Yugoslav Journal of Operations Research
31 (2021), Number 4, 471–481
DOI: https://doi.org/10.2298/YJOR201121014C

COMPLEXITY INDICES FOR THE TRAVELING SALESMAN PROBLEM CONTINUED

Dragoš CVETKOVIĆ
Faculty of Electrical Engineering, University of Belgrade and Mathematical Institute SANU, Belgrade, Serbia
ecvetkod@etf.rs

Zorica DRAŽIĆ
Faculty of Mathematics, University of Belgrade, Belgrade, Serbia
zdrazic@matf.bg.ac.rs

Vera KOVAČEVIĆ - VUJČIĆ
Faculty of Organizational Sciences, University of Belgrade, Belgrade, Serbia
verakov@fon.bg.ac.rs

Received: November 2020 / Accepted: March 2021

Abstract: We consider the symmetric traveling salesman problem (TSP) with instances represented by complete graphs G with distances between cities as edge weights. A complexity index is an invariant of an instance I by which we predict the execution time of an exact TSP algorithm for I. In the paper [5] we have considered some short edge subgraphs of G and defined several new invariants related to their connected components. Extensive computational experiments with instances on 50 vertices with the uniform distribution of integer edge weights in the interval [1,100] show that there exists correlation between the sequences of selected invariants and the sequence of execution times of the well-known TSP Solver Concorde. In this paper we extend these considerations for instances up to 100 vertices.

Keywords: Traveling salesman problem, Complexity index, Concorde TSP Solver, Random instances, Correlation.

MSC: 90C27, 05C80, 62H20.

1. INTRODUCTION

A comprehensive survey of work relevant to measuring the difficulty of instances of combinatorial optimization problems, with the focus on six well-known problems: assignment, traveling salesman, knapsack, bin-packing, graph coloring and timetabling, is given in [11].

Our approach to this problem uses the notion of a complexity index [4], which is defined as follows:

Definition. *Let A be an exact algorithm for solving an NP-hard combinatorial optimization problem C and let I be an instance of C of dimension n. A complexity index of I with respect to A is a real number, computable in polynomial time $P(n)$ from I, by which we can predict (in a statistical sense) the execution time of A for I.*

A complexity index of an instance I is an invariant of I. An invariant of an instance can serve as a complexity index with respect to several algorithms for solving the considered problem but, of course, with different efficiencies. The efficiency of a complexity index has to be statistically estimated using the correlation between the index value and the execution time of the algorithm A. One can use the standard correlation coefficient (Pearson) or the rank correlation coefficient (Spearman). Also, one should specify the number of tested instances and the way they are generated.

The coefficient of linear correlation for two sequences of length s is defined by

$$C_{BC} = \frac{1}{\sqrt{v_B v_C}} \sum_{i=1}^{s} (b_i - m_B)(c_i - m_C) \tag{1}$$

where b_i, c_i, m_B, m_C are values and mean values of the corresponding sequences B and C, respectively, and $v_B = \sum_{i=1}^{s}(b_i - m_B)^2, v_C = \sum_{i=1}^{s}(c_i - m_C)^2$. The Spearman correlation coefficient S_{BC} is defined as linear correlation coefficient between the ranked variables. For each b_i, c_i, their ranks rgb_i, rgc_i are determined and we have

$$S_{BC} = \frac{1}{\sqrt{v_{rg_B} v_{rg_C}}} \sum_{i=1}^{s} (rgb_i - m_{rg_B})(rgc_i - m_{rg_C}) \tag{2}$$

where $rg_B = (rgb_i)$ and $rg_C = (rgc_i)$.

We consider the symmetric traveling salesman problem (TSP) with instances represented by complete graphs G, where distances between cities are edge weights.

Several complexity indices for the TSP with respect to exact branch and bound algorithms have been studied in [8], [6]. For earlier references see [7], [8].

In [5] we have defined several new invariants related to connected components of short edge subgraphs of G as well as to the solution of the assignment problem. Experiments conduced on instances of 50 vertices with the uniform distribution of integer edge weights in the interval [1,100] show that there exists correlation between the sequences of selected invariants and the sequence of execution times of TSP solver Concorde. Another application of the notion of a complexity index is given in [9], where the authors extend the concept of conditioning in integer programming [12] to the concept of a complexity index for the multidimensional knapsack problem (MKP) that corresponds to the minimum and maximum eigenvalue of a Dikin matrix placed in the centre of a polyhedron defined by MKP constraints. The experiments with instances from the OR-library and MIPLIB show medium to strong correlation between the proposed indices and execution times of branch and bound algorithms.

