Diameter-2-critical graphs with at most 13 nodes

Radosavljević, Jovan

Abstract: ***Diameter-2-critical graphs (abbr. D2C) are diameter 2 graphs whose diameter increases by removing any edge. The procedure used to obtain the list of D2C graphs of the order at most 13 is described. This is achieved by incorporating the diameter 2 test and the criticality test into $geng$, the program from the package $nauty$ that generates the list of all non-isomorphic connected graphs. Experiments with the two heuristics in diameter 2 test, which is intensively used during the search, show that it is slightly more efficient to start the test with the largest degree node using BFS algorithm. As an application of the obtained list, the three conjectures concerning the maximum number of edges in D2C graphs were checked for graphs of the order at most 13 and one counterexample was found.***

Index Terms: ***diameter-2-critical graphs, graph diameter, primitive graph.***

1. Introduction

For an undirected graph $G = (V, E)$ (only undirected graphs are considered here) with the set of nodes V and the set of edges E let $\mathrm{d}_G(u, v)$ denote the distance between the nodes $u, v \in V$, i.e. the length of the shortest path connecting them. The order of G is the number $|V|$ of its nodes. If from the context it is clear which graph is considered, the notation $\mathrm{d}(u, v)$ is used. Let $N_k(v) = \{u \mid \mathrm{d}(v, u) \leq k\}$ denote the (open) k-neighborhood of v, $N(v) = N_1(v)$, and let $N_k[v] = \{u \mid \mathrm{d}(v, u) = k\}$ be closed k-neighborhood. The diameter $\mathrm{diam}(G)$ of G is the largest distance $\mathrm{d}(u, v)$ between any two distinct nodes $u, v \in V$. A graph $G = (V, E)$ is diameter-k-critical if $\mathrm{diam}(G) = k$, and for all edges $e \in E$ $\mathrm{diam}(G \setminus e) > k$. The abbreviation D2C is used for diameter-2-critical graphs.

An isomorphism of two graphs $G = (V, E)$ and $H = (U, F)$ is a bijection $\phi : V \longrightarrow U$ such that $(u, v) \in E$ if and only if $(\phi(u), \phi(v)) \in F$. Two graphs G and H are isomorphic if there exists an isomorphism between them. An automorphism of a graph is any isomorphism of a graph to itself. The program $geng$ [2] generates one graph for each class of mutually isomorphic undirected connected graphs, the canonical form of all graphs in the class [7].

D2C graphs are considered by many authors, see [2; 3; 4] for example. There are many open problems related to them, see [5] for example. A possible approach to tackle open problems related to D2C graphs is the analysis of low order graphs. For this purpose, a list of all D2C graphs of order up to 13 was prepared.

So far, the most efficient method for generating a list of D2C graphs is to filter the list of non-equivalent connected undirected graphs obtained by the program $geng$. In the previous paper [10], similar work was done for D2C graphs of order up to 10. The more efficient method used to obtain the list of all non-isomorphic D2C graphs of order up to 13 is described below. As far as the author is aware, there are no other published results of this type.

The rest of the paper is organized as follows. Section 2 describes two variants of the procedure used to generate D2C graphs using the program $geng$ [2], modified by incorporating functions testing the diameter and the criticality. Section 3 presents some statistics concerning D2C graphs of order $n \leq 13$ and the results obtained by checking some conjectures about D2C graphs from [5]. Section 4 draws a

Radosavljević, Jovan is doctoral student at Faculty of Mathematics, Belgrade University; e-mail: jocabnmb@gmail.com

```
N2(A, n, v)
// A[n] - an integer array,
// the adjacency matrix
// of G = (V, E);
// n = |V|;
// v ∈ V - a starting node;
// returns N2v coding N_2[v].
    N2v ← 0
    for i ← 0 to n − 1 do
        if Bit(A_v, i) = 1 then
            N2v ← N2v | A_i
    return N2v
```

Figure 1: 2-neighborhood of v, $N2(A, n, v)$.

conclusion based on the results.

2. Generation of D2C graphs

To generate canonical forms of all undirected graphs, a set of tools from the *nauty* package is used. This package also includes a program for reading and creating files containing listed graphs in supported formats [8]. Technical details about the integration of the diameter-2 test and the criticality test into *geng* are given in the Appendix. Parallelization of the program is discussed in subsection 3.1.

