Communication in Mathematical Modeling and Applications
http://ntmsci.com/cmma

Finding minimal Ferrers-esque graphs on path graphs ans 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 preliminares

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.

* Corresponding author e-mail: s.topal@beu.edu.tr

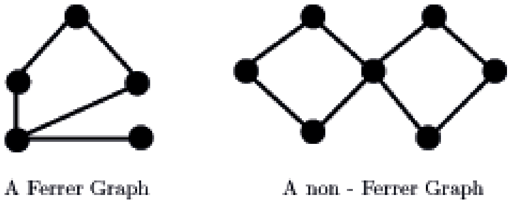


Fig. 1: A Ferrer and a non-Ferrer graph.

Definition 3.[2] Given a universe $\mathscr{U}$ and a family $\mathscr{S}$ of subsets of $\mathscr{U}$, a *cover* is a subfamily $\mathscr{C} \subseteq \mathscr{S}$ of sets whose union is $\mathscr{U}$. In the set covering *decision problem*, the input is a pair $(\mathscr{U}, \mathscr{S})$ 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.

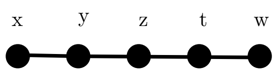


Fig. 2: Path graph P_5.

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.

Fig. 3: Two induced subgraphs of P_5.

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.

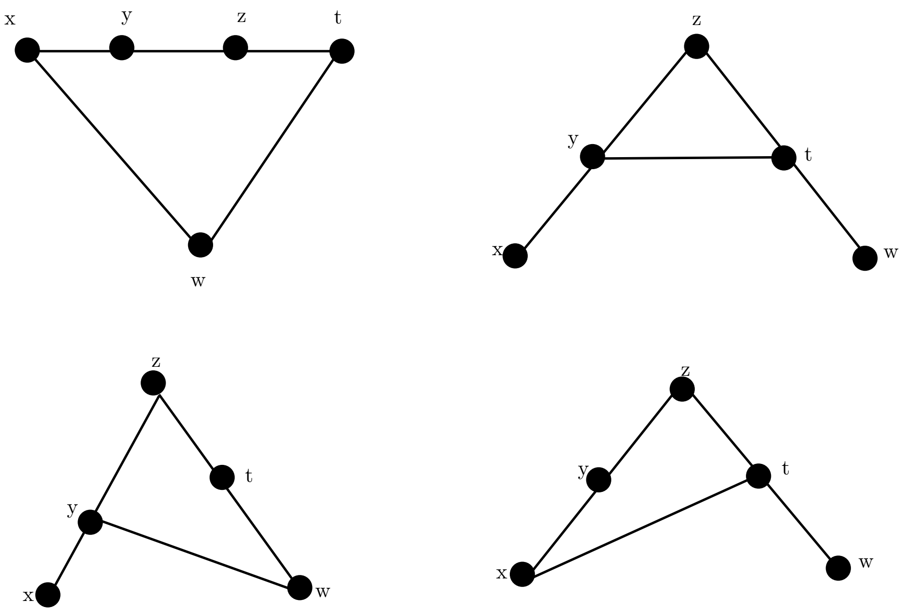


Fig. 4: Minimal Ferrer graphs on P_5.

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 $n > 5$.

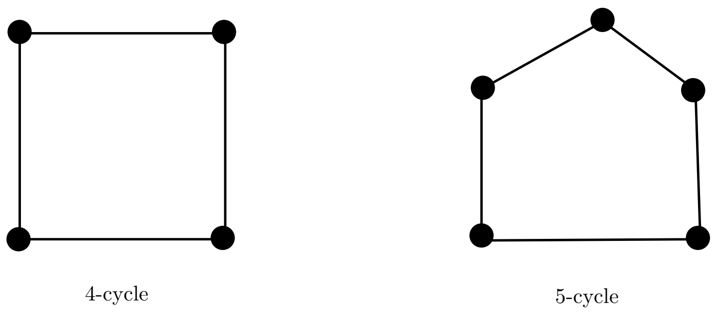


Fig. 5: C_4 and C_5.

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.