In this paper we extend the analysis considerations from [5] to TSP instances with up to 100 vertices. The results of extensive computational experiments on randomly generated instances show that complexity indices based on the solution of the assignment problem perform in a similar way for 70 and 100 vertices as for 50, while indices based on connected components of short edge subgraphs are inefficient for 70 and 100 vertices . We also investigate whether Generalized Peterson graphs, known as hard instances for the TSP, can be recognized by the assignment problem indices.

2. EXPERIMENTS

We first give some details from the paper [5].

We managed to obtain very good results with the Concorde TSP Solver and instances with 50 vertices. This is the greatest number of vertices ever used in the literature regarding this type of experiments.

For the dimension $n = 50$ we have generated randomly two sets $S(1)$, and $S(2)$, each consisting of hundred TSP instances with integer weights uniformly distributed in interval $[1, 100]$. Since Concorde execution times slightly vary when the same instance is run several times, we recorded the average execution time for five executions of the same instance. The corresponding standard deviations are reasonably small. The average execution time for instances in set $S(1)$ vary between 26 and 872 milliseconds, and for instances in set $S(2)$ from 24 and 1206 milliseconds.

First, we observed the short edge subgraph coming out from the solution of the corresponding assignment problem, obtained from the integer linear programming formulation of the TSP when subtour elimination constraints are omitted. In this subgraph all components are cycles. The following invariants of this subgraph were considered:

I_1: the product of the numbers of vertices of components

I_2: the number of components.

Linear and Spearman correlation coefficients between values of complexity indices I_1 and I_2 for instances in sets $S(1)$ and $S(2)$, and Concorde execution times are given in Table 1.

Second, we observed the short edge subgraph consisting of edges of length 1 or 2. In the theory of random graphs, such subgraphs are random graphs of Erdős-

Table 1: Correlation coefficients related to the assignment problem

	Linear correlation		Spearman correlation	
Complexity indices	Set $S(1)$	Set $S(2)$	Set $S(1)$	Set $S(2)$
I_1	0.24	0.11	0.42	0.33
I_2	0.25	0.23	0.40	0.33

Rényi type with probability $p = 0.02$ for any two vertices to be connected by an edge.

We considered, among other things, the following six invariants for these subgraphs:

J_1: the product of the numbers of vertices of the components

J_2: the product of the squares of the numbers of vertices of the components

J_3: the product of the numbers of vertices in the longest self-avoiding (cycle free) paths in the components

J_4: the product over all components of the product of the number of vertices in the component and the length of the longest self-avoiding path in the component (in the case that the length is equal to 0 we put 1 instead)

J_5: the product over all components of the product of the number of vertices in the component, the number of the longest self-avoiding paths in the component and their length (in the case that the length is equal to 0 we put 1 instead)

J_6: the product over all components of the square of the length of the longest self-avoiding path in the component increased by one.

We see that in all cases there is no contribution of isolated vertices (i.e. trivial components) to the value of the invariant. Suppose that the components of the short edge subgraph are indexed by i and consider the i-th component. Let k_i be the number of vertices, d_i the length of the longest self-avoiding path and S_i the number of such paths in the i-th component. The following formulas hold:

$$J_1 = \prod_i k_i, \quad J_2 = \prod_i k_i^2, \quad J_3 = \prod_i d_i, \tag{3}$$

$$J_4 = \prod_i k_i d_i, \quad J_5 = \prod_i k_i S_i d_i, \quad J_6 = \prod_i (d_i + 1)^2, \tag{4}$$

where $\prod_i$ denotes the product over all non-trivial components of the subgraph.

Linear correlation coefficients between values of complexity indices $J_1 - J_6$ for instances in sets $S(1)$ and $S(2)$, and Concorde execution times vary between 0.46 and 0.60. This represents a moderate but for our purposes very important correlation.

Now, we describe new experiments.

Continuing in the same spirit, we have randomly generated two sets $S(3)$ and $S(4)$ of 100 instances on 70 vertices. Linear and Spearman correlation coefficients between values of complexity indices for these instances and Concorde execution times are given in Table 2.

