The Alternative of Sensor Placement in Multi-Story Buildings (Representation of Generalized Petersen Graphs with Certain Operation) through the Metric Dimension Approach

Asmiati[1]*, Akmal Junaidi[2], Ahmad Ari Aldino[3], and Arif Munandar[4]

[1-]Mathematics Department, Faculty of Mathematics and Natural Sciences, Universitas Lampung
Jln. Prof. Dr. Ir. Sumantri Brodjonegoro No.1, Bandar Lampung 35141, Indonesia
[2,4]Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Lampung
Jln. Prof. Dr. Ir. Sumantri Brodjonegoro No.1, Bandar Lampung 35141, Indonesia
[3]Informatics Department, Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia
Jln. ZA. Pagar Alam No. 9 -11, Bandar Lampung 35132, Indonesia
[1]asmiati.1976@fmipa.unila.ac.id; [2]akmal.junaidi@fmipa.unila.ac.id;
[3]aldino@teknokrat.ac.id; [4]arif.munandar1574@students.unila.ac.id

Received: 30th Maret 2021/ **Revised:** 17th September 2021/ **Accepted:** 20th September 2021

How to Cite: Asmiati, Junaidi, A., Aldino, A. A., & Munandar, A. (2022). The Alternative of Sensor Placement in Multi-Story Buildings (Representation of a Generalized Petersen Graph with Certain Operations) through the Metric Dimension in Graph Approach. *ComTech: Computer, Mathematics and Engineering Applications, 13*(2), 123−132. https://doi.org/10.21512/comtech.v13i2.7268

***Abstract* -** In a public facility or private office where many people can get together, a fire detection device is a mandatory tool as an emergency alarm in the facility. However, the expense of the installation of the device is a troublesome matter. So, optimization is needed to minimize the number of these devices. The way to implement is to select the appropriate position to place the devices in public facilities. The research discussed the placement of the sensors in multi-story buildings. The multi-story buildings could be represented as cube composition graphs with the number of rooms, and the connectivity between the floor and its rooms was equal. The concept of this multi-story building was modeled into a generalized Petersen graph where a vertex represented a room, and an edge was the connectivity of rooms. The basis obtained on that metric dimension was represented as a sensor placed on the building. Then, the optimization of device placement was seen as determining the metric dimensions of the Petersen graph. In the research, the alternative sensor placements were computed using the graph metric dimension approach implemented in Python. The research successfully implements the metric dimension of $1P_{4,1}$ to $4P_{4,1}$ using Python code to obtain the alternative of its basis. A basic alternative indicates the location of the device placement like fire detectors, network access points, or other sensors inside a building.

Keywords: metric dimension, resolving set, generalized Petersen graph, python programming

I. INTRODUCTION

The application of metric dimensions plays a role in robotic navigation, optimization of the placement of threat detection sensors, and classification of chemical compound data (Saenpholphat & Zhang, 2004). In the field of chemistry, the bonding of molecules of an element or chemical compound can be described as a graph. The recording of the bonds of chemical molecules is done by classifying the structure of the graphs depicted. Problems will arise if the graph depicted is a large one. The bigger the chemical graph is, the bigger the data need attention. The concept of metric dimensions is used to determine the molecules that form the basis of these large chemical graphs to facilitate the data collection process. Furthermore, the data entered are from these base molecules.

Moreover, in an important and public facility, a threat detection tool is obligatory as a form of security provision in the facility. However, spending on procuring and installing such tools is also a complicated problem. Therefore, optimization is required in determining the number of these detection tools. One way to achieve this goal is to pay attention to the selection of the placement of such tools in critical and public facilities. This problem can be modeled into a graph by assuming the room in the facility as a vertex and the connection between rooms as an edge. Meanwhile, the optimization of the sensor placement is seen as determining the metric dimensions of the graph representing the substantial or public facility.

The minimum determination of sensors and their placement rules on multi-story buildings is an interesting issue to review. The success of the research can be used to decide the location of the sensors and reduce the installation budget. In the research, alternative sensor arrangements are established by Python programming.

