Research Project Report

The game chromatic number of graphs

By

Siriwan Wasukree

A Research Project Funded by

Research Institute of Rangsit University

Academic Years 2015

Contents (1)

Contents (2)

CHAPTER I

Introduction to Wolfram Mathematica 10.0

1.1 Basic Calculation

Mathematica is a convenient tools for doing basic mathematics. You may use Mathematica like a calculator. Type your equation, press $\boxed{\text{Shift}}+\boxed{\text{Enter}}$ to obtain the answer.

After you open the Mathematica and run a first calculation, you need to wait a few seconds before the Kernel is loaded. When the kernel keep running, your calculation will be fast; hence, keep it running. It will run as long as the Front End is open. When you close the Front End, the Kernel will be immediately closed.

We start first Mathematica calculations. Type $1 + 3$ and run the command by pressing $\boxed{\text{Shift}}+\boxed{\text{Enter}}$. (See Input[1] and Output[1])

Second, type $6 - 3$ and run the command. (See Input[2] and Output[2]).

Next, type 3*5 and 3 5 and run both commands. (See Input[3], Output[3], Input[4] and Output[4]). Both output are 15 because you can multiply two numbers by using * or a space.

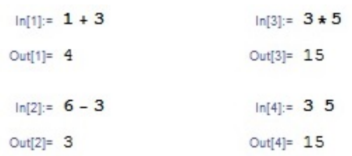

Figure 1.1: Add, subtract and multiply two numbers

The imaginary unit is represented by I. You can calculate any complex number by using I for imaginary part. For example, type $(1+I)(2-I)$ and press $\boxed{\text{Shift}}+\boxed{\text{Enter}}$ to obtain the answer. See Figure 1.2

```
In[5]:= (1 + I) (2 - I)
Out[5]= 3 + i
```

Figure 1.2: Complex number calculation

Next, you may try dividing two numbers by using symbol \. Suppose that you want to find the answer of $8 \div 2$. You can find the answer by typing 8\2 and press $\boxed{\text{Shift}}+\boxed{\text{Enter}}$. See Figure 1.3.

The order in calculations in Mathematica is normal. That is, Mathematica will give more priority to multiplication and division than summation and subtract. To change the order, you have to use brackets. When there is no brackets, the calculation will start from the left.

```
In[6]:= 8 / 2
Out[6]= 4

In[7]:= (2 + 4) - 3 * 2
Out[7]= 0
```

Figure 1.3: Divide two numbers

You can use the formula x^y to obtain x to the power y. For example, type x^y to calculate 3^4 which is 81. To find $\sqrt{x}$, you may use the command **Sqrt**[**x**].

1.2 Matrix operations

Matrix operation in Mathematica can do both nummeric and symbolic matrices. It will apply highly efficient algorithms. You calculate a complicated matrix operation in a second by using the following function

`Transpose[m]`	transpose m^T
`ConjugateTranspose[m]`	conjugate transpose $m^\dagger$ (Hermitian conjugate)
`Inverse[m]`	matrix inverse
`Det[m]`	determinant
`Minors[m]`	matrix of minors
`Minors[m,k]`	k^{th} minors
`Tr[m]`	trace
`MatrixRank[m]`	rank of matrix

Figure 1.4: List of matrix operations

For Mathematica language, a matrix is a list of lists. We can assign a matrix m by writing a list of lists and when we like see m as a matrix, we need the function **MatrixForm**.

```
In[1]:= m = {{1, -2, 3}, {0, 3, -2}, {4, 2, 1}}
Out[1]= {{1, -2, 3}, {0, 3, -2}, {4, 2, 1}}

In[2]:= MatrixForm[m]
Out[2]//MatrixForm=
 1  -2   3
 0   3  -2
 4   2   1
```

Figure 1.5: How to build a matrix

The *transpose* of an $m \times n$ matrix is an $n \times m$ matrix obtained from interchanging the rows and columns in the matrix. The *conjugate transpose* of an $m \times n$ matrix with complex entries is the $n \times m$ matrix obtained from the matrix by taking the transpose and then taking the complex conjugate of each entry. When all elements of matrices are real numbers, transpose and conjugate transpose are the same.

```
In[3]:= Transpose[m]
Out[3]= {{1, 0, 4}, {-2, 3, 2}, {3, -2, 1}}

In[21]:= ConjugateTranspose[m]
Out[21]= {{1, 0, 4}, {-2, 3, 2}, {3, -2, 1}}
```

Figure 1.6: Transpose and conjugate transpose

To find determinant and an inverse matrix, we just apply the function **Det** and **Inverse**, respectively.

```
In[4]:= Inverse[m]
Out[4]= {{-7/13, -8/13, 5/13}, {8/13, 11/13, -2/13}, {12/13, 10/13, -3/13}}

In[5]:= Det[m]
Out[5]= -13
```

Figure 1.7: Determinant and an inverse matrix

The function **Minor** gives all possible minors of a matrix while **Tr** gives the summation of all elements in the main diagonal. The rank of a matrix is the number of linearly independent rows or columns.

```
In[6]:= Minors[m]
Out[6]= {{3, -2, -5}, {10, -11, -8}, {-12, 8, 7}}

In[7]:= Tr[m]
Out[7]= 5

In[8]:= MatrixRank[m]
Out[8]= 3
```

Figure 1.8: Minor, Tr and MatrixRank

1.3 Basic plotting

Plotting a graph as a series of one or more points, lines, curves or areas represents the equations in comparison with one or more variables. In other words, the plotting displays on a Cartesian coordinate in several ways which depends on what you want to interpret.

Plotting by using Mathematica is quite easy. We can do a basic plotting by using the function **Plot**. The syntax is in the form **Plot[f,{x,min,max}]** First, plot $\cos(x)$ as a function of x from 0 to 2π.

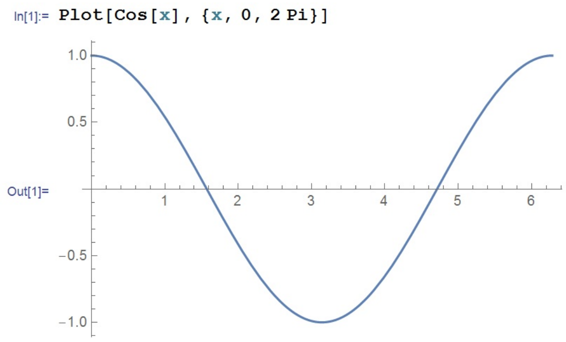

Figure 1.9: A cosine graph

In the second example, we plot a graph of $\tan(x)$ as a function of x from -4 to 4. According to the default, the plotting have singularities. Mathematica will try to choose appropriate scales.

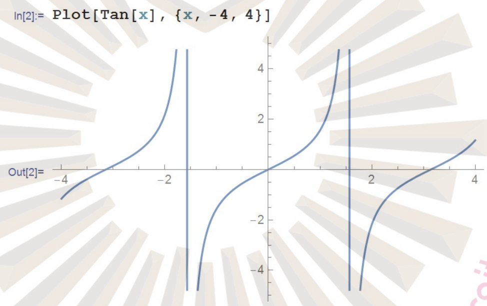

Figure 1.10: A tan graph with singularities

In case, we also can plot $\tan(x)$ without singularities by using the Exclusions option.

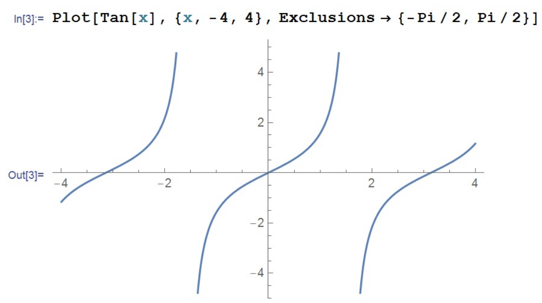

Figure 1.11: A tan graph without singularities

In the next example, we plot $\cos(x), \cos(2x)$ and $\cos(3x)$ as a function of x from 0 to 2π in the same axis. A different color will automatically be used for each function.

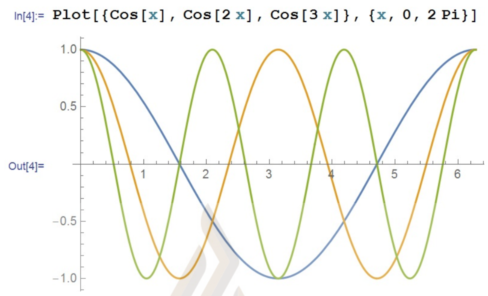

Figure 1.12: Three graphs in same axis

1.4 Three-dimensional plotting

To do 3-dimension plotting, we use the function **Plot3D**. For Mathematica, it is not difficult to plot 3-dimension graph. For example, we plot $z = \sin(xy)$ as a function of x and y from -3 to 3 and -2 to 2, respectively.

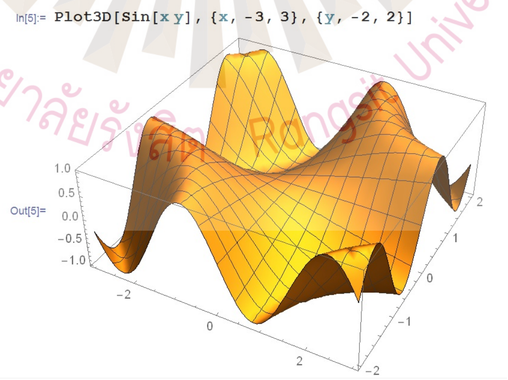

Figure 1.13: Function $z = \sin(xy)$

Next, we want to plot $z = x^2 + y^2$ as a function of x and y from -3 to 3 and -2 to 2, respectively.

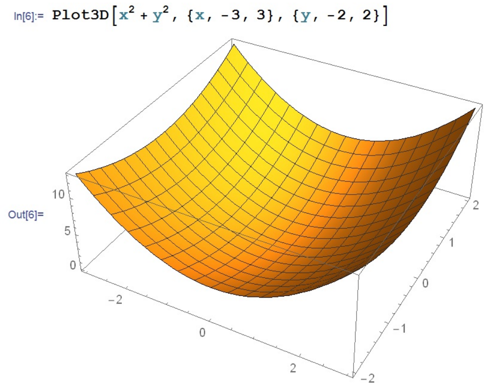

Figure 1.14: Function $z = x^2 + y^2$

Next, we want to plot $z = x + y$ as a function of x and y from -3 to 3 and -2 to 2, respectively.

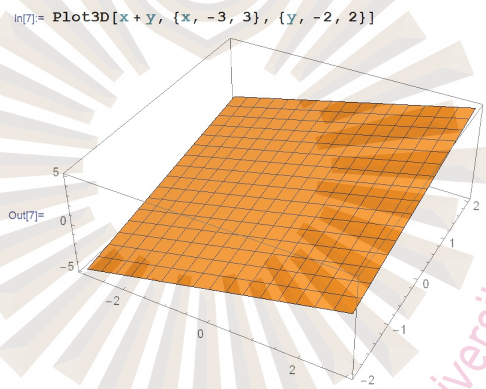

Figure 1.15: Function $z = x + y$

1.5 Parametric plotting

Parametric equation of a curve show the coordinate of points of a curve as functions of a variable, which is called *parameter*. For instant, $x = \sin(t)$ and $y = \cos(t)$ are parametric equations of a unit circle; t is the parameter of the equations. Notice that $x^2 + y^2 = \sin^2(t) + \cos^2(t) = 1$.

We usually use the valuable t as the parameter because the parametric equations often represent a physical movement in time. However, the parameter can represent some other physical values such as a geometric valuable.

ParametricPlot generates a parametric plot of a curve with coordinates $f(u)$ and $f(u)$ as a function of a parameter u. In Figure 1.16, we draw a unit circle with coordinates $\cos(t)$ and $\sin(t)$ where t from 0 to 2π

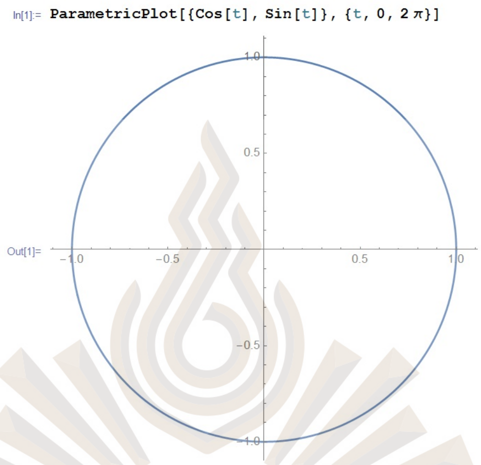

Figure 1.16: A unit circle constructed by parametric plot

We can change coordinate functions to obtain a different line. In Figure 1.17, we draw a curve with coordinates $\cos(t)$ and $\sin(4t)$ where t from 0 to 2π.

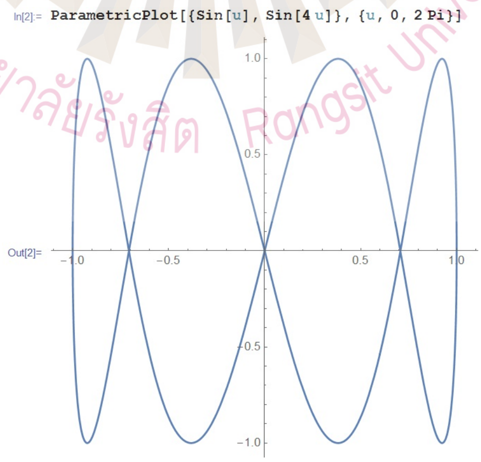

Figure 1.17: Another curve constructed by parametric plot

ParametricPlot3D generates a parametric plot of a 3-dimensional curve. We can draw a three dimensional line in by using this function. For example, we draw a line with coordinate $(x, y, x) = (\sin(u), \cos(u), \frac{u}{30}$ where u from 0 to 50.