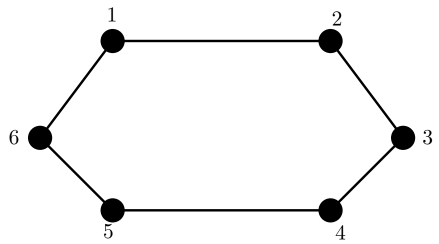


Fig. 6: Cycle graph C_6.

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.

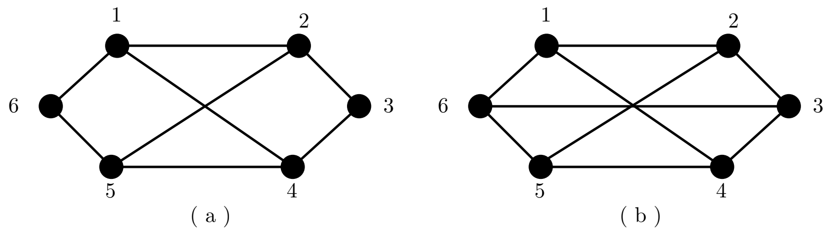


Fig. 7: Two Ferrer graphs on C_6.

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.$\triangleright$: comment for statements to explain operataions or to give an example

2.$\longleftarrow$: value assigment

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

$i \longleftarrow 0$, $j \longleftarrow 7$

$'i'$ means that $'0'$. $[['i']]$: set $[['0']]$ $'(i+1)(j+2)'$ means that the string $'19'$. $'(i+1)j'$ 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 dict $Universe = \{'12' : \{'1','2'\},'36' : \{'4','8'\}\}$. $Universe['12']$ is $\{'1','2'\}$ and

$Universe['36']$ is $\{'4','8'\}$. We can do union of $Universe['12']$ and $Universe['36']$, that returns to $\{'1','2','4','8'\}$. On the other hand, we can initialize a key and its value to a dictionary. For example, if we do $Universe['47'] = \{'3','2'\}$ and then dictionary $Universe = \{'12' : \{'1','2'\},'36' : \{'4','8'\},'47' : \{'3','2'\}\}$.

Algorithm 1 Constructions of minimal Ferrer graphs on P_n

1: **procedure** CONSTRUCTION OF MINIMAL FERRER GRAPHS ON P_n
2: $s \leftarrow 0$, $i \leftarrow 1$, $k \leftarrow 1$, $f \leftarrow 1$, $h \leftarrow 1$, $l \leftarrow 0$, $m \leftarrow 0$, $j \leftarrow 4$, $[[FullEgdeSet]] \leftarrow \emptyset$, $Universe = dict()$
3: Input a number n $(n > 4)$
4: **while** $i < n-3$ **do**
5: $m \leftarrow i+3$
6: **while** $m < n$ **do**
7: $s \leftarrow s+1$
8: $[[s]] \leftarrow \emptyset$
9: $[[s]] \leftarrow [[s]] \cup \{'ij'\}$ ▷ $[[1]] \leftarrow' 14'$, adding edge $e = (1,4)$ to set $[[1]]$
10: $[[s]] \leftarrow [[s]] \cup \{'i(j+1)'\}$ ▷ $[[1]] \leftarrow' 15'$, adding edge $e = (1,5)$ to set $[[1]]$
11: $[[s]] \leftarrow [[s]] \cup \{'(i+1)j'\}$ ▷ $[[1]] \leftarrow' 24'$, adding edge $e = (2,4)$ to set $[[1]]$
12: $[[s]] \leftarrow [[s]] \cup \{'(i+1)(j+1)'\}$ ▷ $[[1]] \leftarrow' 25'$, adding edge $e = (2,5)$ to set $[[1]]$
13: $m \leftarrow m+1$
14: $j \leftarrow j+1$
15: **end while**
16: $i \leftarrow i+1$
17: **end while**
18: **while** $h \leq s$ **do**
19: $[[FullEgdeSet]] \leftarrow [[FullEgdeSet]] \cup [[h]]$
20: $h \leftarrow h+1$
21: **end while**
22: $l \leftarrow |[[FullEgdeSet]]|$ ▷ $|[[FullEgdeSet]]|$ is cardinality of the set $[[FullEgdeSet]]$
23: **while** $f \leq l$ **do**
24: **while** $k \leq s$ **do**
25: **if** $[[FullEgdeSet]][f] \in [[k]]$ **then**
26: $[['[[FullEgdeSet]][f]']] \leftarrow \emptyset$ ▷ $[['12']] = \emptyset$
27: $[['[[FullEgdeSet]][f]']] \leftarrow [['[[FullEgdeSet]][f]']] \cup 'k'$ ▷ $[['14']] = \{'1'\}$
28: **end if**
29: $key \leftarrow' [[FullEgdeSet]][f]'$ ▷ $key \leftarrow' 14'$
30: $value \leftarrow [['[[FullEgdeSet]][f]']]$ ▷ $value \leftarrow' 1'$
31: $Universe[key] = value$ ▷ $Universe = \{'14' :' 1'\}$
32: $k \leftarrow k+1$
33: **end while**
34: $f \leftarrow f+1$
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.