The definition of the metric dimension of a graph given in the following is taken from Chartrand, Eroh, Johnson, and Oellermann (2000). A graph is a set pair *(V, E)* where V(Q) is an infinite set of objects called a vertex. Then, E(Q) is a set of unordered pairs of different vertices called edges. Each edge connects exactly two vertices, and each vertex can have many edges connecting them with the other. Suppose vertex u and v are elements of vertices on a connected graph Q. The distance $d(a,b)$ is the length of the shortest path between vertex a and b on Q. A sequenced set $W = (u_1, u_2, ..., u_k)$ of vertices on connected graph Q and vertex v on Q, $r(W) = (d(v, u_1), d(v, u_2), ..., d(v, u_k))$ shows the representation of v on W. Set of W is called resolving set of Q. Every vertex on Q has a different representation. Then, elements of the resolving set are called basis. The minimum cardinality of the resolving set is called the metric dimension of Q, notated by *dim*(Q).

In general, determining the metric dimension of a graph is a complex problem. So far, no effective algorithm can be used to determine the metric basis for any graph. It is due to the various shapes and structures of the graphs. However, some criteria, constraints, and the metric dimension of certain classes can already be demonstrated. For example, Chartrand et al. (2000) determined the metric dimension of a circle, path, and complete graphs. Meanwhile, Bača, Baskoro, Salman, Saputro, and Suprijanto (2011) studied the metric dimensions of several classes of regular graphs. Hernando, Mora, Pelayo, Seara, and Wood (2010) succeeded in determining all classes of graphs of order n and diameter d and having metric dimensions of n-d. Similarly, Simanjuntak, Uttunggadewa, and Saputro (2013) succeeded in determining the metric dimension of a graph composition of products with stars and obtained the metric dimensions of an amalgamation graph and Saputro et al. (2013) for lexicographic product of graphs. Next, Dudenko and Oliynyk (2017) determined a unicyclic graph of metric dimension 2. In the same year, Saputro, Mardiana, and Purwasih (2017) obtained metric dimension for comb product.

The concept of metric dimensions is developing rapidly, one of which is about local multiset dimension introduced by Alfarisi, Dafik, Kristiana, and Agustin (2019). For example, Adawiyah, Agustin, Prihandini, Alfarisi, and Albirri (2019) determined the local multiset dimension of m-shadow graph. Meanwhile, Adawiyah, Prihandini, Albirri, Agustin, and Alfarisi (2019) analyzed it for unicyclic graph. Next, Okamoto, Phinezy, and Zhang (2010) obtained local metric dimension of graph. Next, Susilowati, Sa'adah, Fauziyyah, Erfanian, and Slamin (2020) searched for dominant metric dimension of graphs.

One of the well-known graph classes is the Petersen graph. The Petersen graph is very popular to study because of its uniqueness as a counter-example and has interesting properties. Generalized Petersen graph of $P_{n,m}$, $n \geq 3$ and $1 \leq m \leq [n-1/2]$ consists of $n-$outer cycle $u_1, u_2, ..., u_n$ set of n radius $u_1, v_1, 1 \leq i \leq n,$ and n edge v_1, v_{1+m} $1 \leq i \leq n$ with an index taken from the modulo n.

Suppose there is a Generalized Petersen graph $P_{n,m}$. Some vertices on the outer cycle $u_i, i \in [1,n]$ for Generalized Petersen graph $y^{th}, y \in [1,x]$, $x \geq 1$ is denoted by u_i^y, while some vertices on the inner cycle $v_i, i \in [1,x]$ for Generalized Petersen graph, $y^{th}, y \in [1,x]$, $x \geq 1$ are denoted by v_i^y. Generalized Petersen graph $xP_{n,m}$ is obtained from $x \geq 1$ graf $P_{n,m}$. Each vertex on the outer cycle $u_i^y, i \in [1,n], y \in [1,x]$ is connected by an edge $u_i^y u_i^{y+1}, y \in [1, x-1], x \geq 2$. Then, the Petersen graph, $xP_{n,m}$ is referred to as a multi-story building with x number of floors.