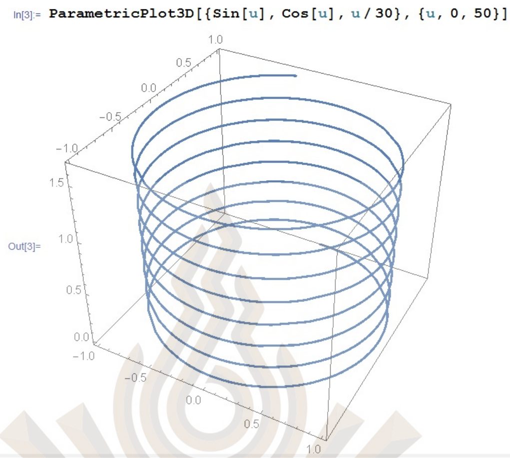

Figure 1.18: A line constructed by ParametricPlot3D

Suppose that we are trying to draw a cylinder, which is a three-dimensional surface. We can easily draw it by using the function **ParametricPlot3D**. Figure 1.19 shows a surface with coordinate $(x, y, x) = (\cos(u), \sin(u), \sin(v)$ where u and v from 0 to 2π and $-\pi$ to π, respectively.

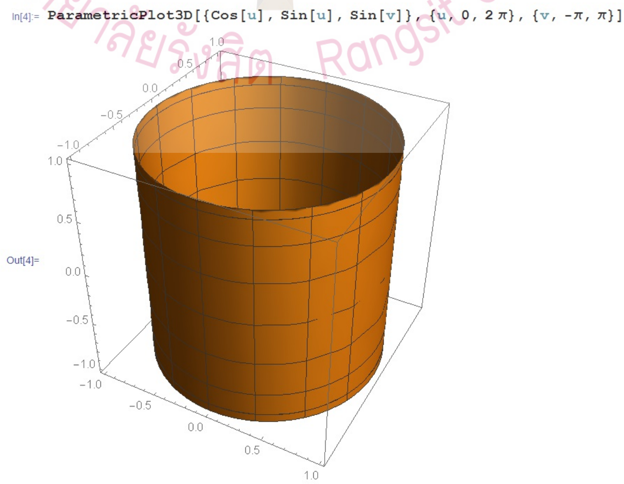

Figure 1.19: Surface constructed by ParametricPlot3D

The next example shows wonderful surface created by usingParametricPlot3D.

```
In[5]:= ParametricPlot3D[{{4 + (3 + Cos[v]) Sin[u], 4 + (3 + Cos[v]) Cos[u], 4 + Sin[v]},
           {8 + (3 + Cos[v]) Cos[u], 3 + Sin[v], 4 + (3 + Cos[v]) Sin[u]}}, {u, 0, 2 Pi},
          {v, 0, 2 Pi}]
```

Out[5]=

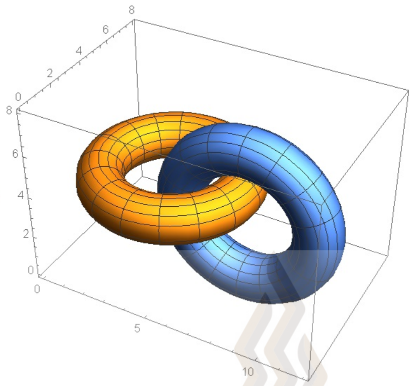

Figure 1.20: Wonderful surface constructed by ParametricPlot3D

CHAPTER II

Mathematica in Graph Theory

2.1 How to draw a graph

In graph theory, a *graph* is a figure of a set of nodes where some *nodes* are linked by *lines*. The nodes sometimes are called *vertices* and the lines sometimes are called *edges*. The definition is the objects in *discrete mathematics*.

The edges in each graph can be directed or undirected, which depend on what the graph are represented. For example, if the vertices represent football team at a tournament and there is an edge between two teams if they meet each other, then this is an undirected graph becasue if team A meets team B, then team B also meet team A. However, if there is an edge from team A to team B when team A defeats team B, then this graph is directed because winning does not a symmetric relation. The first type of graphs is called an *undirected graph* and the edges are called *undirected edges* while the second type of graphs is called an *directed graph* and the edges are called *directed edges*.

In formal definition of graphs, a graph G is an order pair (V(G),E(G)) where $V(G)$ is the set of vertices of G and $E(G)$ is the set of edges of G. Without said otherwise, a graph is *simple* and *undirected*.

The other type of graph is obtained from the different type of the edge set. In more generalized definitions, $E(G)$ allows a *loop* or *multiple edges*. The graph which allows a loop and multiple edges is callled a *multigraph*.

A *loop* is an edge whose both end points are on the same vertex.

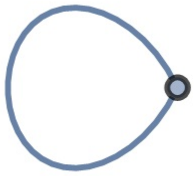

Figure 2.1: A loop

Multiple edges are two or more edges that whose endpoints are on the same two vertices

Figure 2.2: multiple edges

Throughout this article, our graphs are always simple and undirected because we are in coloring topics. In an undirected graph, if there is an edge between vertex 1 and vertex 2, we often use the notation $1 \leftrightarrow 2$

Wolfram mathematica 10.0 has several functions to help us construct graphs. The function **Graph** is easiest way to draw a general graph. For example, if we want to draw a paw.

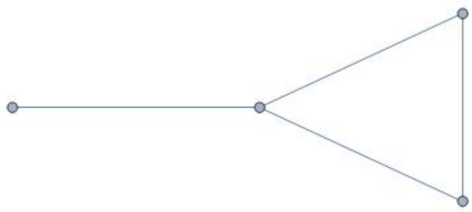

Figure 2.3: A paw graph

We just labels all vertices with different numbers. Normally, we use $1, 2, \ldots, n$

for an n-vertex graph.

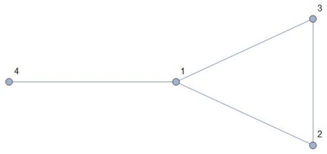

Figure 2.4: A paw graph with numbers on vertices

To obtain the graph, we write

$$Graph[1 <-> 2, 2 <-> 3, 3 <-> 1, 1 <-> 4]$$

The following figures show what we obtain.

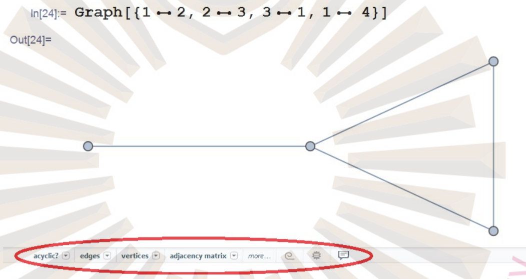

Figure 2.5: A paw and its code.

2.2 Frequently used functions

The most interesting things is the commands under the figures. Sometimes, after we draw a big graph but we do not sure its properties. We can check its properties by using the functions.

For example, we can check whether a paw is acyclic by clicking on a word **acyclic?** to get the answer shown belowed.

According to Figure 2.6, the answer is false. That is, the graph is not a cyclic becasue it contains a cycle.

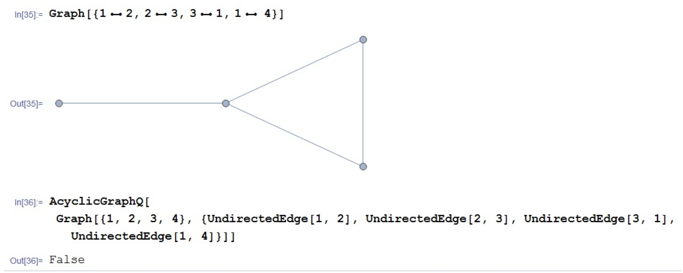

Figure 2.6: An answer after clicking on **acyclic?**

In case we want to check other properties, we can click the button beside the word **acyclic?** to obtain other functions.

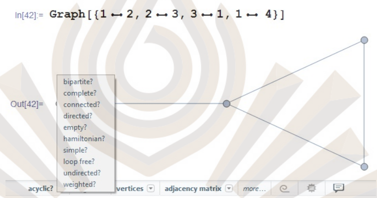

Figure 2.7: List of functions

The following are lists of properties we can check by using the first tab in Wolfram Mathematica 10.0.

- acyclic : Check whethere a graph contains a cycle.
- bipartite : Check whethere a graph contains has no odd cycle.
- complete : Check whether a graph is a complete graph.
- connected : Check whether a graph is connected
- directed : Check whether a graph is directed.
- empty : Check whether a graph is empty.
- hamiltonian : Check whether a graph is hamiltonian.
- simple : Check whether a graph is simple.

- loop free : Check whether a graph has no loop.
- undirected : Check whether a graph is undirected.
- weighted : Check whether a graph is weighted.

On the second tab, you will find some useful functions related to edges of graphs such as the number of edges. On the third tab, you will find several useful functions related to vertices of graphs such as the number of vertices and vertex degree. On the fourth tab, you can obtain adjacency matrices from these functions.

2.3 More useful functions

Surprisingly, when we click on **More...** under the figure, we obtain several useful functions in graph theory.

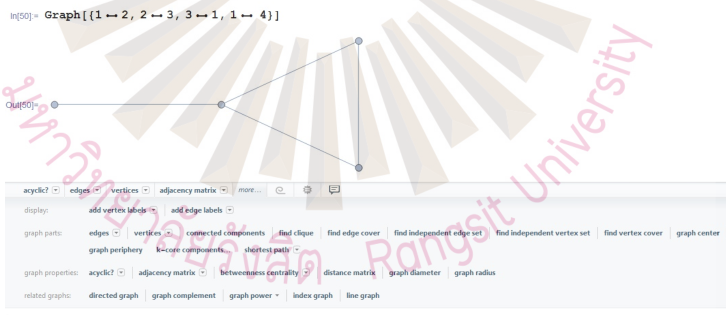

Figure 2.8: A number of useful functions

The useful functions are categorized into 4 groups. The first group is *display*. This group consists of two main functions which are **add vertex labels** and **add edge labels**. The two functions are quite straightforward because the tools are applied to add labels to vertices or edges.

The second group is *graph part*. This group consists of twelve main functions which are shown in the following list.

- edges
- vertices
- connected component
- find clique
- find edge cover
- find independent edge set
- find independent vertex set
- find vertex cover
- find graph center
- find graph periphery
- k-core components
- Shortest path

For instant, we want to find an edge cover of the paw. We just click on **find edge cover** to get an answer. Recall that an *edge cover* is a set of edges such that all edges are in the set or are incident to an edge in the set.

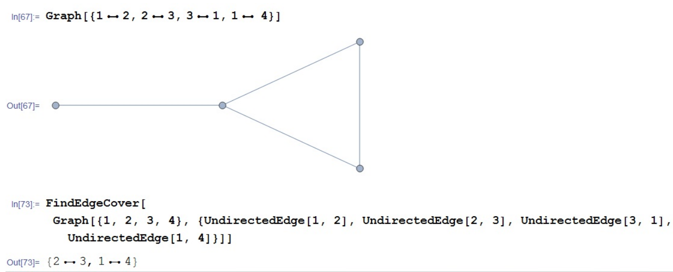

Figure 2.9: An edge cover of the paw

The function will give us one of edge covers.

Similarly, if we want to find a vertex cover of the paw, then we just click on **find vertex cover** to get an answer. Recall that a *vertex cover* is a set of vertices such that all vertices are in the set or are adjacent to a vertex in the set.

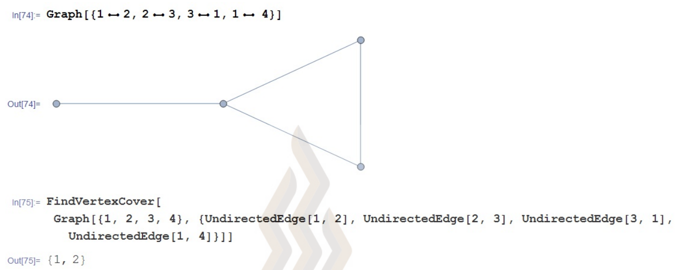

Figure 2.10: A vertex cover of the paw

The function will give us one of vertex cover.

The third group is *graph properties* consists of six main functions which are shown in the following list.

- acyclic?
- adjacency metrix
- betweeness centrality
- distance metrix
- graph diameter
- graph radius

The function **Acyclic?** and its similar functions have already mentioned in the beginning of this section. We raise the function **graph diameter** as an example of this group. Recall that the *graph diameter* is the longest shortest path or a graph. We can easily find the graph diameter by using this function

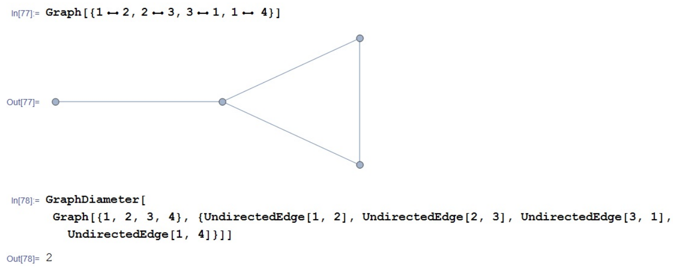

Figure 2.11: The graph diameter of a paw

According to the figure, it is easy to see that the diameter of a paw is 2.

The last group is *related graph* consists of five main functions which are shown in the following list.

- directed graph
- graph complement
- graph power
- index graph
- line graph

The function **directed graph** constructs a directed graph from an undirected graph.

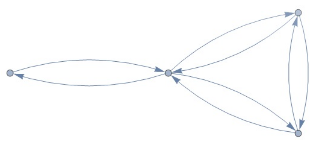

Figure 2.12: A directed paw graph

A *graph complement* G' is obtained from a graph G such that $V(G') = V(G)$ and $e \in E(G')$ if and only if $e \notin E(G)$

A *k-power graph* G^k is the graph obtained from a graph G such that $V(G^k) = V(G)$ and two vertices are adjacent when their distant in G is at most k.

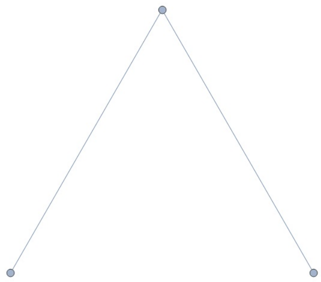

Figure 2.13: The graph complement of a paw

Figure 2.14: A 2-power graph of a paw

The function **index graph** is not in graph theory but it is defined to relabel the name of vertices

A *line graph* $L(G)$ is a graph obtained from a graph G such that each vertex of $L(G)$ represents an edge of G and two vertices of $L(G)$ are adjacent if and only if their corresponding edges are incident in G.