Table 2: Correlation coefficients for sets $S(3)$ and $S(4)$ with $n = 70$ vertices

	Linear correlation		Spearman correlation	
Complexity indices	Set $S(3)$	Set $S(4)$	Set $S(3)$	Set $S(4)$
I_1	-0.08	0.01	0.32	0.24
I_2	0.12	0.17	0.31	0.27

Indices $J_1 - J_6$ are quite inefficient and therefore, they are not included in this table. In most of the cases, the corresponding correlation coefficients were below or around 0.10. Note that we had difficulties in calculating parameters d_i and S_i because of the large time and space complexity of the applied algorithms.

In the end, we have randomly generated two sets $S(5)$ and $S(6)$ of 100 instances on 100 vertices to see whether the effects appearing for 50 vertices are also present in this case. Indices I_1 and I_2 perform in a similar way (see Table 3). Indices $J_1, \ldots, J_6$ appear to be inefficient also for this dimension. This is in accordance with the conclusion, stated in [5], that correlation coefficients of the complexity indices decrease as the number of vertices increases (see also Section 3 here).

We turn to the indices based on the assignment problem from Table 3.

Table 3: Correlation coefficients for sets $S(5)$ and $S(6)$ with $n = 100$ vertices

	Linear correlation		Spearman correlation	
Complexity indices	Set $S(5)$	Set $S(6)$	Set $S(5)$	Set $S(6)$
I_1	0.30	0.15	0.11	0.11
I_2	0.23	0.04	0.15	0.17

Motivated by the efficiency of indices I_1 and I_2, we extended the set of invariants related to the short edge subgraph coming out from the solution to the corresponding assignment problem:

I_3: the number of vertices in the largest component,

I_4: the number of vertices in the largest component divided by the total number of components,

I_5: the reciprocal of I_1,

I_6: the number of vertices in the largest component divided by the number of vertices in the second largest component. If the number of components is equal to one, we use I_3,

I_7: the number of components multiplied by I_6.

Suppose that C_1,...,C_m are the components of the short edge subgraph obtained from the solution to the assignment problem such that $k_1 \geq k_2 \geq ... \geq k_m$, where k_i represents the number of vertices in the component C_i, $1 \leq i \leq m$. The following formulas hold:

$$I_3 = k_1, \quad I_4 = \frac{I_3}{m} = \frac{k_1}{m}, \quad I_5 = \frac{1}{I_1} = \frac{1}{\prod\limits_{i=1}^{m} k_i},$$

$$I_6 = \begin{cases} \frac{k_1}{k_2}, & \texttt{if } m > 1, \\ k_1, & \texttt{if } m = 1 \end{cases}, \quad I_7 = m \cdot I_6,$$

Linear and Spearman correlation coefficients between values of these complexity indices for instances in sets $S(1)$ - $S(6)$ and Concorde execution times are given in Table 4.

We see that complexity indices of this type are more stable, as the number of vertices increases, than the other considered indices.

3. BEHAVIOR OF COMPLEXITY INDICES AS THE NUMBER OF VERTICES INCREASES

An important empirical conclusion follows from all experiments (both current and the past ones): *Correlation coefficients for all considered complexity indices decrease as the number of vertices increases.*

Most of the complexity indices considered are based on some kind of short edge subgraphs. However, we have to distinguish between indices based on the solution to an optimization problem (e.g. the assignment problem) and those based on subgraphs which literally contain edges with the smallest weights.

Complexity indices based on the assignment problem appear to be stable as the number of vertices increases as we can see from Table 4 and the previous tables. Better performance of these indices is based on the fact that the corresponding short edge subgraph is not defined at the beginning. It is rather determined in the course of an optimization problem. In this way, such subgraphs could contain relatively long edges.

Similar effects appear with complexity indices based on the minimum spanning tree [7], [8].