In the research, it reviews the placement of sensors in multi-story buildings. A multi-story building can be presented as a graph of cube composition with location, and the number of rooms on each floor is equal. The concept of this multi-story building can be expressed as a modified Petersen graph. Then, the basis obtained on that metric dimension can be represented as a sensor placed on the building. Last, Python programming is used to enumerate all alternatives of the sensor located in the building.

II. METHODS

The steps taken to determine sensor placement of generalized placement on generalized Petersen graph, (multi-story building) for some value of x is as $xP_{n,1}$ follows. First, it constructs the basis of the generalized Petersen graph, $xP_{n,1}$ manually. As examples, Imran, Siddiqui, and Naeem (2018) obtained the metric dimension of the generalized Petersen graph $P_{n,1}$ of 2 for $n \geq 3$ odd number and 3 for even number. Asmiati, Aldino, Notiragayu, Zakaria, and Anshori (2020) determined the metric dimension of the generalized Petersen graph $xP_{n,1}$ of x+1 for n odd number and x+2 for n even number.

Next, the research gives an example of a basis or sensor placement on a generalized Petersen graph, $xP_{n,1}$ (multi-story building), for some value of x. The first is for $x = 1$, basis set $W = \{u_1^1, u_h^1, u_n^x\}$, with $h = \frac{n+2}{2}$. The representation of vertices graph $P_{n,1}$ can be seen in Table 1.

For floor $x=4$, basis set $W=\{u_1^1,u_h^1,u_n^1,\dots,u_n^4\}$, with $h=\frac{n+2}{2}, m=\{2,\dots,x\}$ is the value of x from each vertex on the basis set. The representation of vertices graph $4P_{n,1}$ with basis set W is shown in Table 2 (see Appendix).

Second, the research determines the input and parameter to get a potential basis. The algorithm is implemented in Python by following the results of Asmiati et al. (2020). This program needs two inputs consisting of s and n variables that respectively represent a number in $xP_{n,1}$. These inputs are managed by the computer program to generate the list of basis alternatives. The algorithm is initialized by setting the basis of $(x + 1)$ vertices for n equal to an odd number or $(x + 2)$ vertices for n equal to an even number. These vertices represent the number of basis vertices for the Petersen graph, $xP_{n,1}$. The main algorithm is given as follows.

```
Get the input of x and n of $xP_{n,1}$ from user;
Initialization of parameters of the graph;
       Draw the graph;
if n is odd
       set num_of_basis = x+1
else
       set num_of_basis = x+2;
basis = {};

repeat:
Searching potential_basis={$v_i$,$v_j$,$v_k$,$v_l$};
basis_criteria(potential_basis);
if potential_basis meet basis_criteria
       put potential_basis to basis;
else
       if rest vertices is not empty
              do repeat
       else
              break;
print basis;
```

Table 1 Representation of Vertices Graph $P_{n,1}$ with Basis Set W

Vertex	**Taken Basis**		
	u_1^1	u_h^1	u_n^1
u_1^1	0	$h-1$	1
u_2^1	1	$h-2$	2
u_3^1	2	$h-3$	3
$\vdots$			
u_h^1	$h-1$	0	$h-2$
u_{h+1}^1	$h-2$	1	$h-3$
$\vdots$			
u_n^1	1	$n-h$	0
v_1^1	1	h	2
v_2^1	2	$h-1$	3
v_3^1	3	$h-2$	4
$\vdots$			
v_h^1	h	1	$h-1$
v_{h+1}^1	$h-1$	2	$h-2$
$\vdots$			
v_n^1	2	$h-1$	1

Third, it determines the basis candidate and visualizes the graph. The sub-algorithm for searching potential_basis is conducted by using the brute force approach. In this approach, the program will explore all possible compositions of the $(x + 1)$ or $(x + 2)$ vertices as potential basis. Each potential_basis is examined carefully. The basis_criteria is applied to decide whether the potential_basis will be incorporated as the set of basis or not. It works as follows.

```
Get the basis_candidate from the main
program.
Compute their distances to all rest
vertices.
Store their shortest distances.
Inspect these distances.
If representation of all vertices to the
basis candidate is unique
       the criteria match
       return the basis_candidate
else
       break
```

The sub-algorithm of basis_criteria receives parameter potential_basis from the main algorithm. The parameter is transformed as a local parameter called basis_candidate for further processing in this sub-algorithm. The distance between basis_candidate and the rest of the vertices is computed, and the shortest distances are stored. The criteria for the basis match whenever a uniqueness in their distances is found. In this case, the basis_candidate returns to the main program as a basis. Otherwise, the basis candidate is dropped, and the execution of this sub-algorithm is released. Then, the control of the running program is handed over to the main program.

