019).
URL `http://oeis.org/A006820`

[21] F. Larrión, M. Pizaña, R. Villarroel-Flores, On self-clique shoal graphs, Discrete Appl. Math. 205 (C) (2016) 86–100. `doi:10.1016/j.dam.2016.01.013`.
URL `https://doi.org/10.1016/j.dam.2016.01.013`

[22] S. P. Vadhan, The complexity of counting in sparse, regular, and planar graphs, SIAM Journal on Computing 31 (2) (2001) 398–427. `doi:10.1137/s0097539797321602`.

[23] V. G. Cerf, D. D. Cowan, R. C. Mullin, R. G. Stanton, A partial census of trivalent generalized moore networks, in: Combinatorial Mathematics III, Springer Berlin Heidelberg, 1975, pp. 1–27. `doi:10.1007/bfb0069540`.

[24] M. Koibuchi, I. Fujiwara, S. Fujita, K. Nakano, T. I. T. Uno, K. Kawarabayashi, Graph Golf: The Order/degree Problem Competition, `http://research.nii.ac.jp/graphgolf/`.

[25] The Sunway Blue Light in Top 500 List (June 2018).
URL `https://www.top500.org/system/177447`

[26] W. Gropp, E. L. Lusk, T. Sterling (Eds.), Beowulf Cluster Computing with Linux (Scientific and Engineering Computation), The MIT Press, 2003.