(i) End of the step 18 in Algorithm 1, there occur $[[1]] = \{'14','15','24','25'\}$, $[[2]] = \{'15','16','25','26'\}$, $[[3]] = \{'25','26','35','36'\}$

(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 $[['14']] = \{'1'\}$, $[['15']] = \{'1','2'\}$, $[['16']] = \{'2'\}$, $[['24']] = \{'1'\}$, $[['25']] = \{'1','2','3'\}$, $[['26']] = \{'2','3'\}$, $[['35']] = \{'3'\}$, $[['36']] = \{'3'\}$
$Universe = \{'14' : \{'1'\},'15' : \{'1','2'\},'16' : \{'2'\},'24' : \{'1'\},$
$'25' : \{'1','2','3'\},'26' : \{'2','3'\},'35' : \{'3'\},'36' : \{'3'\}\}$

(iv) In the step 36 of the algorithm, the goal of set cover method is to select mininum 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 sufficent 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 value of keys $'24'$ nor of $'26'$ covers, we prefer to have minimum number of edges.

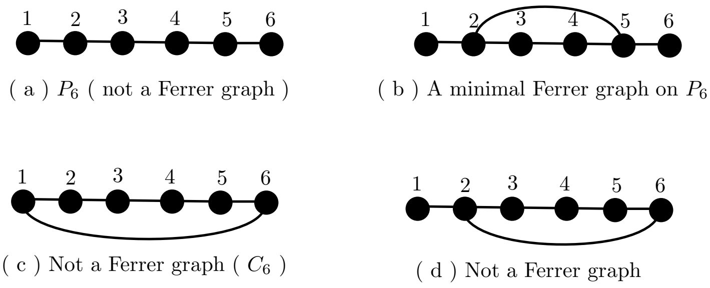
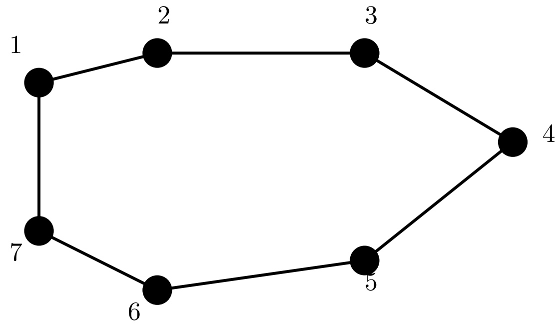


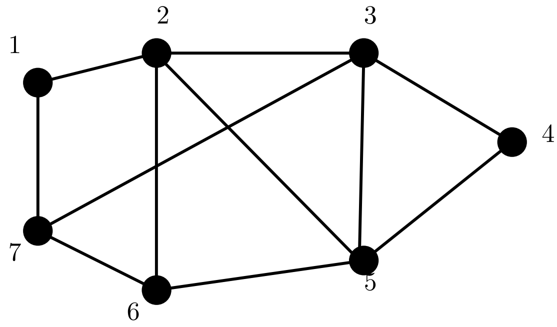
Fig. 8: An illustration of Algorithm 1 for P_6.