Next, the process searches for the next potential_basis in the main algorithm. The checking of each potential_basis is performed on basis_criteria. The flow execution ends after running out all composition of vertices in the graph. The final output is the list of the whole basis alternatives for the Petersen graph, $xP_{n,1}$.

Last, it runs the program to get an alternative placement of basis or sensors. The result shows the visualization of the Petersen Graph as a representation of multi-story buildings.

III. RESULTS AND DISCUSSIONS

The research uses Python code to process the result using the hardware and software, as shown in Table 3. In the research, the results are generated for the Petersen graph of type $1P_{4,1}$, $2P_{4,1}$, $3P_{4,1}$, and $4P_{4,1}$. The results have been collected to compare the running time and number of the basis for each type of Petersen Graph. This restriction is applied due to the system's limitation for running the code. The execution of a more complex type of the Petersen graph requires more resources than such a system.

The summary of the execution time and the number of the basis for each type of Petersen graph is reported in Table 4. The $1P_{4,1}$ has 32 bases that can be obtained in less than one second. When the layer of the Petersen graph is added on one layer to $2P_{4,1}$, the computation time is three times, and 989 bases are successfully identified. A significant increase in execution time and the number of these bases occurs for the Petersen graph in the three layers $3P_{4,1}$. In this case, the execution time becomes 91 seconds to formulate 26.106 bases. Both the time and the basis escalate around 30 folds compared to the former output. The number of bases rises significantly for $4P_{4,1}$ with a total time of 3.107 seconds producing 638.597 bases. Each one-layer expansion of $P_{4,1}$ will increase the number of bases by about 25 folds with an exponential rate in time. It makes sense since the code runs a brute-force approach.

The drawing of the $1P_{4,1}$ in Table 4 can be observed in Figure 1. Some possible bases of $1P_{4,1}$ can be produced. It means that a single floor with a room layout corresponding to $1P_{4,1}$ can have 32 alternatives for sensor placement. The location of this placement refers to Figure 1. The sensor can be set in the room with a black vertex.

The visualization of a higher Petersen graph from $2P(4,1)$ to $4P(4,1)$ can only be derived for some samples due to insufficient system resources.

Table 3 System Specification

Items	**Specification**
Operating System	Debian 10
Processor	Intel® core™ i3 CPU 2.3 GHz.
Memory (RAM)	4 GB DDR3
OS Architecture	64 bit
Python	3.7.3 [GCC 8.3.0]
Libraries	ipython 7.13.0 jupyter 1.0.0 networkx 2.4 notebook 6.0.3

The samples of the basis formation within the three-dimensional representation for each Petersen graph are depicted in Figure 2. Assume that there are buildings with room connectivity represented by Figure 2. In parts (a) and (b), the buildings consist of four floors with eight rooms for each floor. Meanwhile, the rest have two and three floors. The placement of the sensor indicates red spots. The option of this placement can have many reasons, from technical aspects, regulation issues, cost consideration, esthetic view, or others.

The sensor may be installed into part (a). This option is because of the presence of signal noise that interferes with the sensor on the second and third floors. Therefore, the technician should install the sensors on the ground and on the highest floor. Another possibility of placement is in part (b). For example, there are technical obstacles to setting up sensors on the second and fourth floors. The sensors are eventually placed on the first and third floors. As another example in part (c), all sensors are equivalently distributed on each floor. This placement model meets one specified requirement of the sensor that all connected rooms can be accessed at most two rooms from the nearest sensor. Next, all yellow spots in part (c) satisfy this requirement. This sensor is a typical feature of Wi-Fi access points where each room should be reachable by the attached access points in a maximum of two consecutive rooms. On the contrary, in the last example in part (d), all sensors are positioned at the top floor on purpose by the wish of the owner of the building. The examples provided in parts (a) to (d) and many more are the possible reasons for selecting the placement design.