2.1 Diameter-2 test

The diameter-2 test of $G = (V, E)$ consists of several runs of the simplified BFS algorithm (see [9], for example), which determines $N_2(v)$ by traversing G starting from a given node $v \in V$, limiting the depth to 2. Figure 1 shows the code of the algorithm. The adjacency matrix $A' = [a_{ij}]$ of G, where $a_{ij} = 1$ if $ij \in E$ and $a_{ij} = 0$ otherwise, is represented by an integer array A, $A[i] = \sum_{j=1}^{|V|} a_{ij} 2^{|V|-j}$, coding the adjacency matrix rows. Here $\mathrm{Bit}(m, i)$ denotes the bit i of the integer m. The returned integer $N2v = \bigvee_{w \in N(v)} A_w$ codes the set $N_2(v)$. The condition $N_2(v) = V$ is equivalent to $N2v = 2^n - 1$. The complexity of this test is $O(|V|)$.

The condition $\mathrm{diam}(G) = 2$ is equivalent to $|E| < \binom{|V|}{2}$ and $N2(A, n, v) = 2^n - 1$ for all $v \in V$. Diameter-2 test can be improved by choosing some node $v \in V$ and checking first if $N_2(v) = V$ (see Figure 2), and then

```
isd2(A, n, m)
// A[n] - an array representing
// the adjacency matrix
// of G = (V, E);
// n = |V|;
// m = |E|;
// return true iff diam(G) = 2.
    if m = (n choose 2) then return false
    choose a starting node v
    N2v ← N2(A, n, v)
    if N2v < 2^n − 1 then
        return false
    for w ← 0 to n − 1 do
        if Bit(N2v, w) = 1 and
            N2(A, n, w) < 2^n − 1 then
            return false
    return true
```

Figure 2: The diameter 2 test $isd2(A, n, m)$.

- if this condition is not satisfied, then $\mathrm{diam}(G) > 2$;
- otherwise check if $N_2(w) = V$ only for all $w \in N_2[v]$ (if $r, s \in N_1[v]$, then $\mathrm{d}(r, s) \leq 2$ because both r and s are connected with v).

The complexity of this check is $O(|V| \cdot |N_2[v]|)$, suggesting the two different heuristics, controlling choice of the starting node v:

- H1: v is one of the largest degree nodes;
- H2: v is one of the smallest degree nodes.

If there are several nodes with the smallest or the largest degree in G, the first such node in the canonical form of G is taken.

The advantage of H1 is that it minimizes the number of calls to $N2(A, n, w)$. On the other side, if $\mathrm{diam}(G) > 2$, then H2 is expected to increase the probability of detecting that $\mathrm{diam}(G) > 2$ after the first test $N2(A, n, v) = 2^n - 1$ applied to starting node v, thus eliminating all the remaining tests $N2(A, n, w) = 2^n - 1$. Overall complexity of the diameter-2 test is $O(|V|^2)$, because

- if $N_2(v) \neq V$, then the complexity is $O(|V|)$;
- otherwise if $N_2(v) = V$, then the complexity is $O(|V||N_2(v)|)$.

```
isd2c(A, n, m)
// A[n] - an array representing
// the adjacency matrix
// of G = (V, E);
// n = |V|;
// m = |E|;
// return true iff G is D2C.
if not isd2(A, n, m) then
  return false
for u ← 0 to n − 2 do
  for v ← u + 1 to n − 1 do
    if Bit_v(A_u) ≠ 0
    and A_u & A_v ≠ 0 then
        A′ ← array of G \ (u, v)
        if not isd2(A′, n, m − 1) then
          return false
return true
```

Figure 3: D2C test $isd2c(A, n, m)$.

2.2 The criticality test

Checking the diameter-2-criticality of a graph is performed only for graphs G such that $\mathrm{diam}(G) = 2$. Let A denote the integer array coding the rows of the adjacency matrix of G. In terms of used coding, an edge $(u, v) \in E$ is *triangular* if and only if $A_u \wedge A_v \neq 0$. The improved criticality test of G is based on the fact that it is sufficient to check the condition $\mathrm{diam}(G \setminus (u, w)) > 2$ only for triangular edges $(u, w) \in E$, see the code $isd2c(A, n, m)$ in Figure 3. The complexity of $isd2c(A, n, m)$ is $O(|E||V|^2) = O(|V|^4)$.