Algorithm 2 Constructing minimal Ferrer graphs on C_n

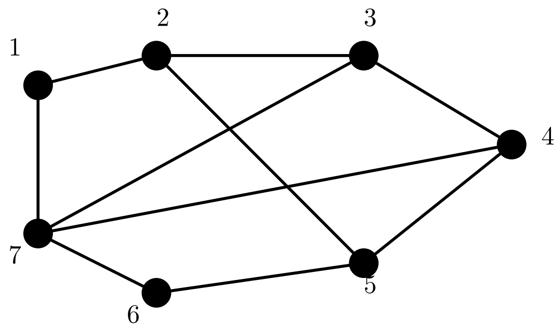
```
procedure CONSTRUCTION OF MINIMAL FERRER GRAPHS ON C_n
    Input a number n (n > 5)
    s ← 0, i ← 1, k ← 1, f ← 1, t ← 2, h ← 1, , z ← 3, j ← 4, [[FullEgdeSet]] ← ∅, Universe = dict()
    while i < n − 3 do
        m ← i + 3
        while m < n do
            s ← s + 1
            [[s]] ← ∅                                          ▷ [[1]] = ∅
            [[s]] ← [[s]] ∪ {'ij'}                             ▷ '14'
             [[s]] ← [[s]] ∪ {'i(j+1)'}                        ▷ '15'
             [[s]] ← [[s]] ∪ {'(i+1)(j)'}                      ▷ '24'
             [[s]] ← [[s]] ∪ {'(i+1)(j+1)'}                    ▷ '25'
             m ← m + 1
             j ← j + 1
         end while
         i ← i + 1
     end while
     s ← s + 1
     while (n − z > 2) and (z > 2) do
         [[s]] ← ∅
         [[s]] ← [[s]] ∪ {'1z'}                                ▷ '13'
         [[s]] ← [[s]] ∪ {'1(z+1)'}                            ▷ '14'
         [[s]] ← [[s]] ∪ {'zn'}                                ▷ '36'
         [[s]] ← [[s]] ∪ {'(z+1)n'}                            ▷ '46'
         s ← s + 1
         z ← z + 1
     end while
     s ← s + 1
     while 2 ≤ t ≤ n − 4 do
         [[s]] ← ∅
         [[s]] ← [[s]] ∪ {'t(n−1)'}                            ▷ '25'
         [[s]] ← [[s]] ∪ {'tn'}                                ▷ '26'
         [[s]] ← [[s]] ∪ {'(t+1)(n−1)'}                        ▷ '36'
         [[s]] ← [[s]] ∪ {'(t+1)n'}                            ▷ '35'
         s ← s + 1
         t ← t + 1
     end while
     while h ≤ s do
         [[FullEgdeSet]] ← [[FullEgdeSet]] ∪ [[s]]
         h ← h + 1
     end while
     l ← |[[FullEgdeSet]]|        ▷ |[[FullEgdeSet]]| is cardinality of the set [[FullEgdeSet]]
     while f ≤ l do
         while k ≤ s do
             if [[FullEgdeSet]][f] ∈ [[k]] then
                 [['[[FullEgdeSet]][f]']] ← ∅                  ▷ [['14']] = ∅
                 [['[[FullEgdeSet]][f]']] ← [['[[FullEgdeSet]][f]']] ∪ 'k'   ▷ [['14']] = {'1'}
             end if
             key ←' [[FullEgdeSet]][f]'                        ▷ key ←' 14'
             value ← [['[[FullEgdeSet]][f]']]                  ▷ value ←' 1'
             Universe[key] = value                             ▷ Universe = {'14' :' 1'}
             k ← k + 1
         end while
         f ← f + 1
     end while
     Apply set cover algorithm for keys and their values of Universe to cover [[FullEgdeSet]].
 end procedure
```



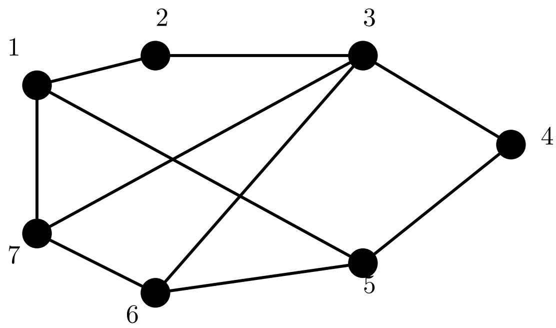
(a) C_7 (not a Ferrer graph)



(b) Not a Ferrer graph



(c) A Ferrer graph on C_7 (minimal)



(d) A Ferrer graph on C_7 (minimal)

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

Example 2. We give an explanation of running of Algorithm 2 for C_7.

