Finding minimal Ferrers-esque graphs on path graphs and cycle graphs via set cover

Selcuk Topal
Department of Mathematics, Bitlis Eren University, Bitlis, Turkey

Received: 2 June 2016, Accepted: 4 August 2016
Published online: 9 October 2016

Abstract: This paper presents minimal construction techniques of a new graph class called Ferrer-esque [10] comes from Ferrers relation [9] on path and cycle graphs by using set cover method. The minimal constructions provide to obtain a Ferrer-esque graph by adding minimum number of edges to paths and cycles. We also state some open problems about Ferrer-Esque graphs to the readers.

Keywords: Graph algorithms, factorization, matching, partitioning, covering and packing, Paths and cycles.

1 Introduction and preliminaries

The notion Ferrers relation is introduced by Riguet [9]. This relation has been used for different purposes in a variety of science fields such as in Formal Concept Analysis [5], in partitions presented as Ferrers diagrams [1] and also in social choice theory [7]. Ehrenborg and van Willigenburg introduced a well-known graph class (Ferrers graph) by using Ferrers diagrams [4], not by Ferrers relation directly [4]. Topal introduced a new graph class, called Ferrers-esque graphs, by using original definition of the relation [10] and precisely different from defined by Ferrers diagrams.

In this paper, we construct minimal Ferrers-esque graphs on path graphs and cycle graphs by using minimum set cover. Path graphs with four nodes and cycle graphs with four and also five nodes are Ferrers-esque graphs, naturally. Both path graphs with nodes greater than four and cycle graphs with nodes greater than five are not Ferrers-esque graphs. We abbreviate Ferrers-esque graphs to Ferrer graphs for readability.

By a simple graph $G = (V, E)$, we will mean an undirected graph without loops or multiple edges. An edge between $u$ and $v$ is denoted by $e = uv$ or $e = \{u, v\}$ interchangeably. A simple graph $G = (V, E)$ is called a path graph if it can be drawn so that all of its vertices and edges lie on a single straight line. We abbreviate a path graph $G = (V, E)$ with $n$ nodes to $P_n$. A simple graph $G = (V, E)$ is called a cycle graph, sometimes simply known as an $n$−cycle, is a graph on $n$ nodes containing a single cycle through all vertices. We abbreviate a cycle graph $G = (V, E)$ with $n$ nodes to $C_n$. A complete graph is a simple graph in which each pair of its vertices is connected by an edge. A complete graph $G = (V, E)$ with $n$ nodes which is shown by $K_n$.

Definition 1.[9] If a relation $R$ over a set $A$ is a Ferrers relation, it holds if $aRb$ and $cRd$ then either $aRd$ or $bRc$ for all distinct elements $a, b, c, d \in A$.

Definition 2.[10] A simple graph $G = (V, E)$ is a Ferrer graph if for all distinct $x, y, z, w \in V$, $xy \in E$ and $zw \in E$ then either $xw \in E$ or $yz \in E$. The definition of Ferrer graph must be extended to “ if $xy \in E$ and $zw \in E$ then either $xw \in E$ or $yz \in E$ or $yw \in E$ or $xz \in E$ ” since $xy \in E \Leftrightarrow yx \in E$ holds for all simple graphs.
Definition 3. Given a universe $\mathcal{U}$ and a family $\mathcal{S}$ of subsets of $\mathcal{U}$, a cover is a subfamily $\mathcal{C} \subseteq \mathcal{S}$ of sets whose union is $\mathcal{U}$. In the set covering decision problem, the input is a pair $(\mathcal{U}, \mathcal{S})$ and an integer $k$; the question is whether there is a set covering of size $k$ or less.

Remark. We will use minimum and unweighted version of set cover method because minimal Ferrer graphs require adding minimum number of edges on $P_n$ and $C_n$ and no need to consider weighted edges in this work.

2 Constructions of Minimal Ferrer Graphs on $P_n$

$P_4$ is Ferrer graph with the smallest number of elements because we need at least four distinct vertices to create a Ferrer graph by Definition 1.2. $P_4$ is called primitive because every path (line) graph with four elements is a Ferrer graph.

Now, we construct Ferrer graphs on $P_5 = (V, E)$ such that $V = \{x, y, z, t, w\}$ and $E = \{(x, y), (y, z), (z, t), (t, w)\}$ in order to give an explicative example. We look for at least one edge that holds for all distinct $x, y, t, w \in V$ then either $xw \in E$ or $yt \in E$ or $yw \in E$ or $xt \in E$ by Definition 1.4.