3. Results

The results of the two heuristics comparison are given in subsection 3.1. Some statistics concerning D2 (for $n \leq 12$) and D2C graphs (for $n \leq 13$) are listed in subsection 3.2. The results of testing some conjectures concerning D2C graphs are given in subsection 3.3.

3.1 The efficiency evaluation

Experiments are conducted in order to compare the two heuristics H1 and H2. The program is run on a computer with the processor AMD Ryzen 7 4800H with 8 cores and 16.0 GB RAM at 2.90 GHz using Ubuntu.

All the nonequivalent connected graphs are divided into the following classes in *isd2* function:

- C0: all the connected graphs;
- C1: the graphs G with $N_2(v) \neq V$ (implying $\mathrm{diam}(G) > 2$), where v is the chosen starting node;
- C2: the graphs G with $N_2(v) = V$ and $\mathrm{diam}(G) > 2$ or $\mathrm{diam}(G) = 1$, and
- D2: the graphs G with $\mathrm{diam}(G) = 2$.

For both heuristics $|C1| + |C2| = |C0| - |D2|$ holds.

As the program *geng* generates graphs of the given order n and with the given number m of edges, in a parallel version of the program processors independently generate the lists of graphs obtained for various m.

The complete lists of D2C graphs of the order up to 11 are obtained in a short time, even without parallelization. For the graphs of the order 12 and 13 the modified version of *geng* is run in parallel. The times in seconds required to obtain all the graphs of diameter 2 of orders less than 13 are shown in Table 1. To obtain the list of D2C graphs of order 13, the parallel execution takes more than a month.

	Sequential		Parallel	
n	H1	H2	H1	H2
9	0.1	0.1		
10	2.9	3.2	0.6	0.7
11	278.8	303.9	41.3	44.6
12	43717.7	47796.0	6147.8	6839.0

Table 1: Execution times of sequential and parallel program versions.

The efficiency of a such parallelization is constrained by the fact that the number of graphs in D2 follow approximately the binomial distribution in terms of the number m of edges. The time required to check all the graphs in C0 with $n = 12$ and $m = 33 = \frac{1}{2}n(n-1)/2$ for H1, H2 is $4136.21s$, $4552.41s$ respectively, while the total time (for all the graphs in C0) is $43717.69s$, $47796.05s$ respectively. As the heuristic H1 proved to be more efficient, it was the only heuristic used for graphs of order 13.

3.2 Some statistics concerning small D2C graphs

The numbers of D2 and D2C graphs are listed in Table 2. As it can be seen, the number of D2 graphs of the order $n \leq 12$ is much larger than the number of D2C graphs. The huge number of D2 graphs of the order 13 were not collected during the one-month program run.

n	$\|D2\|$	$\|D2C\|$
3	1	1
4	4	2
5	14	3
6	59	5
7	373	10
8	4154	30
9	91518	103
10	4116896	519
11	369315249	3746
12	64093257952	40866
13		688118

Table 2: The numbers of D2 and D2C graphs.

The diagram in Figure 4 shows the dependence of $|D2|/|D0|$ and $\log_{10}(|D2C|/|D2|)$ on n. It is well known [1] that for the large order n almost all graphs have diameter 2. Figure 4 shows that the number of diameter-2 graphs is above $1/3$ of the number of all connected graphs for $n \leq 12$. On the other side, the ratio $|D2C|/|D2|$ rapidly diminishes in this range.

The minimum and the maximum number of edges in graphs of diameter 2 are extensively studied. A graph is primitive if there are no two nodes with the same neighborhood. Table 3 shows the minimum and the maximum number of edges in D2C and PD2C (primitive D2C) graphs of order up to 13. Minimal D2C graphs are the stars S_n (the graphs of order n with the $n-1$ edges connecting one node with all the other nodes) and only they reach the lower bound on the number of edges.