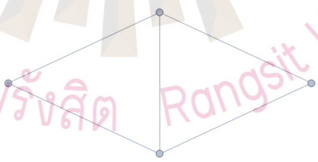

Figure 2.15: A Line graph of a paw

CHAPTER III

Classes of graphs with Mathematica

When we try to make any conjecture in Graph Theory, we usually test the conjecture in a well-know graph. In order to check it, we need to draw the graph. Unfortunately, sometimes, we have to draw a lot of graphs and some graphs are too large. Thanks to Mathematica, we can easily draw a large well-known graph in a second.

3.1 Path

A *path* in a graph is a finite or infinite sequence of edges which connect a sequence of vertices which, by most definitions, are all distinct from one another. In a directed graph, a directed path (sometimes called *dipath*) is again a sequence of edges (or arcs) which connect a sequence of vertices, but with the added restriction that the edges all be directed in the same direction. A path with n vertices is denoted by P_n

Paths are fundamental concepts of graph theory, described in the introductory sections of most graph theory texts.

We can use Mathematica to draw a path P_n by typing **PathGraph**$[v_1, v_2, \ldots, v_n]$. For example, we can draw a path P_5 by typing **PathGraph**$[a,\ b,\ c,\ d,\ e]$. The name of five vertices are a, b, c, d and e.

Figure 3.1: Path P_5

3.2 Cycle

In graph theory, a *cycle* or *circular graph* is a graph that consists of a single cycle, or in other words, some number of vertices connected in a closed chain. The cycle graph with n vertices is called C_n. The number of vertices in C_n equals the number of edges, and every vertex has degree 2; that is, every vertex has exactly two edges incident with it.

There are many synonyms for *cycle*. These include *simple cycle graph* and *cyclic graph*, although the latter term is less often used, because it can also refer to graphs which are merely not acyclic. Among graph theorists, *cycle*, *polygon*, or *n-gon* are also often used. A cycle with an even number of vertices is called an *even cycle*; a cycle with an odd number of vertices is called an *odd cycle*.

Thanks to Mathematica, we can easily draw a cycle with n-vertices by typing **CycleGraph**[**n**]. For example, we can draw C_5 by typing **CycleGraph**[**5**] to obtain the figure.

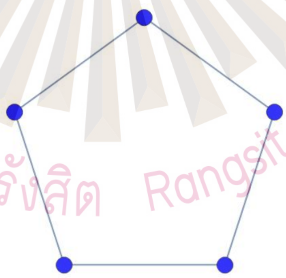

Figure 3.2: Cycle C_5 obtaining from Mathematica

Suppose we want to draw $C_3, C_5, \ldots, C_{10}$ in the same figure. We type **Table**[**CycleGraph**[**i**], **i, 3, 10**] to obtain the figure.

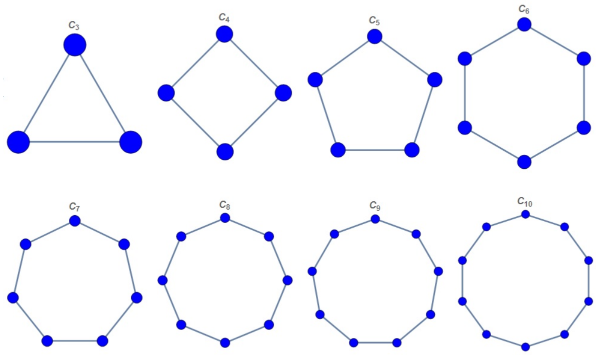

Figure 3.3: Eight Cycles in one figure

3.3 Complete graph

A *complete graph* is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge. A complete digraph is a directed graph in which every pair of distinct vertices is connected by a pair of unique edges (one in each direction).

The complete graph on n vertices is denoted by K_n.

K_n has $\frac{n(n1)}{2}$ edges (a triangular number), and is a regular graph of degree $n-1$. All complete graphs are their own maximal cliques. They are maximally connected as the only vertex cut which disconnects the graph is the complete set of vertices. The complement graph of a complete graph is an empty graph. If the edges of a complete graph are each given an orientation, the resulting directed graph is called a *tournament*.

We can easily draw a complete graph with n vertices by using the function

$$CompleteGraph[n]$$

For example, we want to draw a complete graph with 5 vertices. Instead of drawing five vertices and ten edges, we just type **CompleteGraph[5]** to obtain the graph.

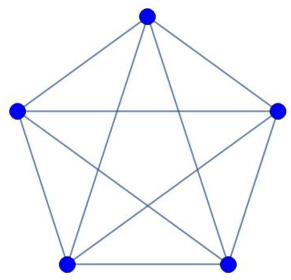

Figure 3.4: Complete graph K_5 obtaining from Mathematica

Suppose that we want to draw a complete graph with 15 vertices. Thanks to Mathematica, we do not need to draw $\binom{15}{2} = 105$ edges. We just type **Complete-Graph15**] to obtain the graph.

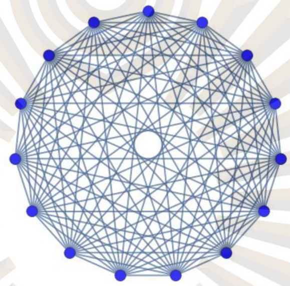

Figure 3.5: Complete graph K_{15} obtaining from Mathematica

Sometimes, we want to draw several graph in a figure. Suppose we want to draw $K_3, K_4, \ldots, K_{10}$ in the same figure. We just type

$$Table[CompleteGraph[i, PlotLabel- > Subscript[K, i]], i, 3, 10]$$

to obtain the figure.

3.4 Complete bipartite graph

A *complete bipartite graph* or *biclique* is a special kind of bipartite graph where every vertex of the first set is connected to every vertex of the second set

A *complete bipartite graph* is a graph whose vertices can be partitioned into

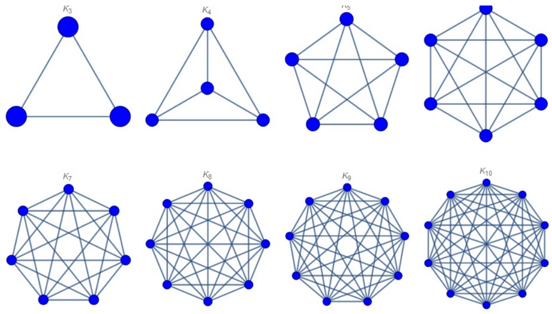

Figure 3.6: Eight complete graphs in one figure

two subsets V_1 and V_2 such that no edge has both endpoints in the same subset, and every possible edge that could connect vertices in different subsets is part of the graph. That is, it is a bipartite graph (V_1, V_2, E) such that for every two vertices $v_1 \in V_1$ and $v_2 \in V_2$, v_1v_2 is an edge in E. A complete bipartite graph with partitions of size $|V_1| = m$ and $|V_2| = n$, is denoted $K_{m,n}$; every two graphs with the same notation are isomorphic.

Some complete bipartite graphs have its own special names as shown in the following.

- $K_{1,k}$ is called a star for any number k.
- $K_{1,3}$ is called a claw.
- $K_{3,3}$ is called the utility graph.

This usage of the *utility graph* comes from a standard mathematical puzzle in which three utilities must each be connected to three buildings; it is impossible to solve without crossings due to the *nonplanarity* of $K_{3,3}$.

We can easily draw a complete bipartite graph $K_{a,b}$ vertices by using the function

$$CompleteGraph[\{a,b\}]$$

For example, we want to draw a complete bipartite graph $K_{3,4}$. Instead of drawing seven vertices and twelve edges, we just type **CompleteGraph[{3,4}]** to obtain the graph.

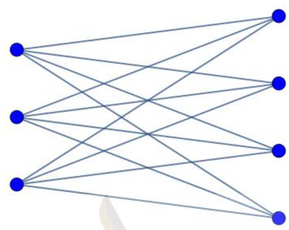

Figure 3.7: Complete bipartite graph $K_{3,4}$

Similarly, we can easily draw a complete graph $K_{5,8}$ by typing **CompleteGraph[{5,8}]**.

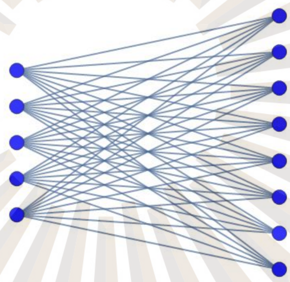

Figure 3.8: Complete bipartite graph $K_{5,8}$

Again, we want a list of complete bipartite graph in one figure. Suppose that we want $K_{3,3}, K_{4,4}, \ldots, K_{10,10}$ in a figure. We just type **Table[CompleteGraph[{i, i}, i, 3, 10]** to obtain the figure.

3.5 Complete multiparite graph

A *complete k-partite graph* is a graph that can be partitioned into k independent sets, so that every pair of vertices from two different independent sets have different colors. These graphs are described by notation with a capital letter K

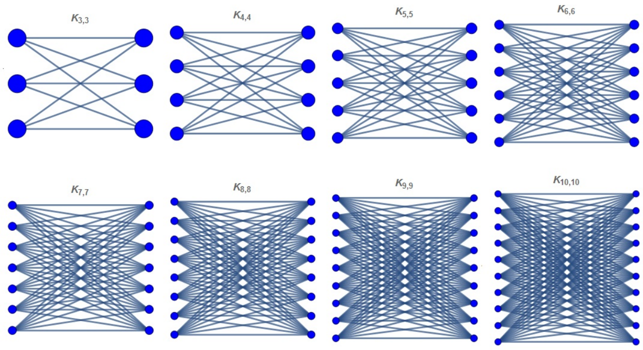

Figure 3.9: Eight complete bipartite graphs in one figure

subscripted by a sequence of the sizes of each set in the partition. For instance, $K_{2,2,2}$ is the complete tripartite graph of a regular octahedron, which can be partitioned into three independent sets each consisting of two opposite vertices. A *complete multipartite graph* is a graph that is complete k-partite for some k.

Thanks to Mathematica, we can easily draw complete multipartite graphs by using the same function. For instant, we type **CompleteGraph[{2,2,2}]** to obtain $K_{2,2,2}$ as shown in Figure 3.10.

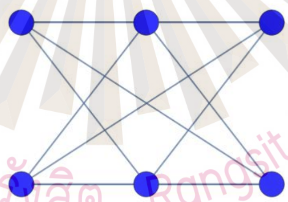

Figure 3.10: Complete tripartite graph $K_{2,2,2}$

When we want to draw complete multipartite graph $K_{4,5,6,7}$, we type **CompleteGraph[{4,5,6,7}]** to obtain the figure.

3.6 Star

A *star* S_k is the complete bipartite graph $K_{1,k-1}$; a tree with one internal node and $k-1$ leaves (but, no internal nodes and k leaves when $k \leq 2$). Alternatively,

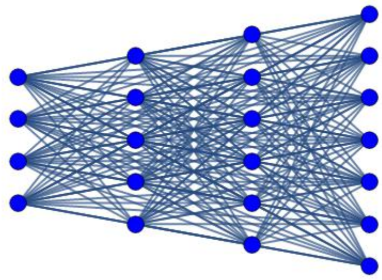

Figure 3.11: Complete multipartite graph $K_{4,5,6,7}$

some authors define S_k to be the tree of order $k + 1$ with maximum diameter 2; in which case a star of $k > 3$ has $k1$ leaves.

A star with 3 edges is called a claw.

The star S_k is edge-graceful if and only if k is odd. It is an edge-transitive matchstick graph, and has diameter 2 (when $k > 2$), girth ∞ (it has no cycles), chromatic index $k - 1$, and chromatic number 2 (when $k \geq 2$). Additionally, the star has large automorphism group, namely, the symmetric group on $k - 1$ letters.

Stars may also be described as the only connected graphs in which at most one vertex has degree greater than one

We can easily draw a star S_n by typing **StarGraph**[**n**]. For instance, we can draw stars S_8 and S_{10} by typing **StarGraph**[**8**] and **StarGraph**[**10**] to obtain the graphs.

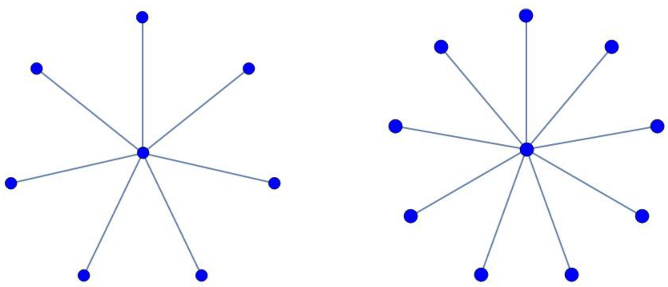

Figure 3.12: Stars S_8 and S_{10}

Several graph invariants are defined in terms of stars. *Star arboricity* is the minimum number of forests that a graph can be partitioned into such that each tree in each forest is a star, [8] and the *star chromatic number* of a graph is the minimum number of colors needed to color its vertices in such a way that every two color classes together form a subgraph in which all connected components are stars.[5] The graphs of branchwidth 1 are exactly the graphs in which each connected component is a star.

The following figure are beautiful big star creating by using Mathematica.

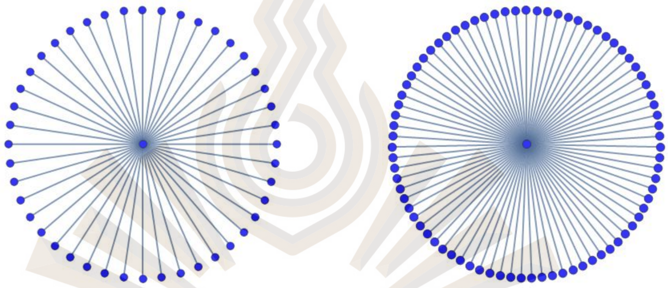

Figure 3.13: Stars S_{45} and S_{80}

3.7 Turan graph

The *Turan graph* $T_{n,r}$ is a complete multipartite graph formed by partitioning a set of n vertices into r subsets, with sizes as equal as possible, and connecting two vertices by an edge whenever they belong to different subsets. The graph will have $(n \mod r)$ subsets of size $\lceil \frac{n}{r} \rceil$, and $r - (n \mod r)$ subsets of size $\lfloor \frac{n}{r} \rfloor$. That is, it is a complete r-partite graph $K_{\lceil n/r \rceil, \lceil n/r \rceil, \ldots, \lfloor n/r \rfloor, \lfloor n/r \rfloor}$. Each vertex has degree either $n - \lceil n/r \rceil$ or $n - \lfloor n/r \rfloor$. The number of edges is $\frac{1}{2}(n^2 - (n \mod r)\lceil n/r \rceil^2 - (r - (n \mod r))\lfloor n/r \rfloor^2) \leq \left(1 - \frac{1}{r}\right)\frac{n^2}{2}$. It is a regular graph, if n is divisible by r.