Table 4 Measurement Results

No.	Type of Petersen Graph	Number of Vertices in Basis	Execution Time	Number of Extracted Basis
1	$1P_{4,1}$	3	≤ 1 second	32
2	$2P_{4,1}$	4	≈ 3 seconds	989
3	$3P_{4,1}$	5	≈ 91 seconds	26.106
4	$4P_{4,1}$	6	≈ 3107 seconds	638.597

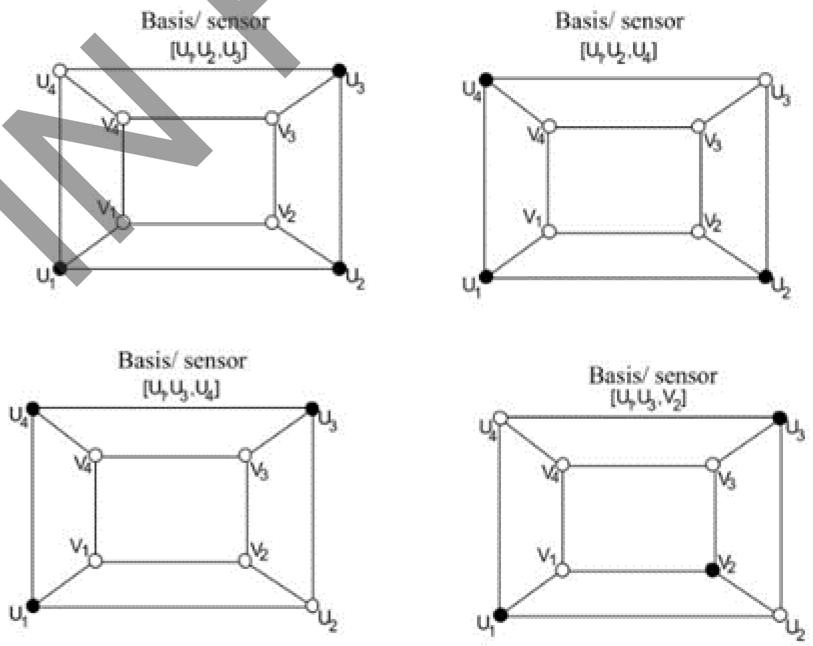

Figure 1 The Sample Configurations of Basis for $1P(4,1)$

IV. CONCLUSIONS

The number of bases of a graph can be defined as its metric dimension. The metric dimension of the graph approach can be used to compute all alternative bases of a generalized Petersen graph representing a multi-story building. The research successfully implements the metric dimension of $1P_{4,1}$ to $4P_{4,1}$ using Python code to obtain the alternative of its basis. A basic alternative indicates the location of the device placement like fire detectors, network access points, or other sensors inside a building. The sensor placement and the number of sensor determination on multi-story buildings is an exciting topic to study. The accomplishment of the research can be used to manage the sensor placement as well as its costs.

The research can be developed by determining alternative sensor placements for real public facilities, such as offices or hospitals. These public facilities can be represented in graphs and find metric dimension. Next, future research can determine the basis, which is an alternative sensor placement in the public facility.