$P_5$ has two induced subgraphs of with four vertices (see Figure 3). It is needless to add extra edges since the subgraphs are already primitive. Let’s consider edges $xy$ and $tw$. Then, either $xw$ or $yt$ or $yw$ or $xt$ is in $E$. 
A minimal Ferrer graph on $P_5$ means that a Ferrer graph formed by adding minimum number of edges to $P_5$. We could add more edges to the graphs in Figure 4 until they are complete graph $K_5$. But, our intention is to obtain such as the graphs in Figure 4.

**Lemma 1.** Every complete graph $K_n$ where $n \geq 4$ is a Ferrer graph.

**Proof.** It is clear that for each $x_i, x_j$ the edge $x_i x_j$ in $E(G)$ hence $K_n$ is Ferrer graph for $n \geq 4$.

### 3 Constructions of minimal Ferrer graphs on $C_n$

In this section, we will construct minimal Ferrer graphs on $C_n$. Both $C_4$ and $C_5$ are primitives, $C_n$ is not primitive where $n > 5$.

![Fig. 4: Minimal Ferrer graphs on $P_5$.](image)

![Fig. 5: $C_4$ and $C_5$.](image)
Let’s take \( C_6 \) in order to construct minimal Ferrer graphs on it. Again, we look for at least one edge that holds for all distinct \( x, y, z, w \in V_{C_6} \), then either \( xw \in E_{C_6} \) or \( yz \in E_{C_6} \) or \( yw \in E_{C_6} \) or \( xz \in E_{C_6} \) by Definition 1.4.

![Fig. 6: Cycle graph \( C_6 \).](image)

When we apply Definition 1.3 for \( C_6 \), we can obtain two of Ferrer graphs in Figure 7 in addition to the complete graph \( K_6 \). We want to have Ferrer graphs on \( C_6 \) by adding minimum edge of numbers on \( C_6 \).

![Fig. 7: Two Ferrer graphs on \( C_6 \).](image)

Both (a) and (b) are Ferrer graphs. We will focus on (a) in order to construct minimal Ferrer graphs on \( C_6 \).

### 4 Algorithms

**Explanations for algorithms** Most of notations in Algorithms we give in this paper comes from Python programming language [8]. Rest of the notations such as \([[]]\) for sets will use for readability.

1.\( > \): comment for statements to explain operations or to give an example  
2.\( <- \): value assignment  
3.Data structures:  
   (i)\([[]]\): sets: \([1] = \{12', 35'\}\): a set 1 with elements '12' and '35'  
   \([A] = \{1', 2'\}\): a set \( A \) with elements '1' and '3'; \([1][1] \): first element of the set \([A][1] \) is '1'. \([A][\] \): The number of elements of the set \([A] \)  
   (ii)'s': converting the value \( s \) to a string \( s \)  
   \( i <- 0, j <- 7 \)  
   'i' means that '0'. \([0']\): \((i + 1)(j + 2)'\) means that the string '19'. \((i + 1)'\) means that the string '17'.  
   (iii)\( A = \{a': b\}\): a dictionary \( A \), key of \( A \) is string 'a' and its value is string 'b'. \( A = dict() \) means that \( A \) is assigned to empty dictionary. Let’s consider \( Universe = \{12' : \{1', 2'\}, 36' : \{4', 8'\}\} \). \( Universe[12] \) is \{1', 2'\}.
Algorithm 1 Constructions of minimal Ferrer graphs on $P_n$