Several choices of the parameter r in a Turan graph lead to notable graphs that have been independently studied.

The Turn graph $T_{2n,n}$ can be formed by removing a perfect matching from a complete graph K_{2n}. As Roberts (1969) showed, this graph has boxicity exactly n; it is sometimes known as the *Roberts graph*. This graph is also the 1-skeleton of an n-dimensional cross-polytope; for instance, the graph $T_{6,3} = K_{2,2,2}$ is the octahedral graph, the graph of the regular octahedron. If n couples go to a party, and each person shakes hands with every person except his or her partner, then this graph describes the set of handshakes that take place; for this reason it is also called the *cocktail party graph*.

The Turan graph $T_{n,2}$ is a complete bipartite graph and, when n is even, a *Moore graph*. When r is a divisor of n, the Turan graph is symmetric and strongly regular, although some authors consider Turan graphs to be a trivial case of strong regularity and therefore exclude them from the definition of a strongly regular graph.

Thanks to mathematica, we can easily draw a Turan graph $T_{n,k}$ by using function **TuranGraph[n,k]**

We may substitute 20 and 3 for n and k to obtain a Turan graph $T_{20,3}$.

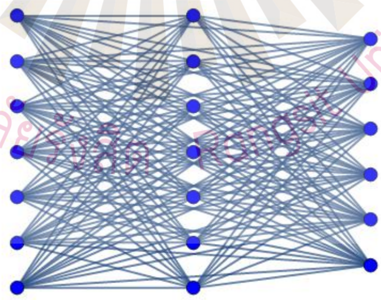

Figure 3.14: Turan graph $T_{20,3}$

We may substitute 30 and 4 for n and k to obtain a Turan graph $T_{30,4}$.

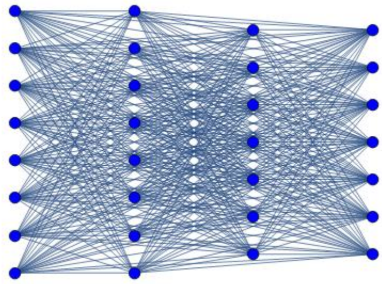

Figure 3.15: Turan graph $T_{30,4}$

3.8 Wheel graph

A *wheel graph* W_n is a graph with $n+1$ vertices ($n3$), obtained from connecting a single vertex to all vertices of a cycle with n vertices. Some authors instead use n to refer to the number of vertices of wheel graphs, so that their W_n is the graph that we denote W_{n-1}. The chromatic number of W_n is three when n is even and four when n is odd.

In mathematica, a wheel graph W_n is a graph with n vertices obtained from connecting a single vertex to all vertices of a cycle with $n-1$ vertices.

We can easily draw a wheel W_n by typing **WheelGraph[n]**. For instance, we can draw wheel graphs W_{20} and W_{30} by typing **WheelGraph[20]** and **WheelGraph[30]** to obtain the graphs.

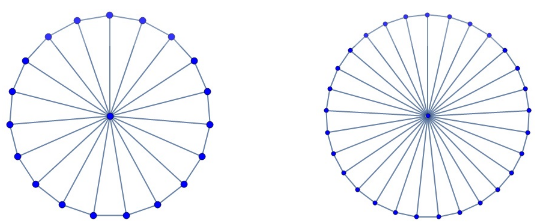

Figure 3.16: Wheel graphs W_{20} and W_{30}

Suppose we want to draw $W_5, W_6, \ldots, K_{10}$ in the same figure. We just type

$$Table[WheelGraph[i, PlotLabel- > Subscript[W, i]], i, 5, 10]$$

to obtain the figure.

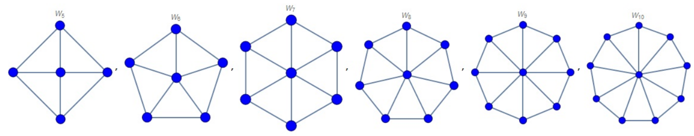

Figure 3.17: Six wheel graphs in one figure

3.9 Hypercube graph

A *hypercube graph* Q_n is a regular graph with $2n$ vertices, $(2^{n-1}n$ edges. It can be obtained as the one-dimensional skeleton of the geometric hypercube; for instance, Q_3 is the graph formed by the 8 vertices and 12 edges of a three-dimensional cube. Alternatively, it can be obtained from the family of subsets of a set with n elements, by making a vertex for each possible subset and joining two vertices by an edge whenever the corresponding subsets differ in exaclty one element.

Hypercube graphs should not be confused with *cubic graphs*, which are graphs that have exactly three edges touching each vertex. The only hypercube, Q_n that is a cubic graph is the cubical graph, Q_3.

The hypercube graph Q_n usually be constructed from the family of subsets of a set with n elements, by making a vertex for each possible subset and joining two vertices by an edge if and only if the corresponding subsets differ in exactlyo one element. Moverover, it can be constructed using $2n$ vertices labeled with n-bit binary numbers and connecting two vertices by an edge if and only if the corresponding binaries differ in exactly one element. These two constructions are closely related.

Furthermore, Q_{n+1} can be constructed from the disjoint union of two hypercubes Q_n, by adding an edge from each vertex in one copy of Q_n to the corresponding vertex in the other copy. The joining edges form a perfect matching.

The last definition of Q_n is the Cartesian product of n two-vertex complete graphs K_2. More generally the Cartesian product of copies of a complete graph is called a Hamming graph; the hypercube graphs are examples of Hamming graphs.

Thanks to Mathematica, we can easily draw a Hypurcube graph Q_n by typing **HypercubeGraph[n]**. For example, we just type **HypercubeGraph[3]** and **HypercubeGraph[4]** to obtain Q_3 and Q_4, respectively.

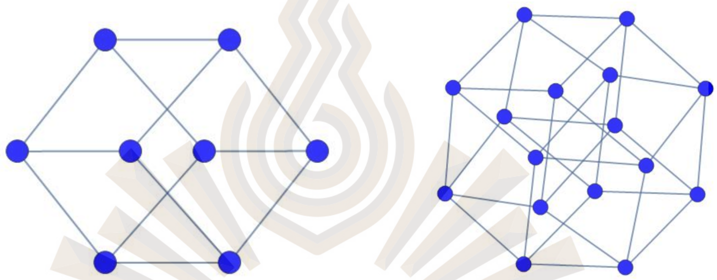

Figure 3.18: Hypercube graphs Q_3 and Q_4

Even we when we want to draw a big hypercube graph such as Q_6, we just type **HypercubeGraph[6]** to obtain the graph.

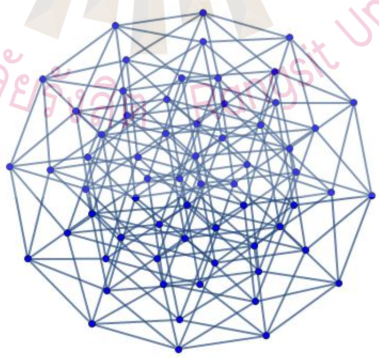

Figure 3.19: Hypercube graph Q_6

3.10 Grid graph

A *grid graph*, *lattice graph*, or *mesh graph*, is a graph whose drawing, embedded in some Euclidean space R^n, forms a regular tiling. This implies that the group of bijective transformations that send the graph to itself is a lattice in the group-theoretical sense. A k-dimensional grid graph with $n_1 \times n_2 \times \ldots n_k$ vertices are denoted by $G_{n_1,n_2,\ldots,n_k}$.

Thanks to Mathematica, we can easily draw a grid $G_{n_1,n_2,\ldots,n_k}$ by typing **GridGraph**[{$n_1, n_2, \ldots, n_k$}]. We can draw a grid graph $G_{3,3,3}$ and $G_{3,3,3,3}$ by typing **GridGraph**[**{3,3,3}**] and **GridGraph**[**{3,3,3,3}**]

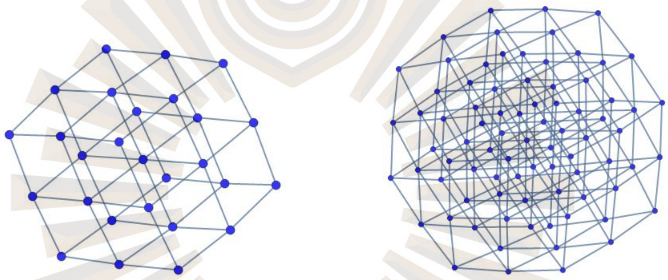

Figure 3.20: Grid graphs $G_{3,3,3}$ and $G_{3,3,3,3}$

Typically, no clear distinction is made between such a graph in the more abstract sense of graph theory, and its drawing in space (often the plane or $3D$ space). This type of graph may more shortly be called just a grid, lattice, or mesh. Moreover, these terms are also commonly used for a finite section of the infinite graph, as in "an 88 square grid".

The term lattice graph has also been given in the literature to various other kinds of graphs with some regular structure, such as the Cartesian product of a number of complete graphs.

A common type of a lattice graph (known under different names, such as square grid graph) is the graph whose vertices correspond to the points in the plane with integer coordinates, x-coordinates being in the range $1, 2, \ldots, n$, y-coordinates being in the range $1, 2, \ldots, ..., m$, and two vertices are connected by an edge whenever the corresponding points are at distance 1.

We can draw a grid graph $G_{3,6}$ and $G_{5,8}$ by typing **GridGraph[{3,6}]** and **GridGraph[{5,8}]**

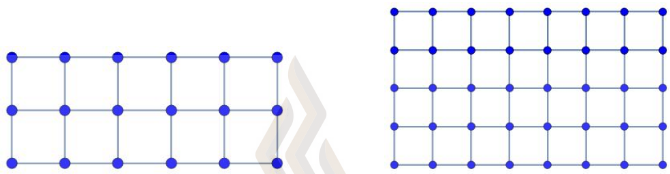

Figure 3.21: Grid graphs $G_{3,6}$ and $G_{5,8}$

Even a big graph such that $G_{10,20}$, we can obtain the graph in a second by typing $G_{10,20}$.

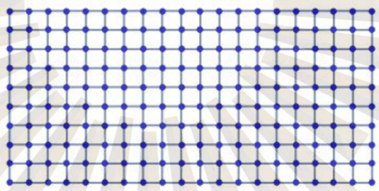

Figure 3.22: Grid graphs $G_{10,20}$

3.11 Knight tour graphs

A *knight's tour* is a sequence of moves of a knight on a chessboard such that the knight visits every square only once. If the knight ends on a square that is one knight's move from the beginning square (so that it could tour the board again immediately, following the same path), the tour is closed, otherwise it is open.

The knight's tour problem is the mathematical problem of finding a knight's tour. Creating a program to find a knight's tour is a common problem given to computer science students.[3] Variations of the knight's tour problem involve chessboards of different sizes than the usual 8×8, as well as irregular (non-rectangular) boards.

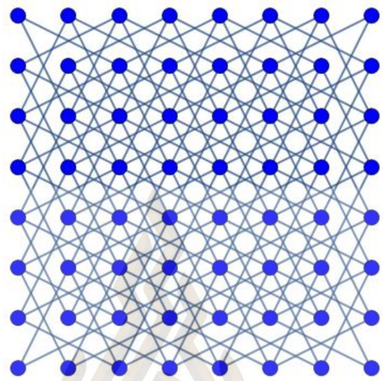

Figure 3.23: A knight's tour on 8×8 chessboard

In graph theory, a *knight graph*, or a *knight tour graph*, is a graph that represents all legal moves of the knight chess piece on a chessboard where each vertex represents a square on a chessboard and each edge is a legal move. More specifically, an $n \times m$ knight's tour graph is a knight's tour graph of an $n \times m$ chessboard.

For a $n \times m$ knight's tour graph the total number of vertices is simply nm. For a $n \times n$ knight's tour graph the total number of vertices is simply n^2 and the total number of edges is $4(n-2)(n-1)$.

To draw a knight graph, we need to think how to find a knight tour on $n \times m$ board. It is not easy and it may take a long time. However, we can find an $n \times m$ knight's graph in second by using the function **KnightTourGraph[n,m]**

For example, we can draw a 4×4 knight's graph and a 5×4 knight's graph by typing **KnightTourGraph[4,4]** and **KnightTourGraph[5,4]**, respectively.

The following figure is a 14×14 knight's graph crating by using mathematica.

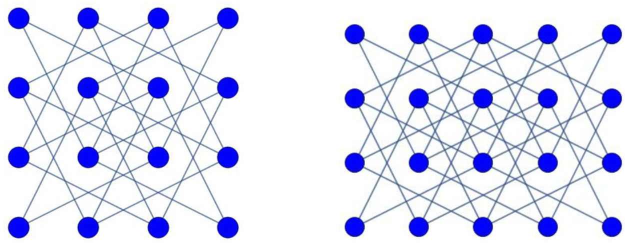

Figure 3.24: 4×4 knight's graph and 5×4 knight's graph

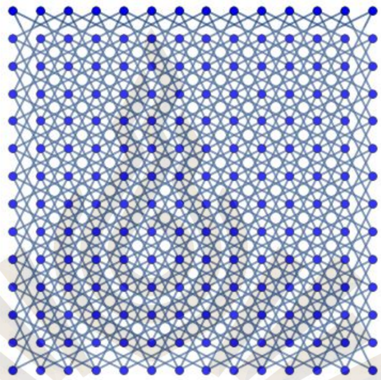

Figure 3.25: A 14×14 knight's graph

3.12 Harary graph

The *Harary graph* $H_{k,n}$ is a particular example of a k-connected graph with n graph vertices having the smallest possible number of edges. The smallest number of edges possible, as achieved by the Harary graph $H_{k,n}$, is $\lceil \frac{kn}{2} \rceil$.

We can easily draw the Harary graph $H_{k,n}$ by using the function **Harary-Graph[k,n]**. For example, we substitute k and n by 4 and 8, respectively to obtain Harary graph $H_{4,8}$.