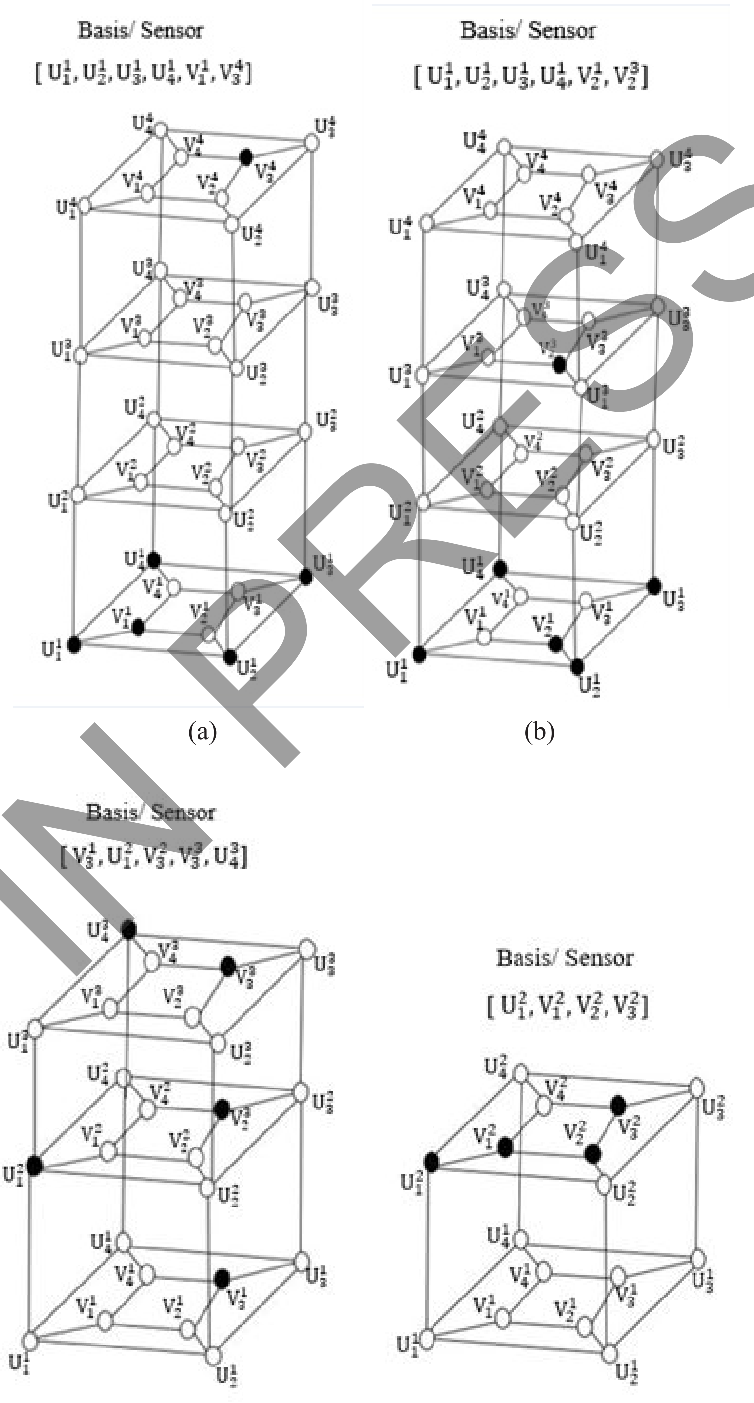
Basis/ Sensor
$[U_1^1, U_2^1, U_3^1, U_4^1, V_1^1, V_3^4]$
Basis/ Sensor
$[U_1^1, U_2^1, U_3^1, U_4^1, V_2^1, V_2^3]$
(a)
(b)
Basis/ Sensor
$[V_3^1, U_1^2, V_3^2, V_3^3, U_4^3]$
Basis/ Sensor
$[U_1^2, V_1^2, V_2^2, V_3^2]$
(c)
(d)

Figure 2 The Samples of Petersen Graph as Representation of Multi-Story Buildings

ACKNOWLEDGEMENT

The research was supported by the Directorate of Higher Education, Ministry of Education and Culture, under contract No. 857/UN26.21/PN/2019. The authors are thankful for the funding of the research.

REFERENCES

Adawiyah, R., Agustin, I. H., Prihandini, R. M., Alfarisi, R., & Albirri, E. R. (2019). On the local multiset dimension of *m*-shadow graph. *Journal of Physics: Conference Series*, 1211, 1–9. https://doi.org/10.1088/1742-6596/1211/1/012006

Adawiyah, R., Prihandini, R. M., Albirri, E. R., Agustin, I. H., & Alfarisi, R. (2019). The local multiset dimension of unicyclic graph. In *IOP Conference Series: Earth and Environmental Science* (pp. 1–8). IOP Publishing. https://doi.org/10.1088/1755-1315/243/1/012075

Alfarisi, R., Dafik, D., Kristiana, A. I., & Agustin, I. H. (2019). The local multiset dimension of graphs. *IJET*, *8*(3), 120–124.