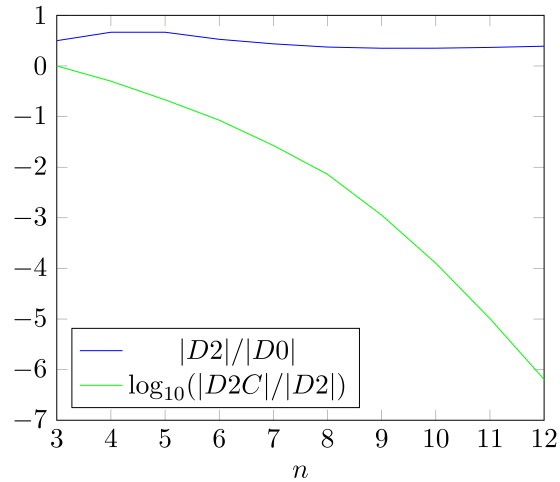

Figure 4: The dependence of $|D2|/|D0|$ and $\log_{10}(|D2C|/|D2|)$ on $n \leq 12$.

3.3 Checking conjectures about D2C graphs

The obtained list of D2C graphs is used to check some conjectures about these graphs for $n \leq 13$ [5].

Plesnik [3] and Murty and Simon [2] independently stated the following conjecture.

Conjecture 1 *A D2C graph of order n has at most $\lfloor \frac{n^2}{4} \rfloor$ edges, with the equality if and only if G is a balanced complete bipartite graph.*

Fan [6] proved Conjecture 1 for $n \leq 24$. This result is partially verified for $n \leq 13$.

A dominating edge in a graph connects a pair of adjacent nodes that have no common non-neighbour. Denote by C_5^+ [4] the family of "expanded 5-cycles", i.e. graphs obtained by replacing three nodes x_1, x_2, x_3 of a 5-cycle by three independent sets X_1, X_2, X_3 of nodes, under the following conditions: (1) x_1, x_2 and x_3 are consecutive on the 5-cycle; (2) $|X_2| \in \{\lfloor \frac{n-2}{3} \rfloor, \lceil \frac{n-2}{3} \rceil\}$, where n is the order of the obtained graph. Dailly [5] stated the following conjecture.

Conjecture 2 *Let G be a non-bipartite D2C graph without a dominating edge. Then G has at most $\lfloor \frac{(n-1)^2}{4} \rfloor + 1$ edges. For sufficiently large n, the equality holds if and only if G belongs to C_5^+.*

The check of conjecture shows that there are no counterexamples for $n \leq 13$.

n	Min m D2C	Min m PD2C	Max m PD2C	Max m D2C
3	2			2
4	3			4
5	4	5	5	6
6	5	8	8	9
7	6	9	10	12
8	7	12	13	16
9	8	14	17	20
10	9	15	20	25
11	10	18	24	30
12	11	20	32	36
13	12	22	36	37

Table 3: Maximal and minimal number of edges in D2C and PD2C graphs of order up to 13.

Conjecture 3 *Let G be a non-bipartite D2C graph of order n. If G is not H_5 [5, Figure 2], then G has at most $\lfloor\frac{(n-1)^2}{4}\rfloor + 1$ edges, with the equality if and only if G belongs to C_5^+ or it is one of the thirteen graphs of order $n \leq 11$ listed in [5, Figure 3].*

The list of exceptions is verified for $n \leq 11$, and the new exception is found, the graph of order 12 with the 32 edges, see Figure 5; here $32 > \lfloor\frac{(n-1)^2}{4}\rfloor + 1 = 31$. The graph contains a dominating edge, which is colored red. The number of edges of this graph appears as the extreme value in Table 3. There are no exceptions of order 13 to Conjecture 3.

4. Conclusion

The list of all D2C graphs of order $n \leq 13$, and the list of all diameter-2 graphs of order $n \leq 12$ are obtained. Two heuristics for diameter-2 testing, concerning the choice of the starting node, were compared. It appears that the choice of the largest degree node as a starting node is slightly more efficient, at least for $n \leq 12$. Using the list of D2C graphs, the three conjectures are checked for $n \leq 13$. The graph of order 12 is found, which is a new exception to Conjecture 3. An interesting question