Similarly, we can type**HararyGraph[6,11]** and **HararyGraph[8,15]** to obtain Harary graphs $H_{6,11}$ and $H_{8,15}$, respectively.

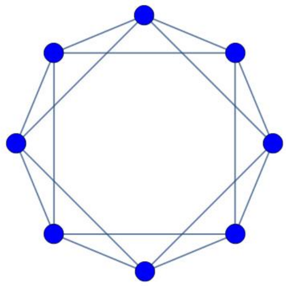

Figure 3.26: Harary graph $H_{4,8}$

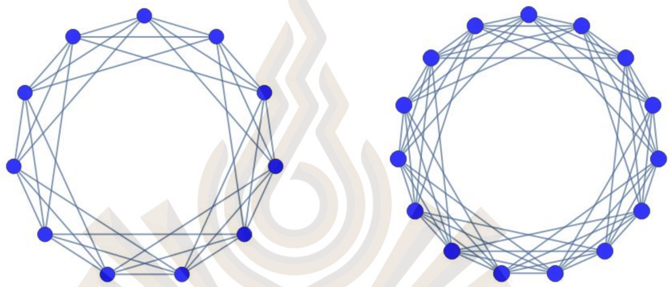

Figure 3.27: Harary graphs $H_{6,11}$ and $H_{8,15}$

3.13 Circulant graph

A *circulant graph* $C_n(k)$ is an n-vertex with $V(G) = \{v_1, v_2, \ldots, v_n\}$ such that v_i and v_j are adjacent if and only if $|i - j| = k$

We can draw a circulant graph $C_n(k)$ by using function **CirculantGraph[n,k]**.

For example, we can draw circulant graphs $C_8(3)$ and $C_{10}(4)$ by using functions **CirculantGraph[8,3]** and **CirculantGraph[10,4]** as showning the figure.

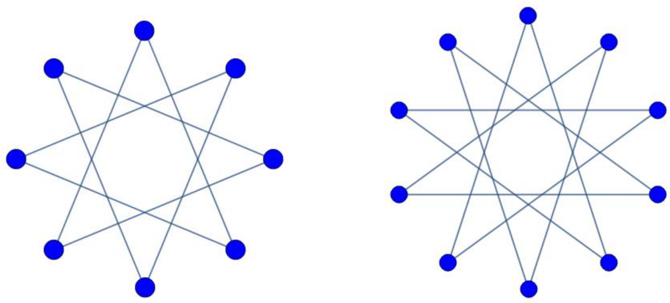

Figure 3.28: Circulant graphs $C_8(3)$ and $C_{10}(4)$

It is also easy when we want to create a beautiful big circulant graph such as $C_{30}(6)$ by using function **CirculantGraph[30,6]**.

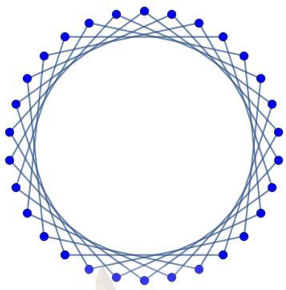

Figure 3.29: Circulant graph $C_{30}(6)$

A *generalized circulant graph* $C_n(j_1, j_2, \ldots)$ is an n-vertex with $V(G) = \{v_1, v_2, \ldots, v_n\}$ such that v_i and v_j are adjacent if and only if $|i - j| \in \{j_1, j_2, \ldots\}$.

Similarly, we can easily draw a generalized circulant graph $C_n(j_1, j_2, \ldots)$ by using function **CirculantGraph[n,**$\{j_1, j_2, \ldots\}$].

For example, we use **CirculantGraph[20,**$\{3, 5\}$] and **CirculantGraph[25,**$\{2, 4, 6\}$] to construct two generalized circulant graphs as shown in Figure 3.30

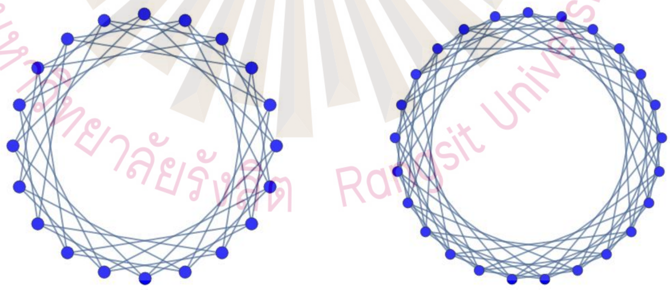

Figure 3.30: Generalized circulant graphs $C_{20}(\{3, 5\})$ and $C_{25}(\{2, 4, 6\})$

3.14 Peterson graph

In the mathematical field of graph theory ample for many problems in graph theory. The Petersen graph is named for Julius Petersen, who in 1898 constructed it to be the smallest bridgeless cubic graph with no three-edge-coloring.

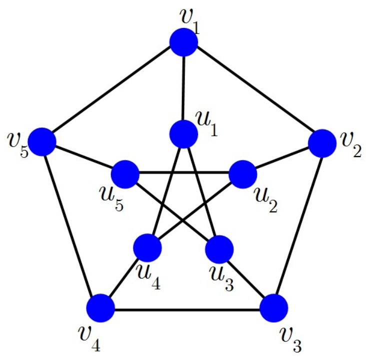

Figure 3.31: the Petersen graph

Although the graph is generally credited to Petersen; however, it first appeared 12 years befored, by A.B. Kempe (1886). Kempe noticed that its vertices can represent the ten lines of the Desargues configuration, and its edges represent pairs of lines that do not meet at one of the ten points of the configuration.

Donald Knuth says that the Petersen graph is *a remarkable configuration that serves as a counterexample to many optimistic predictions about what might be true for graphs in general.*

The Petersen graph is nonplanar. Any nonplanar graph has the complete graph K_5, or the complete bipartite graph $K_{3,3}$ as a minor, but the Petersen graph has both graphs as minors. The K_5 minor can be formed by contracting the edges of a perfect matching. The $K_{3,3}$ minor can be formed by deleting one vertex (for instance the central vertex of the 3-symmetric drawing) and contracting an edge incident to each neighbor of the deleted vertex.

The *generalized Petersen graphs* P_{n_k} are a family of cubic graphs formed by connecting the vertices of a regular n-gon to the corresponding vertices of a circulant graph $C_n(k)$. They include the Petersen graph and generalize one of the ways of constructing the Petersen graph. The generalized Petersen graph family was introduced in 1950 by H. S. M. Coxeter and these graphs were given their name in 1969 by Mark Watkins.

Thanks to Mathematica, we can easily draw generalized Petersen graphs $P_{n,k}$ by using the function **PetersenGraph[n,k]**.

For instant, we can apply the functions **PetersenGraph[7,3]** and **PetersenGraph[8,5]** to construct generalized Petersen graphs $P_{7,3}$ and $P_{8,5}$, respectively.

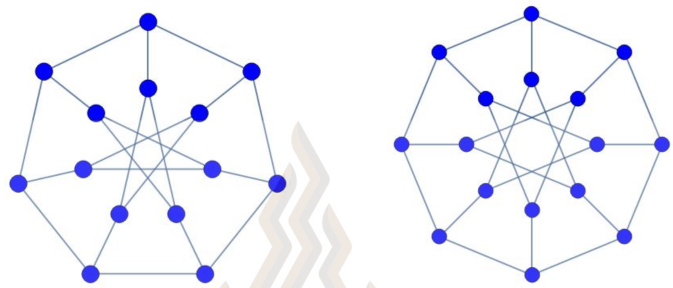

Figure 3.32: Generalized Petersen graphs $P_{7,3}$ and $C_{8,5}$

Suppose that we would like to draw a big generalized Petersen graph such as $P_{25,7}$. We just type **PetersenGraph[25,7]** to draw the graph.

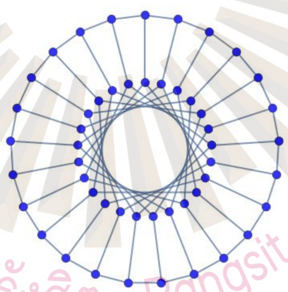

Figure 3.33: The generalized Petersen graph $P_{25,7}$

CHAPTER IV

Game coloring

4.1 Definition and background

Two players, Alice and Bob, alternatively color vertices of a graph using a fixed set of colors with Alice staring first so that no two adjacent vertices receive the same color. Alice wins if all vertices are successfully colored and Bob win if, at some time before all vertices is completely colored, one of the players has no legal move. The *game chromatic number* of a graph G, denoted by $\chi_g(G)$, is the least number of colors such that Alice has a winning strategy.

The well-known game coloring was invented by Steven J. Bram and was published in 1981 by Martin Gardner [6]. Bodlaender [2] reinvented this game in 1991. Define $\chi_g(\mathcal{G}) = \max\{\chi_g(G)|G \in \mathcal{G}\}$. The game chromatic number of several classes of graphs are investigated. For example, $\chi_g(\mathcal{F}) = 4$ when $\mathcal{F}$ is the class of forests [4], $6 \leq \chi_g(\mathcal{OP}) \leq 7$ when $\mathcal{OP}$ is the class of outerplanar graphs [7] [9], $8 \leq \chi_g(\mathcal{P}) \leq 17$ when $\mathcal{P}$ is the class of planar graphs [9] [14], $\chi_g(\mathcal{KT}) = 3k+2$ when $k \geq 2$ and $\mathcal{KT}$ is the class of k-trees [12] [15].

In 2007, Bartnicki, Brear, Grytczuk, Kove, Miechowicz and Peterin [1] investigated the game chromatic number of the Cartesian product of two graphs. The *Cartesian product* $G\Box H$ is the graph with vertex set $V(G) \times V(H)$ where two vertices (u_1, v_1) and (u_2, v_2) are adjacent if and only if either $u_1 = u_2$ and $v_1v_2 \in E(H)$ or $v_1 = v_2$ and $u_1u_2 \in E(G)$. In [1], the authors proved that $\chi_g(G\Box H)$ is not bounded above in term of $\chi_g(G)$ and $\chi_g(H)$. However, Zhu [13] found the the upper bound of $\chi_g(G\Box H)$ in terms of the game chromatic number of G and acyclic chromatic number of H. In [1], Bartnicki et. al. also gave the exact values of $\chi_g(P_2\Box P_n)$, $\chi_g(P_2\Box C_n)$ and $\chi_g(P_2\Box K_n)$. Later, Sia [11] found the exact value of $\chi_g(S_m\Box P_n), \chi_g(S_m\Box C_m), \chi_g(P_2\Box W_n)$ and $\chi_g(P_2\Box K_{m,n})$. In [10], the author proved that $\chi_g(C_m\Box C_n) \leq 5$ and $\chi_g(C_{2m}\Box C_n) = 5$ for $m \geq 3$ and $n \geq 7$.

The *degree* of a vertex v, denoted by $d(v)$, of a graph is the number of edge incident to v. The *maximum degree* of a graph G, denoted by $\Delta(G)$, is the maximum degree of its vertices.

The next remark will introduce a relation between $\chi_g(G)$ and $\Delta(G)$.

Remark 4.1. Let G be a graph. Then $\chi_g(G) \leq \Delta(G) + 1$.

Proof. During the game, suppose that a player want to color a vertex, say v. Since $d(v) \leq \Delta(G) < \Delta(G) + 1$, there is available color c for v. Hence, the player can label vertex v by using color c. $\square$

4.2 The game chromatic number of paths

A *path* in a graph is a finite or infinite sequence of edges which connect a sequence of vertices which, by most definitions, are all distinct from one another. A path with n vertices is denoted by P_n

Theorem 4.2. *Let n be a positive number and P_n be a path with n vertices Then*

$$\chi_g(P_n) = \begin{cases} 1 & ; n = 1 \\ 2 & ; n = 2, 3 \\ 3 & ; n \geq 4 \end{cases}$$

Proof. Denote the vertices of the fiber of P_n by $u_1, u_2, \ldots, u_n$

Case 1. $n = 1$. Alice chooses any color to label u_1. Then her goal is achieved.

Case 2. $n = 2$. To label P_2, it requires at least two colors because u_1 and u_2 need different colors. Suppose that there are two available colors. First, Alice labels u_1 by color 1. By the game rule, Bob must use another color to label u_2; hence, Alice can achieves her goal.

Case 3. $n = 3$. To label P_3, it also requires at least two colors. Suppose that there are two available colors. First, Alice labels u_2 by color 1. Without loss of generality, Bob has to label u_1 by color 2. Finally, Alice labels u_3 by color 2; hence, Alice wins.

Case 4. $n \geq 4$. If there are three available colors. All vertices of P_n can be labeled because $\Delta(P_n) = 2 < 3$. Suppose that there are only two available colors. On the first move, if Alice labels v_i by color 1, then Bob chooses to label v_{i-2} or v_{i+2} by color 2. Then, v_{i-1} or v_{i+1} cannot be labeled. Hence, Bob wins. □

4.3 The game chromatic number of stars

A *star* S_k is the complete bipartite graph $K_{1,k-1}$; a tree with one internal node and $k-1$ leaves

Theorem 4.3. *Let n be a positive number and S_n be a star with n vertices. Then*

$$\chi_g(S_n) = 2$$

Proof. Let n be a positive number and S_n be a star with n vertices. It is obvious that $\chi_g(S_n) \geq 2$. It remains to show that $\chi_g(S_n) \leq 2$. Suppose that there are two available colors. Alice first label the center vertex by color 1. Then all remaining leaves can be labeled by color 2. □

4.4 The game chromatic number of complete graphs

A *complete graph* is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge. The complete graph on n vertices is denoted by K_n.

Theorem 4.4. *Let n be a positive number and K_n be a complete graph with n vertices Then*

$$\chi_g(K_n) = n$$

Proof. The proof is quite straightforward because all n vertices are mutually adjacent. Hence, it requires at least n colors to label all vertices of K_n. Obviously, an n-vertex graph requires at most n colors. That is, the game chromatic number of K_n is always n. □

4.5 The game chromatic number of cycles

A *cycle* or *circular graph* is a graph that consists of a single cycle, or in other words, some number of vertices connected in a closed chain. The cycle graph with n vertices is called C_n.

Theorem 4.5. *Let $n \geq 3$ be a positive number and C_n be a cycle with n vertices Then*

$$\chi_g(C_n) = 3$$

Proof. By Theorem 4.4, we obtain that $\chi_g(C_3) = \chi_g(K_3) = 3$.