The conclusion on decreasing complexity indices of the second type of short edge subgraphs, i.e., the subgraphs which literally contain edges with the smallest weights, is contained in the previous literature, especially in [5]. Indeed, if a short edge subgraph S, on which a complexity index is based, contains no more than cn edges (n the number of vertices of the TSP instance, c a constant), then the ratio

Table 4: Correlation coefficients for the new complexity indices related to solutions to the assignment problem for instances with $n = 50, 70, 100$ vertices

	Linear correlation		Spearman correlation	
Complexity indices	Set $S(1)$	Set $S(2)$	Set $S(1)$	Set $S(2)$
I_3	-0.36	-0.24	-0.42	-0.32
I_4	-0.32	-0.23	-0.39	-0.31
I_5	-0.33	-0.23	-0.39	-0.32
I_6	-0.34	-0.22	-0.45	-0.29
I_7	-0.38	-0.23	-0.45	-0.25
	Set $S(3)$	Set $S(4)$	Set $S(3)$	Set $S(4)$
I_3	-0.15	-0.20	-0.32	-0.25
I_4	-0.26	-0.25	-0.32	-0.23
I_5	-0.27	-0.26	-0.32	-0.23
I_6	-0.25	-0.27	-0.33	-0.26
I_7	-0.21	-0.28	-0.34	-0.27
	Set $S(5)$	Set $S(6)$	Set $S(5)$	Set $S(6)$
I_3	-0.20	-0.21	-0.10	-0.10
I_4	-0.22	-0.15	-0.11	-0.11
I_5	-0.22	-0.15	-0.10	-0.10
I_6	-0.22	-0.16	-0.10	-0.11
I_7	-0.22	-0.19	-0.07	-0.10

$cn/\binom{n}{2} = 2c/(n-1)$ (the part of all the edges which is contained in S) goes to 0 as n increases. Hence, the information on the structure of the instance contained in S is expected to disappear.

The assumed upper bound cn for the number of edges in a short edge subgraph means, in fact, that mean value of this number in a random graph of considered type can be majorized in this way. A simplified example is the case when the short edge subgraph is formed by taking just n shortest edges.

The case of invariants J_1 - J_6 is very specific. The corresponding short edge subgraphs contain in average 2% of all edges, independently of the number of vertices. The explanation of the fact that the efficiency of these indices go down as the number of vertices increases should be looked for in the theory of random graphs. Some details in this direction are given in [5]. Here we mention that the corresponding short edge subgraphs for the cases $n = 50, 70, 100$ contain in average 24.5 , 48.3 and 99 edges, respectively.

4. HARD CASES

The paper [1] describes several hard cases for the Traveling salesman problem, in particular, Generalized Petersen graphs (GPGs).

We have investigated whether our technique with complexity indices can recognize GPGs as difficult instances, i.e. instances which require a long time to be solved.

The Generalized Petersen graph family was first studied by Coxeter in 1950 [3] and was named in 1969 by Watkins [13]. In Watkin's notation the Generalized Petersen graph $GP(p,k)(p \geq 3; 1 \leq k < p/2)$ has $n = 2p$ vertices and $m = 3p$ edges, with vertex set $V = \{u_i, v_i | 0 \leq i \leq p-1\}$, and edge set $E = \{\{u_i, u_{i+1}\}, \{u_i, v_i\}, \{v_i, v_{i+k}\} | 0 \leq i \leq p-1\}$, where vertex indices are taken modulo p. Certain choices of p and k result in graphs with special characteristics, in particular, graphs with a high degree of symmetry and known number of Hamiltonian cycles. The following theorem from [10] gives the number of Hamiltonian cycles of $GP(p,2)$.

Theorem. *Let F_p denote the Fibonacci number defined by $F_0 = 0, F_1 = 1, F_p = F_{p-1} + F_{p-2}$, $p \geq 2$. For $p \geq 3$, the number of Hamiltonian cycles in $GP(p,2)$ depends on the congruence class of p (mod 6).*

For $p \equiv 0$ or $2(mod\ 6)$ there are $2F_{p/2} - 2F_{p/2-2} - 2$ Hamiltonian cycles.

For $p \equiv 1(mod\ 6)$ there are n Hamiltonian cycles.

For $p \equiv 3(mod\ 6)$ there are 3 Hamiltonian cycles.

For $p \equiv 4(mod\ 6)$ there are $p + 2F_{p/2+2} - 2F_{p/2-2} - 2$ Hamiltonian cycles.

For $p \equiv 5(mod\ 6)$ there are 0 Hamiltonian cycles.