–End of the step 37 in Algorithm 2, we have $[[1]] = \{'14','15','24','25'\}$, $[[2]] = \{'15','16','25','26'\}$, $[[3]] = \{'25','26','35','36'\}$, $[[4]] = \{'26','27','36','37'\}$,
$[[5]] = \{'36','37','46','47'\}$, $[[6]] = \{'13','14','37','47'\}$, $[[7]] = \{'14','15','47','57'\}$

–End of the step 41 in the algorithm , there occur the set $[[FullEgdeSet]] = \{'14','15','16','24','25','26','27','35','36','37','46','47','57'\}$

–End of the step 55 in the algorithm , there occur $[['14']] = \{'1','6','7'\}$, $[['15']] = \{'1','2','7'\}$, $[['16']] = \{'2'\}$, $[['24']] = \{'1'\}$,
$[['25']] = \{'1','2','3'\}$, $[['26']] = \{'2','3','4'\}$,$[['27']] = \{'4'\}$, $[['35']] = \{'3'\}$, $[['36']] = \{'3','4','5'\}$, $[['37']] = \{'4','5','6'\}$, $[['46']] = \{'5'\}$, $[['47']] = \{'5','6','7'\}$, $[['57']] = \{'7'\}$.
$Universe = \{'14' : \{'1','6','7'\},'15' : \{'1','2','7'\},'16' : \{'2'\},'24' : \{'1'\},$
$'25' : \{'1','2','3'\},'26' : \{'2','3','4'\},'27' : \{4\},'35' : \{'3'\},'36' : \{'3','4',5,\},'37' : \{'4','5','6'\},'46' : \{'5'\},'47' : \{'5','6','7'\}\},'57' : \{'7'\}$

–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.

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 $\mathscr{M}^{\setminus}(P_n)$ and $\mathscr{M}^{n}(C_n)$, and also $\mathscr{M}(P_n)$ and $\mathscr{M}(C_n)$ should be given ($\mathscr{M}(G)$ is the number of minimum edges which make G being a Ferrer graph and $\mathscr{M}^{n}(G)$ 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 Çiftçi, Furkan Tekelioğlu and Zafer Cömert who have discussed this topic with us.

References

[1] Andrews, G. E., The Theory of Partitions, Cambridge, England: Cambridge University Press, pp. 6-7, 1998.

[2] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., Stein, C., Introduction to Algorithms, Cambridge, Mass. MIT Press and McGraw-Hill, pp. 1033-1038, 2008.

[3] David S. Johnson, Approximation algorithms for combinatorial problems, Journal of Computer and System Sciences, 9:256-278, 1974.

[4] Ehrenborg R. and van Willigenburg S., Enumerative properties of Ferrers graphs, Discrete Computational Geometry, pp. 481-492, 2007.

[5] Ganter, B. and Willer R., Formal Concept Analysis, Springer, 1999.

[6] Korte, B. and Vygen, J., Combinatorial Optimization: Theory and Algorithms (5 ed.), 2012.

[7] Öztürk, M., Ordered sets with interval representation and (m, n)-Ferrers relation, Annals of Operations Research, 163.1: 177-196, 2008.

[8] Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org

[9] Riguet J., Les relations de Ferrers, Comptes Rendus des Seances hebdomadaires del Academie des Sciences, Paris 232, 1951.

[10] Topal, S., A New Graph Class Defined by Ferrers Relation, New Trends in Mathematical Sciences, Volume: 3, Issue: 3, (September), pp:181-183, 2015.