1: procedure CONSTRUCTION OF MINIMAL FERRER GRAPHS on $P_n$
2: $s ← 0, i ← 1, k ← 1, f ← 1, h ← 0, m ← 0, j ← 4, |FullEgdeSet| ← 0, Universe = dict()
3: Input a number $n (n > 4)$
4: while $i < n - 3$ do
5: \[ m ← i + 3 \]
6: while $m < n$ do
7: \[ s ← s + 1 \]
8: \[ |s| ← 0 \]
9: \[ |s| ← |s| \cup \{\{i\}\} \]
10: \[ |s| ← |s| \cup \{\{i + 1\}\} \]
11: \[ |s| ← |s| \cup \{\{i + 1\} \cup \{i + 1\}\} \]
12: \[ m ← m + 1 \]
13: \[ j ← j + 1 \]
14: end while
15: \[ i ← i + 1 \]
16: end while
17: while $h ≤ l$ do
18: \[ h ← h + 1 \]
19: \[ |FullEgdeSet| ← |FullEgdeSet| \cup |h| \]
20: end while
21: \[ |FullEgdeSet| \] is cardinality of the set $|FullEgdeSet|$.
22: while $f ≤ l$ do
23: \[ f ← f + 1 \]
24: \[ if |FullEgdeSet| \cup |j| \]
25: \[ |FullEgdeSet| \cup |j| \] is cardinality of the set $|FullEgdeSet|$.\]
26: \[ |FullEgdeSet| \cup |j| \] is cardinality of the set $|FullEgdeSet|$.\]
27: \[ Universe[j] = value \]
28: \[ Universe[j] = value \]
29: \[ k ← k + 1 \]
30: \[ value ← |FullEgdeSet| \cup |j| \]
31: \[ value ← |FullEgdeSet| \cup |j| \]
32: \[ k ← k + 1 \]
33: end while
34: \[ f ← f + 1 \]
35: end while
36: Apply set cover algorithm for keys and their values of Universe to cover $|FullEgdeSet|$.\]
37: end procedure

Example 1. We give an explanation of running of Algorithm 1 for $P_6$.


(ii) End of the step 21 in the algorithm, there occur the set $|FullEgdeSet| = \{14', 15', 24', 25', 16', 26', 35', 36'\}$

(iii) End of the step 35 in the algorithm, there occur $[14'] = \{1'\}$, $[15'] = \{1', 2'\}$, $[16'] = \{2'\}$, $[24'] = \{1'\}$, $[25'] = \{1', 2', 3'\}$, $[26'] = \{2', 3'\}$, $[35'] = \{3'\}$, $[36'] = \{3'\}$

\[ Universe = \{1'4': \{1'\}, 15': \{1', 2'\}, 16': \{2'\}, 24': \{1'\}, 25': \{1', 2', 3'\}, 26': \{2', 3'\}, 35': \{3'\}, 36': \{3'\} \}$

(iv) In the step 36 of the algorithm, the goal of set cover method is to select minimum number of subsets (values of the dictionary Universe). Here, set cover method will select the key '25' which covers $\{1', 2', 3'\}$. This means that it is sufficient to add the edge (2,5) to generate a minimal Ferrer graph from $P_6$ (see (b) in Figure 8). Neither value of the key '24' nor of '26' does not cover set of $\{1', 2', 3'\}$. Even though Union of values of keys '24' nor of '26' covers, we prefer to have minimum number of edges.
Algorithm 2 Constructing minimal Ferrers graphs on \(C_n\)

1: procedure CONSTRUCTION OF MINIMAL FERRERS GRAPHS ON \(C_n\)
2: \[ x \leftarrow 0, \ i \leftarrow 1, \ t \leftarrow 2, \ h \leftarrow 1, \ z \leftarrow 3, \ j \leftarrow 4, \ |[FullEdgeSet]| < 0, \ Universe = dict() \]
3: while \(i < n - 3\) do
4: \[ m \leftarrow i + 3 \]
5: \[ \text{while} \ m < n \ do \]
6: \[ x \leftarrow x + 1 \]
7: \[ z \leftarrow z + 1 \]
8: \[ m \leftarrow m + 1 \]
9: \[ \text{end while} \]
10: \[ i \leftarrow i + 1 \]
11: \[ t \leftarrow t + 1 \]
12: \[ \text{end while} \]
13: \[ m \leftarrow m + 1 \]
14: \[ \text{while} \ (n - z > 2) and (z > 2) \ do \]
15: \[ x \leftarrow x + 1 \]
16: \[ z \leftarrow z + 1 \]
17: \[ \text{end while} \]
18: \[ m \leftarrow m + 1 \]
19: \[ \text{while} \ 2 \leq t \leq n - 4 \ do \]
20: \[ x \leftarrow x + 1 \]
21: \[ z \leftarrow z + 1 \]
22: \[ \text{end while} \]
23: \[ h \leftarrow h + 1 \]
24: \[ i \leftarrow i + 1 \]
25: \[ t \leftarrow t + 1 \]
26: \[ |[FullEdgeSet]| = \text{cardinality of the set } |[FullEdgeSet]| \]
27: \[ \text{end while} \]
28: \[ h \leftarrow h + 1 \]
29: \[ i \leftarrow i + 1 \]
30: \[ \text{end while} \]
31: \[ t \leftarrow t + 1 \]
32: \[ h \leftarrow h + 1 \]
33: \[ i \leftarrow i + 1 \]
34: \[ |[FullEdgeSet]| = |\{k\}| \]
35: \[ |[FullEdgeSet]| = \text{cardinality of the set } |[FullEdgeSet]| \]
36: \[ h \leftarrow h + 1 \]
37: \[ t \leftarrow t + 1 \]
38: \[ h \leftarrow h + 1 \]
39: \[ i \leftarrow i + 1 \]
40: \[ |[FullEdgeSet]| = |\{k\}| \]
41: \[ \text{end while} \]
42: \[ i \leftarrow i + 1 \]
43: \[ \text{while} \ f < l \ do \]
44: \[ \text{while} \ k \leq x \ do \]
45: \[ \text{if } |[FullEdgeSet]| \subset \{k\} \text{ then} \]
46: \[ |[FullEdgeSet]| \text{ or } \{k\} \]
47: \[ \text{end if} \]
48: \[ \text{end while} \]
49: \[ \text{end while} \]
50: \[ \text{end while} \]
51: \[ \text{Apply set cover algorithm for keys and their values of } Universe \text{ to cover } |[FullEdgeSet]|. \]
52: \[ \text{end procedure} \]
Example 2. We give an explanation of running of Algorithm 2 for $C_7$.