Let $n \geq 4$. By Remark 4.1, we obtain that $\chi_g(C_n) \leq \Delta(C_3) + 1 = 3$.

Suppose that there is only two available colors. When Alice label the first vertex by color 1, Bob choose to label a vertex with distance two from v by color 2. Hence, the middle vertex requires the third color. Therefor, Bob wins.

In conclusion, we obtain that $\chi_g(C_n) = 3$. □

4.6 The game chromatic number of Petersen graph

Theorem 4.6. *The game chromatic number of Petersen graph is* 4.

Proof. Let G be the Petersen graph as shown in the Figure 4.1

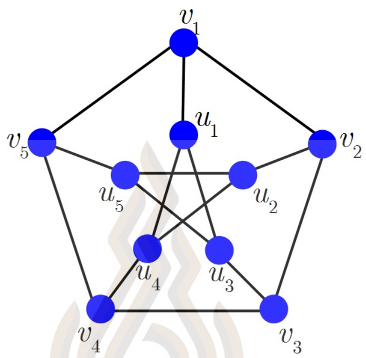

Figure 4.1: the Petersen graph

If there are four available colors, then all vertices can be labeled because of $\Delta(G) = 3$. Hence, Alice wins. Suppose that there are only three available colors. We will prove that Bob wins.

Without loss of generality, suppose that Alice first labels v_1 by color 1. We divide Alice's next move into three cases.

Case 1. Alice labels neither v_2 or u_2. Then Bob labels u_2 by color 3. Therefore, there is no available color for v_2. Hence, Bob wins.

Case 2. Alice labels u_2. Alice cannot label u_2 by color 3; otherwise, there is no color for v_2. Without loss of generality, suppose that Alice labels u_2 by color 1. Then Bob label v_4 by color 3. On the next move, if Alice labels neither v_5 nor u_5, then Bob labels u_5 by color 2. Hence, there is no color for v_5. If Alice labels v_5 or u_5, then Bob labels u_1 by color 2. Hence, there is no color for u_4. That is, Bob wins.

Case 3. Alice labels v_2. The color on v_2 must be color 3. Then Bob labels u_4 by color 2. On the next move, if Alice labels neither u_1 nor u_3, then Bob labels

u_3 by color 3. Hence, there is no color for u_1. If Alice labels u_1 or u_3, then Bob labels u_5 by color 3. Hence, there is no color for u_2. That is, Bob wins. □

4.7 The game chromatic number of wheel graphs

A *wheel graph* W_n is a graph with $n+1$ vertices (n3), obtained from connecting a single vertex to all vertices of a cycle with n vertices. Some authors instead use n to refer to the number of vertices of wheel graphs, so that their W_n is the graph that we denote W_{n-1}. The chromatic number of W_n is three when n is even and four when n is odd. However, the game chromatic number of W_n is four unless $n = 4$.

Theorem 4.7. *Let $n \geq 3$ be a positive number and W_n be a wheel graph with $n+1$ vertices. Then*

$$\chi_g(W_n) = \begin{cases} 3 & ; n = 4 \\ 4 & ; \textit{otherwise} \end{cases}$$

Proof. Let $V(W_n) = \{u, v_1, v_2, \ldots, v_n\}$ where u is the center of W_n.

Case 1. $n = 3$. We obtain that $\chi_g(W_4) = \chi_g(K_4) = 4$.

Case 2. $n = 4$. Notice that W_4 has K_3 as a subgraph. It requires at least three colors to label all vertices of W_4. It remains to show that Alice has a winning strategy when there are at least three colors.

Alice first labels the center by color 1. Without loss of generality, Bob labels v_1 by color 2. On the next move, Alice labels v_3 by color 2. Then remaining vertices can be labeled by color 3. Therefore, $\chi_g(W_4) = 3$.

Case 3. $n \geq 4$. The proof is divided into two parts.

Suppose that there are four available colors. Alice first labels the center by color 1. Notice that the remaining vertices form a cycle with n vertices. Then all remaining vertices can be labeled because $\Delta(C_n) = 2$ but there are three remaining colors for each remaining vertex. Hence, Alice wins.

Suppose that there are only three available colors.

Case 3.1. The first vertex that Alice labels is not the center. Without loss of generality, suppose that Alice labels v_1 by color 1. Bob labels v_3 by color 2. Notice that both v_2 and u are adjacent to v_1 and v_3. Moreover, v_2 and u need different colors because they are adjacent. That is, v_2 and u require the third and the fourth colors. It is impossible to color all vertices of W_n. Then Bob wins.

Case 3.2. The first vertex, that Alice labels, is the center. Suppose that Alice labels the center by color 1. Without loss of generality, Bob labels v_1 by color 2. On the next move, Bob choose to label v_3 or v_{n-1} by color 3. Hence, the middle vertex v_2 or v_n requires the fourth color. It is impossible to color all vertices of W_n. Then Bob wins. □

Before talking about generalized wheel graphs, we need to mention a graph operation callled *the join of graphs*. The *join of graphs* G and H, written $G \vee H$, is the graph obtained from G and H by adding the edges between all vertices of G and all vertices of H. Here, We investigated the game chromatic numner of $G \vee H$.

We can say that a wheel graph W_n is obtained from $K_1 \vee C_n$. Then generalized wheel graphs is $K_m \vee C_n$.

Lemma 4.8. *If $m \geq 4$, then $\chi_g(K_m \vee C_n) \leq 2m - 1$.*

Proof. Let $m \geq 4$. Suppose that there are $2m - 1$ colors. At most Alice's first m turn, she can make all vertices of K_m be labeled. Each vertex of C_n has at least three available colors. Then the remaining vertices can always be labeled. Therefore, Alice wins. □

Theorem 4.9. *Let m, n be a positive number where $n \geq 3$ and C_n be a cycle with n vertices. Then*

$$\chi_g(K_m \vee C_n) = \begin{cases} m+2 & ;\textit{if } n = 4 \textit{ and } m \textit{ is odd} \\ m+3 & ;\textit{if } n = 3, \textit{ or } n = 4 \textit{ and } m \textit{ is even, or } n \geq 5 \textit{ and } m \leq 3 \\ 2m-1 & ;\textit{if } n \geq 5 \textit{ and } n \geq 4 \textit{ and } m \leq \lceil \frac{n}{2} \rceil + 1 \\ m + \lceil \frac{n}{2} \rceil & ;\textit{if } n \geq 5 \textit{ and } n \geq 4 \textit{ and } m \geq \lceil \frac{n}{2} \rceil + 2 \end{cases}$$

Proof. Let m, n be a positive number where $n \geq 3$ and C_n be a cycle with n vertices. Let $V(K_m) = \{u_1, u_2, \ldots, u_m\}$ and $V(C_n) = \{u_1, u_2, \ldots, u_n\}$.

Case 1. $n = 3$. We obtain that $\chi_g(K_m \vee C_3) = \chi_g(K_{m+3}) = m + 3$.

Case 2. $n = 4$ and m is odd. Notice that $K_m \vee C_4$ has K_{m+2} as a subgraph. It requires at least $m + 2$ colors to label all vertices of $K_m \vee C_4$. It remains to show that Alice has a winning strategy when there are at least $m + 2$ colors.

Since m is odd, Alice can force Bob to first label a vertex of C_4, say v_1 Whenever Bob first labels v_1, Alice labels v_3 by using the same color. Moreover, when Bob first labels a vertex of $\{v_2, v_4\}$, Alice labels the remaining vertices by the same color. Hence, $m + 2$ colors are enough.

Case 3. $n = 4$ and m is even. It requires to show that Bob wins when there are $m + 2$ colors and Alice wins when there are $m + 3$ colors.

Suppose that there are $m+2$ colors. Because of m is even, Bob can force Alice to first label a vertex of C_4, say v_1 Whenever Alice first labels v_1, Bob labels v_3 by using a different color. Then v_2 needs the third colors. Hence, C_4 requires at least 3 colors and K_m requires another m colors. Hence, $m + 2$ colors are not enough. That is, Bob wins.

Suppose that there are $m + 3$ colors. If Bob first labels a vertex from C_n, say v_1, then Alice labels v_3 by the same color. Hence, all vertices of $K_m \vee C_4$ can be labeled by $m+3$ colors. Otherwise, Bob will try to force Alice to first label a vertex from C_n by keep labeling only vertices of K_n. Alice also keep labeling vertices of K_m. Then their m first turns are only on vertices of K_m. Consequently, the remaining vertices are C_4 and each vertex has three available colors. Hence, the remaining vertices can always be labeled because of $\Delta(C_4) = 2$. That is, Alice wins.

Case 4. $n \geq 5$ and $m \leq 3$. On the first turn, if Alice labels a vertex from C_n, say v_1, then Bob labels v_3 by a different color. Hence, C_n requires three colors. If Alice does not label a vertex of C_n, then Bob labels v_1. On his next turn he labels v_3 or v_{n-1} by a different color to confirm that C_n requires three colors. Therefore, $K_m \vee C_n$ requires $m+3$ colors.

Suppose that there are $m+3$ colors. At most Alice's first m turn, she can make all vertices of K_m be labeled. Each vertex of C_n has at least three available colors. Then the remaining vertices can always be labeled. Hence, Alice wins.

Case 5. $n \geq 5$ and $n \geq 4$ and $m \leq \lceil \frac{n}{2} \rceil + 1$.

By Lemma 4.8, it remains to show that Bob wins when there are $2m-2$ colors. Suppose that there are only $2m-2$ colors. On Bob's first $m-1$ turns, he will label $m-1$ vertices of C_n by using different $m-1$ colors. The goal can be achieved because $m-1 \leq \lceil \frac{n}{2} \rceil$. After reaching the goal, there are only $m-1$ color for all vertices of K_m. Hence, Bob wins.

Case 6. if $n \geq 5$ and $n \geq 4$ and $m \geq \lceil \frac{n}{2} \rceil + 2$

Suppose that there are only $m-1+\lceil \frac{n}{2} \rceil$ colors. On all Bob's first $\lceil \frac{n}{2} \rceil$ turns, he will label $\lceil \frac{n}{2} \rceil$ vertices of C_n by using $\lceil \frac{n}{2} \rceil$ different colors. He may finish the goal before $\lceil \frac{n}{2} \rceil$ turns if Alice help him label C_n. If he achieves the goal then C_n uses at least $\lceil \frac{n}{2} \rceil$ colors. Then $K_m \vee C_n$ requires at least $m+\lceil \frac{n}{2} \rceil$ colors. If he runs out of colors, then there exists a vertex of K_m which is not yet labeled because

$m \geq \lceil \frac{n}{2} \rceil + 2$. There is no color for the vertex; hence, Bob wins.

Suppose that there are $m + \lceil \frac{n}{2} \rceil$. Alice keep labeling vertices of K_m until all vertices of K_m are labeled or Bob first labels a vertex of C_n. If All vertices of K_m are labeled, then all vertices of C_n can be labeled because of $\lceil \frac{n}{2} \rceil \geq 3$. If Bob first labels a vertex of C_n, then Alice keep labeling the remaining vertices of C_n by using the same colors. Alice goal is that all vertices of C_n be labeled by at most $\lceil \frac{n}{2} \rceil$ colors. Since she can always reach the goal, there are m remaining colors for K_m. Hence, Alice wins. $\square$

CHAPTER V

Conclusions and future work

5.1 Conclusions

In this research repost, we give the exact value of the game chromatic number of paths, stars, cycles, complete graphs, Petersen graphs, wheel graphs and generalized wheel graphs.

Theorem 5.1. *Let n be a positive number and P_n be a path with n vertices Then*

$$\chi_g(P_n) = \begin{cases} 1 & ; n = 1 \\ 2 & ; n = 2, 3 \\ 3 & ; n \geq 4 \end{cases}$$

Theorem 5.2. *Let n be a positive number and S_n be a star with n vertices. Then*

$$\chi_g(S_n) = 2$$

Theorem 5.3. *Let n be a positive number and K_n be a complete graph with n vertices Then*

$$\chi_g(K_n) = n$$

Theorem 5.4. *Let $n \geq 3$ be a positive number and C_n be a cycle with n vertices Then*

$$\chi_g(C_n) = 3$$

Theorem 5.5. *The game chromatic number of Petersen graph is* 4*.*

Theorem 5.6. *Let $n \geq 3$ be a positive number and W_n be a wheel graph with $n+1$ vertices. Then*

$$\chi_g(W_n) = \begin{cases} 3 & ; n = 4 \\ 4 & ; otherwise \end{cases}$$

Theorem 5.7. *Let m, n be a positive number where $n \geq 3$ and C_n be a cycle with n vertices. Then*

$$\chi_g(K_m \vee C_n) = \begin{cases} m+2 & ; \textit{if } n = 4 \textit{ and } m \textit{ is odd} \\ m+3 & ; \textit{if } n = 3, \textit{ or } n = 4 \textit{ and } m \textit{ is even, or } n \geq 5 \textit{ and } m \leq 3 \\ 2m-1 & ; \textit{if } n \geq 5 \textit{ and } n \geq 4 \textit{ and } m \leq \lceil \frac{n}{2} \rceil + 1 \\ m + \lceil \frac{n}{2} \rceil & ; \textit{if } n \geq 5 \textit{ and } n \geq 4 \textit{ and } m \geq \lceil \frac{n}{2} \rceil + 2 \end{cases}$$

5.2 Future Work

The Game chromatic number of the following classes of graphs are still open

1. Grid graphs
2. Hypercube graphs
3. Complete multipartite graphs
4. Knigh tour graphs
5. Harary graphs
6. Turan graphs
7. Circulant graphs
8. Generalized Petersen graphs

The *Cartesian product* $G\Box H$ is the graph with vertex set $V(G)\times V(H)$ where two vertices (u_1, v_1) and (u_2, v_2) are adjacent if and only if either $u_1 = u_2$ and $v_1v_2 \in E(H)$ or $v_1 = v_2$ and $u_1u_2 \in E(G)$. Grid graphs can be written in the form of $P_n\Box P_n$ where $\Box$. In [1], Bartnicki et. al. also gave the exact values of $\chi_g(P_2\Box P_n)$. Howover, $\chi_g(P_n\Box P_n)$ is not yet investigated.