Asmiati, A., Aldino, A. A., Notiragayu, N., Zakaria, L., & Anshori, M. (2020). Dimensi metrik hasil operasi tertentu pada graf Petersen diperumum. *Limits: Journal of Mathematics and Its Applications*, *16*(2), 87–93. https://doi.org/10.12962/limits.v16i2.5594

Bača, M., Baskoro, E. T., Salman, A. N. M., Saputro, S. W., & Suprijanto, D. (2011). The metric dimension of regular bipartite graphs. *Bulletin Mathématique de La Société Des Sciences Mathématiques de Roumanie, 54*(102), 15–28. https://www.jstor.org/stable/43679200

Chartrand, G., Eroh, L., Johnson, M. A., & Oellermann, O. R. (2000). Resolvability in graphs and the metric dimension of a graph. *Discrete Applied Mathematics*, *105*(1–3), 99–113. https://doi.org/10.1016/S0166-218X(00)00198-0

Dudenko, M., & Oliynyk, B. (2017). On unicyclic graphs of metric dimension 2. *Algebra and Discrete Mathematics, 23*(2), 216–222.

Hernando, C., Mora, M., Pelayo, I. M., Seara, C., & Wood, D. R. (2010). Extremal graph theory for metric dimension and diameter. *The Electronic Journal of Combinatorics*, *17*(1), 1–28. https://doi.org/10.37236/302

Imran, M., Siddiqui, M. K., & Naeem, R. (2018). On the metric dimension of generalized Petersen multigraphs. *IEEE Access*, *6*, 74328–74338. https://doi.org/10.1109/ACCESS.2018.2883556

Okamoto, F., Phinezy, B., & Zhang, P. (2010). The local metric dimension of a graph. *Mathematica Bohemica*, *135*(3), 239–255. https://doi.org/10.21136/MB.2010.140702

Saenpholphat, V., & Zhang, P. (2004). Conditional resolvability in graphs: A survey. *International Journal of Mathematics and Mathematical Sciences*, *2004*, 1997–2017. https://doi.org/10.1155/S0161171204311403

Saputro, S. W., Mardiana, N., & Purwasih, I. A. (2017). The metric dimension of comb product graphs. *Matematicki Vesnik, 69*(4), 248–258.

Saputro, S. W., Simanjuntak, R., Uttunggadewa, S., Assiyatun, H., Baskoro, E. T., Salman, A. N. M., & Bača, M. (2013). The metric dimension of the lexicographic product of graphs. *Discrete Mathematics, 313*(9), 1045–1051. https://doi.org/10.1016/j.disc.2013.01.021

Simanjuntak, R., Uttunggadewa, S., & Saputro, S. W. (2013). *Metric dimension of amalgamation of graphs*. Retrieved from https://arxiv.org/abs/1312.0191

Susilowati, L., Sa'adah, I., Fauziyyah, R. Z., Erfanian, A., & Slamin. (2020). The dominant metric dimension of graphs. *Heliyon*, *6*(3), 1–6. https://doi.org/10.1016/j.heliyon.2020.e03633