Note that for $p \equiv 5(mod\, 6)$, graph $GP(p,2)$ is non-Hamiltonian, but it has been proved in [2] that it is hypohamiltonian and hence, the addition of any edge to the graph will introduce Hamiltonian cycles. It has been noted in [1] that classes $GP(p,2)$ with $p \equiv 1(mod\ 6), p \equiv 3(mod\ 6)$, and $p \equiv 5(mod\ 6)$ constitute very difficult examples for most TSP algorithms, including Concorde, Lin-Kernighan and Snakes and Ladders, even when the size of the graph is small. That motivated the investigation of a possible application of complexity indices to class $GP(p,2)$.

In our experiments the distance matrix that corresponds to $GP(p,2)$ is formed as follows: the distance between any two cities i and j is set equal to 0 if the edge (i,j) exists in $GP(p,2)$ and equal to 1 otherwise, diagonal entries being equal to 0. The length of the optimal tour will be 1 for $p \equiv 5(mod\ 6)$ since $GP(p,2)$ is hypohamiltonian and the optimal tour has to contain one edge with weight 1. In all the other cases, the optimal tour of length 0 will be contained in $GP(p,2)$. Since Concorde execution times slightly vary when the same instance is run several times, we recorded the average execution time for five executions of the same instance. Table 5 gives the Concorde average execution times in seconds for $GP(p,2)$ for $25 \leq p \leq 50$.

Table 5: Performance of Concorde on Generalized Petersen graphs.

n	p		t
50	25	1 (mod 6)	0.052
52	26	2 (mod 6)	0.056
54	27	3 (mod 6)	0.677
56	28	4 (mod 6)	0.046
58	29	5 (mod 6)	80.830
60	30	0 (mod 6)	0.056
62	31	1 (mod 6)	0.060
64	32	2 (mod 6)	0.056
66	33	3 (mod 6)	7.784
68	34	4 (mod 6)	0.282
70	35	5 (mod 6)	609.536
72	36	0 (mod 6)	0.064
74	37	1 (mod 6)	12.218
76	38	2 (mod 6)	0.058
78	39	3 (mod 6)	121.098
80	40	4 (mod 6)	0.068
82	41	5 (mod 6)	2400*
84	42	0 (mod 6)	0.068
86	43	1 (mod 6)	10.006
88	44	2 (mod 6)	0.084
90	45	3 (mod 6)	319.743
92	46	4 (mod 6)	0.142
94	47	5 (mod 6)	6000*
96	48	0 (mod 6)	0.080
98	49	1 (mod 6)	150.294
100	50	2 (mod 6)	0.114

Sign '*' next to a number means the reported failures due to the inability of Concorde to find the optimal tour within a designated number of seconds.

Table 5 shows that among the difficult classes the hardest is $GP(p,2)$ with $p \equiv 5(mod\ 6)$, followed by $GP(p,2)$ with $p \equiv 3(mod\ 6)$ and $p \equiv 1(mod\ 6)$. According to the Theorem these classes are characterized by a small number of Hamiltonian cycles (0 for $p \equiv 5(mod\ 6)$, 3 for $p \equiv 3(mod\ 6)$ and n for $p \equiv 1(mod\ 6)$, which is a probable cause of difficulties for Concorde.

Our experiments are related to TSP instances on 50, 70, and 100 vertices. However, GPG instances on 50 and 100 vertices are easy, and therefore we have to concentrate on the case of $n = 2p = 70$ vertices, where we do have difficult instances.

To check whether our complexity indices can recognize $GP(35,2)$ as a difficult instance, we have calculated invariants I_1 - I_7 for twenty corresponding TSP instances generated randomly in the following way. The distance matrices are obtained by replacing zeros in the adjacency matrix of Generalized Petersen graph with randomly generated numbers from the interval $[2, 100]$.

The distribution of calculated invariants in the intervals between lower and upper bounds for these invariants in both series $S(3)$ and $S(4)$ is almost uniform. This shows that complexity indices based on the assignment problem are not strong enough to recognize the Generalized Petersen graph on 70 vertices as a difficult instance for TSP.

If we use the interval $[1, 100]$ instead of $[2, 100]$ in the generating process, we get easy cases! Namely, with high probability we generate at least one number 1 and the corresponding edge closes a Hamiltonian cycle with edges of $GP(35, 2)$ since $GP(35, 2)$ is hypohamiltonian. This cycle of length 70 will be quickly found by Concorde, while in the case of interval $[2, 100]$, the optimal length is equal to 71 since, almost always, there exists an edge of length 2 which has to appear in the optimal Hamiltonian cycle.