In [1], the authors gave a conjecture that for a hypercube graph Q_n, $\chi_g(Q_n) = n + 1$. The conjecture is still open, as well.

For the remainnig classes of graphs, there is no one mention about its game chromatic number.

REFERENCES

[1] Bartnicki, T., Bresar, B., Grytczuk, J., Kovse, M., Miechowicz, Z. and Peterin, I., Game chromatic number of Cartesian product graphs, **Electron. J. Combin.** 15(1) (2008), R72.

[2] Bodlaender, H.L., On the complexity of some coloring games, **Internat. J. Found. Comput. Sci.** 2 (1991): 133-147.

[3] Deitel, H.M., and Deitel, P.J., Java How To Program Fifth Edition Prentice Hall, Upper Saddle River, New Jersey, 326328. (2003).

[4] Faigle, U., Kern, U., Kierstead, H.A., and Trotter, W.T., On the game chromatic number of some classes of graphs, **Ars Combin.** 35 (1993): 143150.

[5] Fertin, G., Raspaud, A., and Reed, B., Star coloring of graphs, **J. Graph Theory** 47(3) (2004): 163182.

[6] Gardner, M., Mathematical games, Scientic American 23, (1981).

[7] Guan, D., and Zhu, X., The game chromatic number of outerplanar graphs, **J. Graph Theory** 30 (1999): 6770.

[8] Hakimi, S.L., Mitchem, J., and Schmeichel, E.E., Star arboricity of graphs, **Discrete Math.** 149 (1996): 9398.

[9] Kierstead, H.A., and Trotter, W.T., Planar graph colouring with an uncooperative partner, **J. Graph Theory** 18(6) (1994): 569584.

[10] Raspaud, A., and Wu, J., Game chromatic number of toroidal grids, **Inform. Process. Lett.** 109 (2009): 11831186

[11] Sia, C., The game chromatic number of some families of Cartesian product graphs, **AKCE Int. J. Graphs Comb.** 6(2) (2009): 315-327.

[12] Wu, J. and Zhu, X., Lower bounds for the game colouring number of partial k-trees and planar graphs, **Discrete Math.** 308 (2008): 2637-2642.

[13] Zhu, X., Game coloring the Cartesian product of graphs. **J. Graph Theory**, 59 (2008): 261-278.

[14] Zhu, X., Refined activation strategy for the marking game, **J. Combin. Theory Ser. B** 98 (2008): 118.

[15] Zhu, X., The game coloring number of pseudo partial k-trees, **Discrete Math.** 215 (2000): 245-262.

APPENDIX

A partial result of this research report is submitted to public to Far East Journal of Mathematical Sciences.

Far East Journal of Mathematical Sciences (FJMS)

Published Online: September 2016
http://dx.doi.org/10.17654/MS100071041
Volume 100, Number 7, 2016, Pages 1041-1054 **ISSN: 0972-0871**

THE GAME CHROMATIC NUMBER OF GENERALIZED WHEEL GRAPHS

Siriwan Wasukree

Department of Mathematics
College of Information Technology and Communication
Rangsit University
Pathumthani 12000, Thailand
e-mail: siriwan.w@rsu.ac.th

Abstract

In this paper, we investigate exact values of the game chromatic numbers of wheel graphs and also of generalized wheel graphs.

1. Introduction

Two players, Alice and Bob, alternatively color vertices of a graph using a fixed set of colors with Alice starting first so that no two adjacent vertices receive the same color. Alice wins if all vertices are successfully colored and Bob wins if, at the some time before all vertices are completely colored, one of the players has no legal move. The *game chromatic number* of a graph *G*, denoted by $\chi_g(G)$, is the least number of colors such that Alice has a winning strategy.

Received: April 19, 2016; Revised: June 14, 2016; Accepted: June 29, 2016
2010 Mathematics Subject Classification: 05C15.
Keywords and phrases: game coloring, game chromatic number, wheel graph.
Communicated by K. K. Azad

The well-known game coloring was invented by Steven J. Bram and was published in 1981 by Gardner [5]. Bodlaender [4] reinvented this game in 1991. Define $\chi_g(\mathcal{G}) = \max\{\chi_g(G) | G \in \mathcal{G}\}$. The game chromatic numbers of several classes of graphs are investigated. For example, $\chi_g(\mathcal{F}) = 4$ when $\mathcal{F}$ is the class of forests [7], $6 \leq \chi_g(\mathcal{OP}) \leq 7$ when $\mathcal{OP}$ is the class of outerplanar graphs [8, 9], $8 \leq \chi_g(\mathcal{P}) \leq 17$ when $\mathcal{P}$ is the class of planar graphs [9, 10], $\chi_g(\mathcal{KT}) = 3k + 2$ when $k \geq 2$ and $\mathcal{KT}$ is the class of k-trees [11, 12].

In this paper, the game chromatic numbers of several classes of graphs are investigated; for example, paths, complete graphs, stars, cycles, wheel graphs and generalized wheel graphs.

2. Preliminaries

Throughout the paper, G denotes a simple, undirected, finite, connected graph; $V(G)$ and $E(G)$ are the vertex set and the edge set of G, respectively.

A *path* in a graph is a finite or infinite sequence of edges connecting a sequence of vertices which, by most definitions, are all distinct from one another. A path with n vertices is denoted by P_n.

Figure 2.1. Path P_5.

Remark 2.1. Let n be a positive integer and P_n be a path with n vertices. Then

$$\chi_g(P_n) = \begin{cases} 1; & n = 1 \\ 2; & n = 2, 3 \\ 3; & n \geq 4. \end{cases}$$

Proof. Denote the vertices of the fiber of P_n by $u_1, u_2, \ldots, u_n$.

Case 1. $n = 1$. Alice chooses any color to label u_1. Then her goal is achieved.

Case 2. $n = 2$. To label P_2, it requires at least two colors because u_1 and u_2 need different colors. Suppose that there are two available colors. First, Alice labels u_1 by color 1. By the game rule, Bob must use another color to label u_2; hence, Alice can achieve her goal.

Case 3. $n = 3$. To label P_3, it also requires at least two colors. Suppose that there are two available colors. First, Alice labels u_2 by color 1. Without loss of generality, Bob has to label u_1 by color 2. Finally, Alice labels u_3 by color 2; hence, Alice wins.

Case 4. $n \geq 4$. If there are three available colors, then all vertices of P_n can be labeled because $\Delta(P_n) = 2 < 3$. Suppose that there are only two available colors. On the first move, if Alice uses color 1 to label a vertex, say u, then Bob uses color 2 to label a vertex with distance two from u. Then the middle vertex requires the third color. Hence, Bob wins. □

A *complete graph* is a simple graph in which all vertices are connected. The complete graph with n vertices is denoted by K_n.

Remark 2.2. Let n be a positive integer and K_n be a complete graph with n vertices. Then

$$\chi_g(K_n) = n.$$

Proof. The proof is quite straightforward because all vertices are mutually adjacent. Hence, it requires at least n colors to label all vertices of K_n. Obviously, an n-vertex graph requires at most n colors. That is, the game chromatic number of K_n is always n. □

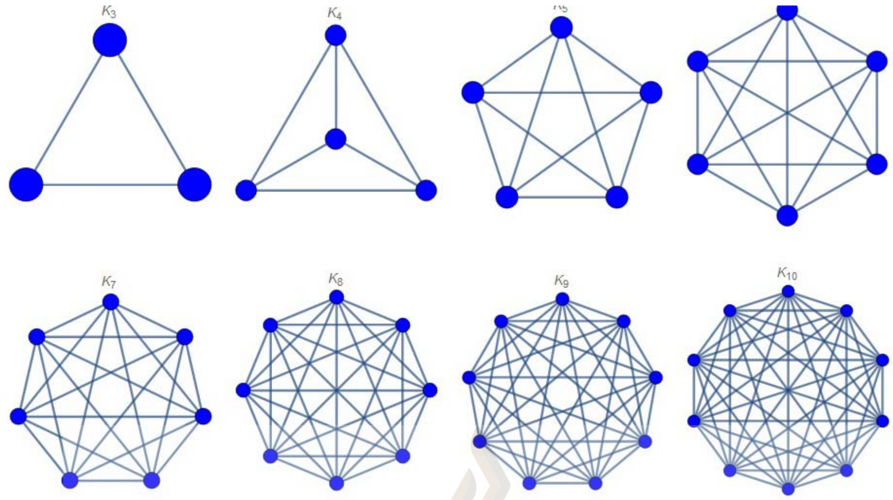

Figure 2.2. Complete graphs.

A *star* S_k is the complete bipartite graph $K_{1,k-1}$; a tree with one internal node and $k-1$ leaves (but, no internal nodes and k leaves when $k \leq 2$). Alternatively, some authors define S_k to be the tree of order $k+1$ with maximum diameter 2; in which case a star of $k > 3$ has $k-1$ leaves.

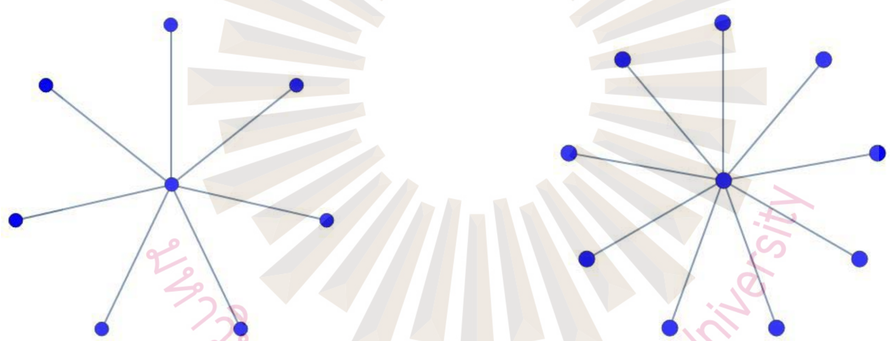

Figure 2.3. Stars S_8 and S_{10}.

Remark 2.3. Let n be a positive integer where $n \geq 2$ and S_n be a star with n vertices. Then

$$\chi_g(S_n) = 2.$$

Proof. Let n be a positive integer where $n \geq 2$ and S_n be a star with n vertices. It is obvious that $\chi_g(S_n) \geq 2$. It remains to show that $\chi_g(S_n) \leq 2$. Suppose that there are two available colors. Alice first labels the center vertex by color 1. Then all remaining leaves can be labeled by color 2. □

A *cycle* is a graph that consists of a single cycle, or in other words, some number of vertices connected in a closed chain. The cycle graph with n vertices is called C_n. The number of vertices in C_n equals the number of edges, and every vertex has degree 2; that is, every vertex has exactly two edges incident with it.

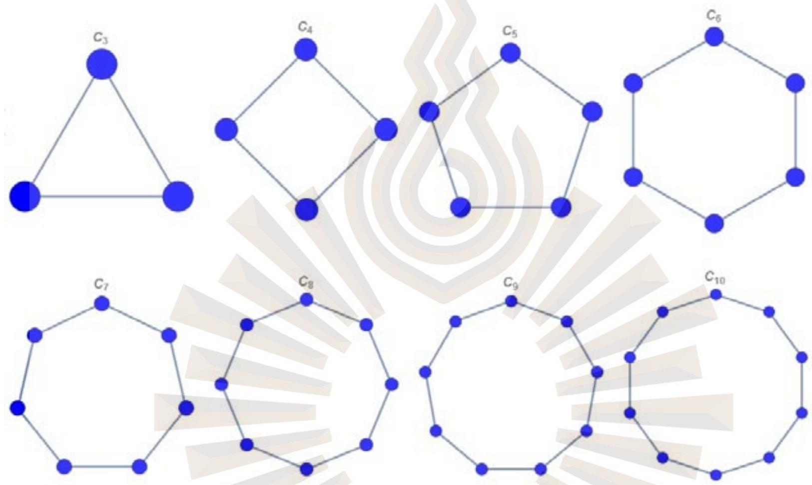

Figure 2.4. Cycles.

Remark 2.4. Let $n \geq 3$ be a positive integer and C_n be a cycle with n vertices. Then

$$\chi_g(C_n) = 3.$$

Proof. By Remark 2.2, we obtain that $\chi_g(C_n) = \chi_g(K_n) = 3$.

Let $n \geq 4$. Because of $\Delta(C_n) = 2$, all vertices of C_n can always be labeled if there are three available colors. That is, Alice always wins if there are three available colors.

Suppose that there are only two available colors. On the first move, if Alice uses color 1 to label a vertex, say u, then Bob uses color 2 to label a vertex with distance two from u. Then the middle vertex requires the third color. Hence, Bob wins.

In conclusion, we obtain that $\chi_g(C_n) = 3$. □

3. Wheel Graphs and Generalized Wheel Graphs

A *wheel graph* W_n is a graph with $n+1$ vertices where $n \geq 3$, obtained from connecting a single vertex to all vertices of a cycle with n vertices.

Theorem 3.1. *Let $n \geq 3$ be a positive integer and W_n be a wheel graph with $n+1$ vertices. Then*

$$\chi_g(W_n) = \begin{cases} 3; & n = 4 \\ 4; & \text{otherwise.} \end{cases}$$

Proof. Let $V(W_n) = \{u, v_1, v_2, \ldots, v_n\}$, where u is the center of W_n.

Case 1. $n = 3$. By Remark 2.2, we obtain that $\chi_g(W_4) = \chi_g(K_4) = 4$.

Case 2. $n = 4$. Notice that W_4 has K_3 as a subgraph. It requires at least three colors to label all vertices of W_4. It remains to show that Alice has a winning strategy when there are at least three colors.

Alice first labels the center by color 1. Without loss of generality, Bob labels v_1 by color 2. On the next move, Alice labels v_3 by color 2. Then remaining vertices can be labeled by color 3. Therefore, $\chi_g(W_4) = 3$.

Case 3. $n \geq 4$. The proof is divided into two parts. The first part is that Alice wins when there are four colors. The other part is that Bob wins when there are three colors.

Suppose that there are four available colors. Alice first labels the center by color 1. Notice that the remaining vertices form a cycle with n vertices.

Then all remaining vertices can be labeled because $\Delta(C_n) = 2,$ but there are three remaining colors for each remaining vertex. Hence, Alice wins.

Suppose that there are only three available colors. On the first move, Alice has two choices. She labels either a non-center vertex or a center vertex.