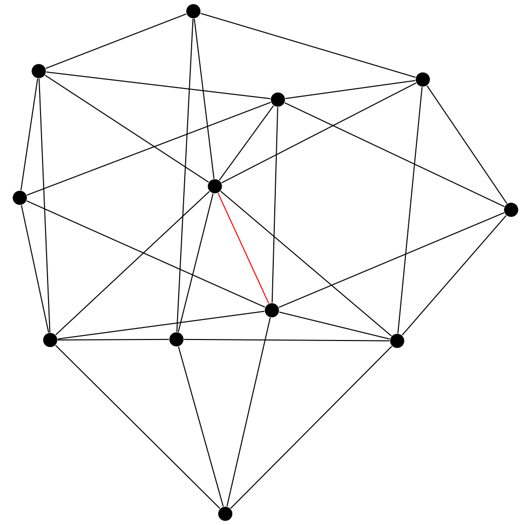

Figure 5: The graph that is a counterexample to Conjecture 3.

is whether the advantage of heuristics H1 also applies to graphs of order greater than 12. Another question is whether more efficient heuristics can be found, which could increase the limit on the order of generated D2C graphs.

5. Acknowledgement

The author thank the anonymous referees for their helpful comments that improved the quality of the manuscript.

References

[1] Bollobás, B. "Graph Theory", Springer-Verlag (1979).
[2] Caccetta, L.; Haggkvist, R. "On diameter critical graphs", Discrete Math. 28(3) (1979), pp. 223–229.
[3] Plesnik, J. "Critical graphs of given diameter", Acta F.R.N. Univ. comenianae Math. 30 (1975), pp. 71–93.
[4] Balbuena, C.; Hansberg, A.; Haynes, T.W.; Henning, M.A. "Total domination edge critical graphs with total domination number three and many dominating pairs", Graphs Combin. 31 (5) (2015), pp. 1163–1176.
[5] Dailly, A.; Foucaud, F.; Hansberg, A. "Strengthening the Murty-Simon conjecture on diameter-2-critical graphs", Discrete Mathematics, Elsevier 342 (11) (2019), pp. 3142–3159.
[6] Fan, G. "On diameter 2-critical graphs", Discrete Math. 67 (3) (1987), pp. 235–240

[7] McKay, B. D. "Practical graph isomorphism", Congr. Numer. (30) (1980), pp. 45—87.
[8] McKay, B. D. and Piperno, A. "Practical Graph Isomorphism, II", J. Symbolic Computation (60) (2013), pp. 94–112.
[9] Cormen, T. H.; Leiserson C. E.; Rivest R. L.; Stein C. "Introduction to Algorithms Third Edition", The MIT Press (2009), pp. 594–602.
[10] Radosavljević, J.; Živković, M. "The List of Diameter-2-Critical Graphs with at Most 10 Nodes", IPSI Transactions (2020).

Appendix: integration of the tests into *geng*

The *geng* program (written in *C*) is a part of the *nauty* package, used to list all non-isomorphic connected graphs that have a given number of nodes and edges [8]. The listed graphs are in *graph6* format, which is supported by the program itself. Here the modification of *geng* used to obtain the list of D2C graphs, that have a given number of nodes and edges, is described.

If the argument *-c* is passed with the graph order n in the call of *geng*, only connected graphs of the order n are generated. When the *main* function of *geng* receives the argument *-c*, the variable *connec1* becomes *TRUE* and the global variable *connec* becomes 1. To generate graphs, the recursive function *genextend* is called. In the function *genextend* the global variable *connec* forces the filtering of connected graphs by calling the function *isconnected*.

In a similar way, the function $isd2c$ implementing criticality test $isd2c$ is incorporated. In the *main* function a check is added if there is an argument *-k* in the program call, indicating that the criticality of graphs should be checked. The new global variable *d2c* is used to store the *-k* argument and to force the *genextend* function to call the function *isd2c*.

The function $isd2c$, implementing the algorithm $isd2c$ above, accepts as arguments *graph *g*, the graph, *int n*, the order of *g*, and *int m*, the number of edges in *g*. The type *graph* is defined in the package *nauty* and is used to represent the graph as an integer sequence (the array A in the call to $isd2c$). In the function $isd2c$ for each triangular edge $e \in E$ the condition $\mathrm{diam}(G \setminus e) = 2$ is checked, by calling the function $isd2$.