APPENDIX

Table 2 The Representation of Vertices Graph $4P_{n,1}$ with Basis Set W

Vertex	Taken Basis					
	u_1^1	u_h^1	u_n^1	u_n^2	u_n^3	u_n^4
u_1^1	0	$h-1$	1	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
u_2^1	1	$h-2$	2	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
u_3^1	2	$h-3$	3	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
$\vdots$						
u_h^1	$h-1$	0	$h-2$	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
u_{h+1}^1	$h-2$	1	$h-3$	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
$\vdots$						
u_n^1	1	$n-h$	0	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
v_1^1	1	h	2	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
v_2^1	2	$h-1$	3	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
v_3^1	3	$hh-2$	4	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
$\vdots$						
v_h^1	h	1	$h-1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
v_{h+1}^1	$h-1$	2	$h-2$	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
$\vdots$						
v_n^1	2	$h-1$	1	$u_h^{m-1}+1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
u_1^2	1	h	2	1	$u_h^{m-1}+1$	$u_h^{m-1}+1$
u_2^2	2	$h-1$	3	2	$u_h^{m-1}+1$	$u_h^{m-1}+1$
u_3^2	3	$h-2$	4	3	$u_h^{m-1}+1$	$u_h^{m-1}+1$
$\vdots$						
u_h^2	h	1	$h-1$	$h-2$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
u_{h+1}^2	$h-1$	2	$h-2$	$h-3$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
$\vdots$						
u_n^2	2	$h-1$	1	0	$u_h^{m-1}+1$	$u_h^{m-1}+1$

Table 2 The Representation of Vertices Graph $4P_{n,1}$ with Basis Set W *(Continued)*

Vertex	Taken Basis					
	u_1^1	u_h^1	u_n^1	u_n^2	u_n^3	u_n^4
v_1^2	2	$h+1$	3	2	$u_h^{m-1}+1$	$u_h^{m-1}+1$
v_2^2	3	h	4	3	$u_h^{m-1}+1$	$u_h^{m-1}+1$
v_3^2	4	$h-1$	5	4	$u_h^{m-1}+1$	$u_h^{m-1}+1$
$\vdots$						
v_h^2	$h+1$	2	h	$h-1$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
v_{h+1}^2	h	3	$h-1$	$h-2$	$u_h^{m-1}+1$	$u_h^{m-1}+1$
$\vdots$						
v_n^2	3	h	2	1	$u_h^{m-1}+1$	$u_h^{m-1}+1$
u_1^3	2	$h+1$	3	2	1	$u_h^{m-1}+1$
u_2^3	3	h	4	3	2	$u_h^{m-1}+1$
u_3^3	4	$h-1$	5	4	3	$u_h^{m-1}+1$
$\vdots$						
u_h^3	$h+1$	2	h	$h-1$	$h-2$	$u_h^{m-1}+1$
u_{h+1}^3	h	3	$h-1$	$h-2$	$h-3$	$u_h^{m-1}+1$
$\vdots$						
u_n^3	3	h	2	1	0	$u_h^{m-1}+1$
v_1^3	3	$h+2$	4	3	2	$u_h^{m-1}+1$
v_2^3	4	$h+1$	5	4	3	$u_h^{m-1}+1$
v_3^3	5	h	6	5	4	$u_h^{m-1}+1$
$\vdots$						
v_h^3	$h+2$	3	$h+1$	h	$h-1$	$u_h^{m-1}+1$
v_{h+1}^3	$h+1$	4	h	$h-1$	$h-2$	$u_h^{m-1}+1$
$\vdots$						
v_n^3	4	$h+1$	3	2	1	$u_h^{m-1}+1$

Table 2 The Representation of Vertices Graph $4P_{n,1}$ with Basis Set W *(Continued)*

Vertex	Taken Basis					
	u_1^1	u_h^1	u_n^1	u_n^2	u_n^3	u_n^4
u_1^4	3	$h+2$	4	3	2	1
u_2^4	4	$h+1$	5	4	3	2
u_3^4	5	h	6	5	4	3
⋮						
u_h^4	$h+2$	3	$h+1$	h	$h-1$	$h-2$
u_{h+1}^4	$h+1$	4	h	v	$h-2$	$h-3$
⋮						
u_n^4	4	$h+1$	3	2	1	0
v_1^4	4	$h+3$	5	4	3	2
v_2^4	5	$h+2$	6	5	4	3
v_3^4	6	$h+1$	7	6	5	4
⋮						
v_h^4	$h+3$	4	$h+2$	$h+1$	h	$h-1$
v_{h+1}^4	$h+2$	5	$h+1$	h	$h-1$	$h-2$
⋮						
v_n^4	5	$h+2$	4	3	2	1