Case 3.1. Alice labels a non-center vertex on the first move. Without loss of generality, suppose that Alice labels v_1 by color 1. Then Bob labels v_3 by color 2. Notice that both v_2 and u are adjacent to v_1 and v_3. Moreover, v_2 and u need different colors because they are adjacent. That is, v_2 and u require the third and the fourth colors. It is impossible to color all vertices of W_n. Then Bob wins.

Case 3.2. Alice labels a center vertex on the first move. Suppose that Alice labels the center by color 1. Then Bob labels v_1 by color 2. On the next move, Bob chooses to label v_3 or v_{n-1} by color 3. Hence, the middle vertex v_2 or v_n requires the fourth color. It is impossible to color all vertices of W_n. Then Bob wins. □

Before talking about generalized wheel graphs, we need to mention a graph operation called *the join of graphs*. The *join of graphs* G and H, written $G \vee H$, is the graph obtained from G and H by adding the edges between all vertices of G and all vertices of H.

Hence, a wheel graph W_n is isomorphic to $K_1 \vee C_n$. A *generalized wheel graph*, denoted by $K_m \vee C_n$ is obtained from connecting all vertices of a complete graph K_m to all vertices of a cycle C_n.

Lemma 3.2. *If* $m \geq 4$, *then* $\chi_g(K_m \vee C_n) \leq 2m - 1$.

Proof. Let $m \geq 4$. It suffices to show that Alice wins when there are $2m - 1$ colors. Suppose that there are $2m - 1$ colors. On Alice first turn, she labels a vertex from K_m by color 1. Then she keeps labeling only vertices from K_m by new colors. During at most Alice's first m turns, Alice can reach her goal because there are $2m-1$ colors. That is, all vertices of K_m can be labeled by m colors; hence, there are $m-1$ colors for all vertices of C_n. Because of $m-1 \geq 3 \geq \Delta(C_n)+1$, all vertices of C_n can always be labeled. Therefore, Alice wins. □

Theorem 3.3. *Let m, n be positive integers where $n \geq 3$ and C_n be a cycle with n vertices. Then*

$$\chi_g(K_m \vee C_n)$$

$$= \begin{cases} m+2; & \text{if } n=4 \text{ and } m \text{ is odd}, \\ m+3; & \text{if } n=3 \text{ or } n=4 \text{ and } m \text{ is even or } n \geq 5 \text{ and } m \leq 3, \\ 2m-1; & \text{if } n \geq 5 \text{ and } m \geq 4 \text{ and } m \leq \left\lceil \frac{n}{2} \right\rceil + 1, \\ m + \left\lceil \frac{n}{2} \right\rceil; & \text{if } n \geq 5 \text{ and } m \geq 4 \text{ and } m \geq \left\lceil \frac{n}{2} \right\rceil + 2. \end{cases}$$

Proof. Let m, n be positive integers where $n \geq 3$ and C_n be a cycle with n vertices. Let $V(K_m) = \{u_1, u_2, ..., u_m\}$ and $V(C_n) = \{v_1, v_2, ..., v_n\}$.

Case 1. $n = 3$. We obtain that $\chi_g(K_m \vee V_3) = \chi_g(K_{m+3}) = m+3$.

Case 2. $n = 4$ and m is odd. Notice that $K_m \vee C_4$ has K_{m+2} as a subgraph. It requires at least $m+2$ colors to label all vertices of $K_m \vee C_4$. It remains to show that Alice has a winning strategy when there are $m+2$ colors.

Since m is odd, Alice can force Bob to first label a vertex of C_4, say v_1. Whenever Bob first labels v_1, Alice labels v_3 by using the same color. Moreover, when Bob first labels a vertex from $\{v_2, v_4\}$, Alice labels the remaining vertex by the same color. Hence, $m+2$ colors are enough.

Case 3. $n = 4$ and m is even. It requires to show that Bob wins when there are $m+2$ colors and Alice wins when there are $m+3$ colors.

Suppose that there are $m+2$ colors. Since m is even, Bob can force Alice to first label a vertex of C_4, say v_1. Whenever Alice first labels v_1, Bob labels v_3 by using a different color. Then v_2 needs the third colors. Hence, C_4 requires at least three colors and K_m requires another m color. Hence, $m+2$ colors are not enough. That is, Bob wins.

Suppose that there are $m+3$ colors. If Bob first labels a vertex from C_4, say v_1, then Alice labels v_3 by the same color. Hence, all vertices of $K_m \vee C_4$ can be labeled by $m+3$ colors. Otherwise, Bob will try to force Alice to first label a vertex from C_4 by keep labeling only vertices of K_n. Alice also keeps labeling vertices of K_m. Then their m first turns are only on vertices of K_m. Consequently, the remaining vertices are C_4 and each vertex has three available colors. Hence, the remaining vertices can always be labeled because of $\Delta(C_4) = 2$. That is, Alice wins.

Case 4. $n \geq 5$ and $m \leq 3$. On the first turn, if Alice labels a vertex from C_n, say v_1, then Bob labels v_3 by a different color. Hence, C_n requires three colors. If Alice does not label a vertex of C_n, then Bob labels v_1. On his next turn, he labels v_3 or v_{n-1} by a different color to confirm that C_n requires three colors. Therefore, $K_m \vee C_n$ requires $m+3$ colors.

It remains to show that Alice wins when there are $m+3$ colors. Suppose that there are $m+3$ colors. On the first turn, Alice labels a vertex from K_m by any color. Whatever Bob does, Alice keeps labeling a vertex of K_m by a new color until all vertices of K_m are labeled. If Bob tries to interfere in Alice's goal, then Bob will use colors as much as he can to label vertices of C_n. Since there are $m+3$ colors but $m \leq 3$, she can make all vertices of K_m be labeled by m colors. Hence, each vertex of C_n has three available colors. Then the remaining vertices can always be labeled. Therefore, Alice wins.

Case 5. $n \geq 5$ and $m \geq 4$ and $m \leq \left\lceil \frac{n}{2} \right\rceil + 1$. By Lemma 3.2, it remains to show that Bob wins when there are $2m-2$ colors. Suppose that there are only $2m-2$ colors. Bob tries to label $m-1$ vertices of C_n by $m-1$ colors in order to remain only $m-1$ colors for K_m. Notice that Alice cannot label all vertices of K_m before Bob reaches his goal. To interfere in Bob's goal, Alice must label a vertex of K_n on her first move and labels vertices of C_n by using already used colors on the remaining moves. Because $m-1 \leq \left\lceil \frac{n}{2} \right\rceil$, Bob can achieve his goal. Hence, Bob wins.

Case 6. $n \geq 5$ and $m \geq 4$ and $m \geq \left\lceil \frac{n}{2} \right\rceil + 2$. The proof is divided into two parts. The first part is that Bob wins where there are $m-1+\left\lceil \frac{n}{2} \right\rceil$ colors. The other part is that Alice wins where there are $m+\left\lceil \frac{n}{2} \right\rceil$ colors.

Suppose that there are only $m-1+\left\lceil \frac{n}{2} \right\rceil$ colors. On all Bob's first $\left\lceil \frac{n}{2} \right\rceil$ turns, he will label $\left\lceil \frac{n}{2} \right\rceil$ vertices of C_n by using $\left\lceil \frac{n}{2} \right\rceil$ different colors. He may finish the goal before $\left\lceil \frac{n}{2} \right\rceil$ turns if Alice helps him label C_n. If he achieves the goal, then C_n uses at least $\left\lceil \frac{n}{2} \right\rceil$ colors. Then $K_m \vee C_n$ requires at least $m+\left\lceil \frac{n}{2} \right\rceil$ colors. If he runs out of colors, then there exists a vertex of K_m which is not yet labeled because $m \geq \left\lceil \frac{n}{2} \right\rceil + 2$. There is no color for the vertex; hence, Bob wins.

Suppose that there are $m + \left\lceil \frac{n}{2} \right\rceil$ colors. Alice keeps labeling vertices of K_m until all vertices of K_m are labeled or Bob first labels a vertex of C_n. If all vertices of K_m are labeled, then all vertices of C_n can be labeled because of $\left\lceil \frac{n}{2} \right\rceil \geq 3$. If Bob first labels a vertex of C_n, then Alice starts labeling vertices of C_n. Alice's goal is that all vertices of C_n be labeled by at most $\left\lceil \frac{n}{2} \right\rceil$ colors. Since she can always reach the goal, there are m remaining colors for K_m. Hence, Alice wins. □

4. Future Work

In 2008, Bartnicki et al. [6] investigated the game chromatic number of the Cartesian product of two graphs. The Cartesian product $G \square H$ is the graph with vertex set $V(G) \times V(H)$, where two vertices (u_1, v_1) and (u_2, v_2) are adjacent if and only if either $u_1 = u_2$ and $v_1 v_2 \in E(H)$ or $v_1 = v_2$ and $v_1 u_2 \in E(G)$. In [6], the authors proved that $\chi_g(G \square H)$ is not bounded above in terms of $\chi_g(G)$ and $\chi_g(H)$. However, Zhu [1] found the upper bound of $\chi_g(G \square H)$ in terms of the game chromatic number of G and acyclic chromatic number of H. In [6], Bartnicki et al. also gave the exact values of $\chi_g(P_2 \square P_n)$, $\chi_g(P_2 \square C_n)$ and $\chi_g(P_2 \square K_n)$. Later, Sia [2] found the exact value of $\chi_g(S_m \square P_n)$, $\chi_g(S_m \square C_m)$, $\chi_g(P_2 \square W_n)$ and $\chi_g(P_2 \square K_{m,n})$. In [3], the author proved that $\chi_g(C_m \square C_n) \leq 5$ and $\chi_g(C_{2m} \square C_n) = 5$ for $m \geq 3$ and $n \geq 7$.

Recall that every generalized wheel graph can be obtained from the join of a cycle and a complete graph. The game chromatic numbers of some join graphs are not yet investigated. The following statements may become the next research paper:

(1) The game chromatic number of $C_m \vee C_n$, where C_m and C_n are cycles.

(2) The game chromatic number of $P_m \vee P_n$, where P_m and P_n are paths.

(3) The game chromatic number of $K_m \vee P_n$, where K_m is a complete graph and P_n is a path.

Here, we will investigate the game chromatic number of some join graphs. It is interesting that $\chi_g(C_4 \vee C_4) = 7$ while $\chi_g(C_4 \vee C_5) = 5$.

Remark 4.1. The game chromatic number of $C_4 \vee C_4$ is seven.

Proof. Notice that all vertices of the first C_4 can be labeled by at most four colors. There are at least three colors for the other cycle. Since the maximum degree of cycles is two, all vertices of cycles can be labeled by three colors. Hence, $\chi_g(C_4 \vee C_4) \leq 7$.

Suppose that there are six available colors. Let u_1, u_2, u_3, u_4 and v_1, v_2, v_3, v_4 be two closed chains of C_4. Without loss of generality, Alice labels u_1 by color 1. Then Bob labels u_3 by color 2. On the next move, Alice has two choices.

Case 1. Alice labels v_1 by color 3. Then Bob labels v_3 by color 4. On the next move, without loss of generality, suppose that Alice labels u_2 by color 5. Then Bob labels u_4 by color 6. Hence, there is no available color for v_2 and v_4.

Case 2. Alice labels u_2 by color 3. Then Bob labels u_4 by color 4. On the next move, without loss of generality, suppose that Alice labels v_1 by color 5. Then Bob labels v_2 by color 6. Hence, there is no available color for v_2 and v_4.

Therefore, Bob wins. □

Remark 4.2. The game chromatic number of $C_4 \vee C_5$ is five.

Proof. It is easy to see that $C_4 \vee C_5$ requires at least five colors because C_4 requires two colors and C_5 requires three colors. Suppose that there are five available colors. Let u_1, u_2, u_3, u_4 and v_1, v_2, v_3, v_4, v_5 be closed chains of C_4 and C_5.

Notice that Alice can force Bob to first label a vertex from u_1, u_2, u_3, u_4. Whenever Bob labels u_1, Alice labels u_3 by the same color and whenever Bob labels u_2, Alice labels u_4 by the same color.

Hence, we focus on C_5. Alice labels v_1 by color 1. When Bob labels a vertex from $\{v_2, v_4\}$ or $\{v_3, v_5\}$, Alice labels the other vertex by a new color. If Bob uses color 1, then Alice uses a new color; otherwise, Alice uses the same color. Hence, Alice wins. □

Acknowledgement

The author thanks the anonymous referees for their valuable suggestions which led to the improvement of the manuscript.

References

[1] X. Zhu, Game coloring the Cartesian product of graphs, J. Graph Theory 59 (2008), 261-278.

[2] C. Sia, The game chromatic number of some families of Cartesian product graphs, AKCE Int. J. Graphs Comb. 6(2) (2009), 315-327.

[3] A. Raspaud and J. Wu, Game chromatic number of toroidal grids, Inform. Process. Lett. 109 (2009), 1183-1186.

[4] H. L. Bodlaender, On the complexity of some coloring games, Internat. J. Found. Comput. Sci. 2 (1991), 133-147.

[5] M. Gardner, Mathematical games, Scientific American, 1981, 23 pp.

[6] T. Bartnicki, B. Brear, J. Grytczuk, M. Kove, Z. Miechowicz and I. Peterin, Game chromatic number of Cartesian product graphs, Electron. J. Combin. 15 (2008), #R72, 13 pp.

[7] U. Faigle, U. Kern, H. A. Kierstead and W. T. Trotter, On the game chromatic number of some classes of graphs, Ars Combin. 35 (1993), 143-150.

[8] D. Guan and X. Zhu, The game chromatic number of outerplanar graphs, J. Graph Theory 30 (1999), 67-70.

[9] H. A. Kierstead and W. T. Trotter, Planar graph colouring with an uncooperative partner, J. Graph Theory 18(6) (1994), 569-584.

[10] X. Zhu, Refined activation strategy for the marking game, J. Combin. Theory Ser. B 98 (2008), 1-18.

[11] J. Wu and X. Zhu, Lower bounds for the game colouring number of partial *k*-trees and planar graphs, Discrete Math. 308 (2008), 2637-2642.

[12] X. Zhu, The game coloring number of pseudo partial *k*-trees, Discrete Math. 215 (2000), 245-262.