We have also generated twenty such easy cases of $GP(35, 2)$ and calculated the corresponding invariants I_1 - I_7. It turns out that hard and easy cases of $GP(35, 2)$ can readily be distinguished by these complexity indices.

In order to make this statement more formal, we have formed a set of 10 hard and 10 easy instances of $GP(35, 2)$ and calculated linear correlation coefficient between the sequences of Concorde execution times (varying from 435,35 to 693,02 seconds for hard instances and from 0,156 to 0,236 seconds for easy ones) and the values of complexity index I_2 . The value of the coefficient of about 0.60 for this correlation enables the statement that hard and easy instances of $GP(35, 2)$ can be distinguished by index I_2. Similar results are obtained for other considered indices based on the assignment problem.

The obtained values of the linear correlation coefficients read: 0.2822 for I_1, 0.6085 for I_2, -0.3853 for I_3, -0.4874 for I_4, -0.3872 for I_5, -0.3300 for I_6, -0.1812 for I_7.

Calculated values of complexity indices for twenty hard and twenty easy instances of $GP(35, 2)$, as well as above mentioned upper and lower bounds, for indices in sets $S(3)$ and $S(4)$ are not presented here.

5. CONCLUSION

We show that complexity indices for TSP based on the assignment problem have moderate correlation coefficients but remain stable in all three considered cases: 50, 70, and 100 vertices. We see that in most of the cases index, I_2 (the number of components in the solution of the assignment problem) behaves well.

Acknowledgement: The work on this paper was supported by the project F-159 of the Serbian Academy of Sciences and Arts.

REFERENCES

[1] Baniasadi, P., Ejov, V., Haythorpe, M., and Rossomakhine, S., ”A new benchmark set for traveling salesman problem and hamiltonian cycle problem”, *arXiv:1806.09285*, (2018).

[2] Bondy, J.A., ”Variations on the Hamiltonian theme”, *Canadian Mathematical Bulletin*, 15 (1) (1972) 57–62.

[3] Coxeter, H.S.M., ”Self-dual configurations and regular graphs”, *Bulletin of the American Mathematical Society*, 56 (5) (1950) 413–455.

[4] Cvetković, D., ”Complexity indices for the travelling salesman problem and data mining”, *Transactions of Combinatorics*, 1 (1) (2012) 35–43.

[5] Cvetković, D., Čangalović, M., Dražić, Z., and Kovačević-Vujčić, V., ”Complexity indices for the traveling salesman problem based on short edge subgraphs”, *Central European Journal of Operations Research*, 26 (3) (2018) 759–769.

[6] Cvetković, D., Čangalović, M., and Kovačević-Vujčić, V., ”Complexity indices for the travelling salesman problem based on a semidefinite relaxation”, *Zbornik radova XXVI Simpozijuma za operaciona istrazivanja*, SYM-OP-IS 1999, Beograd, (1999) 177–180.

[7] Cvetković, D., Dimitrijević, V., and Milosavljević, M., ”A survey of some non-standard traveling salesman problems”, *Yugoslav journal of operations research*, 2 (2) (1992) 163–185.

[8] Cvetković, D., Dimitrijević, V., and Milosavljević, M., *Variations on the Travelling Salesman Theme*, Libra produkt, Beograd, 1996.

[9] Derpich, I., Herrera, C., Sepúlveda, F., and Ubilla, H., ”Complexity indices for the multidimensional knapsack problem”, *Central European Journal of Operations Research*, 29 (2) (2021) 589–609.

[10] Schwenk, A.J., ”Enumeration of Hamiltonian cycles in certain generalized Petersen graphs” *Journal of Combinatorial Theory, Series B*, 47 (1) (1989) 53–59.

[11] Smith-Miles, K., and Lopes, L., ”Measuring instance difficulty for combinatorial optimization problems”, *Computers & Operations Research*, 39 (5) (2012) 875–889.

[12] Vera, J.R., and Derpich, I., ”Incorporating condition measures in the context of combinatorial optimization”, *SIAM Journal on Optimization*, 16 (4) (2006) 965–985.

[13] Watkins, M.E., ”A Theorem on Tait Colorings with an Application to the Generalized Petersen Graphs”, *Journal of Combinatorial Theory*, 6 (2) (1969) 152–164.