-End of the step 37 in Algorithm 2, we have $[[1]] = \{14',15',24',25'\}$, $[[2]] = \{15',16',25',26'\}$, $[[3]] = \{25',26',35',36'\}$, $[[4]] = \{26',27',36',37'\}$.

$[[5]] = \{36',37',46',47'\}$, $[[6]] = \{13',14',37',47'\}$, $[[7]] = \{14',15',47',57'\}$.


$[14']] = \{1',6',7\}, [[15']] = \{1',2',7\}, [[16']] = \{2\}$, $[[24']] = \{1'\}$, $[[25']] = \{1',2',3\}$, $[[26']] = \{2',3',4\}$, $[[27']] = \{4'\}$, $[[35']] = \{3'\}$, $[[36']] = \{3',4',5\}$, $[[37']] = \{4',5',6\}$, $[[46']] = \{5'\}$, $[[47']] = \{5',6',7\}$, $[[57']] = \{7'\}$.

$Universe = \{14':\{1',6',7\}, 15':\{1',2',7\}, 16':\{2\}, 24':\{1\}, 25':\{1',2',3\}, 26':\{2',3',4\}, 27':\{4\}, 35':\{3\}, 36':\{3',4',5\}, 37':\{4',5',6\}, 46':\{5'\}, 47':\{5',6',7\}, 57':\{7'\}$.

-In the step 56 of the algorithm, the set cover method may select three keys '25', '37' and '47' or else '14', '15' and '36' so that values of them cover $\{1',2',3',4',5',6',7'\}$. Of course, we have other possibilities to cover set $\{1',2',3',4',5',6',7'\}$. It is sufficient to select to add the edges (2,5), (3,7), (4,7) or (1,4), (1,5), (3,6) to generate a minimal Ferrer graph from $C_7$ (see (c) and (d) in Figure 9). Note that even though the graph in (b) of Figure 9 has more edges than the graphs in (c) and (d) have, it is not a Ferrer graph.

Fig. 9: An illustration of Algorithm 2 for $C_7$. 

(a) $C_7$ (not a Ferrer graph) 
(b) Not a Ferrer graph 
(c) A Ferrer graph on $C_7$ (minimal) 
(d) A Ferrer graph on $C_7$ (minimal)
5 Conclusion

In this paper, we have given algorithms for minimal Ferrer graph constructions on $P_n$ and $C_n$. Our techniques we have used for the constructions include forming sets by edges of graphs and then applying set covering problem to the sets.

6 Open problems

Minimal Ferrer graph constructions should be extended on Tree-like graphs or other graphs. More efficient algorithms for the constructions should be investigated because decision version of set covering is in NP-complete and the optimization version of set cover is in NP-hard [6]. Finally, Graph products of two minimal Ferrer graphs should be surveyed. General combinatorial formulations of $M_n(P_n)$ and $M_n(C_n)$, and also $M(P_n)$ and $M(C_n)$ should be given ($M(G)$ is the number of minimum edges which make $G$ being a Ferrer graph and $M^n(G)$ is the number of minimal Ferrer graphs which can be constructed on $G$). Finally, every Ferrers, particularly minimal Ferrers graph, is a $2K_2$ graph.

7 Thanks

We would like to thank Canan Çiftci, Furkan Tekelioğlu and Zafer Cömert who have discussed this topic with us.

References