THE MAXIMUM ACYCLIC MATCHING PROBLEM

by

Cemre Çelebi

B.S., Chemical Engineering, Boğaziçi University, 2019

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University
2022

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor Prof. Tınaz Ekim Aşıcı for her comments and encouraging support throughout the process of this master thesis. Also, I anı very grateful to Prof. Z. Caner Taşkın and Prof. Didem Gözüpek for taking part in my thesis committee and for their valuable suggestions.

I should not skip thanking my dearest friends Cihan, Şiar, Taha, and Özge for their unconditional support.

Last but not least, I owe my deepest gratitude to my parents Zübeyda and Haluk, and my sister Cansın. We went through too many things during this process and I feel very lucky to have you. I would not be me without you. Thanks a lot!

ABSTRACT

THE MAXIMUM ACYCLIC MATCHING PROBLEM

The aim of this thesis is to develop exact and heuristic methods to solve the Maximum Acyclic Matching problem, which deals with obtaining maximum matching such that the subgraph induced by saturated vertices is acyclic. The maximum matching problem tries to find the most extensive possible matching set. it is a well-studied problem that can be solved with combinatorial algorithms. However, for the maximum acyclic matching problem, we are searching for not only a maximum size matching but also we require that the subgraph induced by saturated vertices does not contain any cycles. For this purpose, an additional acyclicity constraint is needed. Even though some exact and approximate algorithms are established for particular graph classes, there is no such algorithm applicable for general graphs to find a maximum acyclic matching. In this study, four algorithms are suggested. Randomly generated graphs with different density levels and sizes are used to analyze their performance. Two of the algorithms are exact algorithms, which are extensive and cutting plane formulations. Based on experimental results, the cutting plane approach performs better since it works also with larger graphs. The other two algorithms are heuristics, which are modification and construction approaches. It is observed that the construction approach performs better than the modification approach in terms of both time efficiency and quality. The construction approach yields feasible and close-to-optimal results in a shorter period. When the results of the best exact and heuristics are compared, the cutting plane algorithm performs better based on optimality. However, it is not applicable for large graphs compared to the construction algorithm. Additionally, it is seen that the effect of acyclicity constraint is increasing while graph size and density are getting larger.

ÖZET

AZAMİ DÖNGÜSÜZ EŞLEŞTİRME PROBLEMİ

Bu tezin amacı, doymuş düğümlerin herhangi bir döngü içermeyen indüklenmiş alt çizgelerinin olduğu Azami Eşleştirme anlamına gelen Azami Döngüsel Eşleştirmeyi bulmaktır. Azami Eşleştirme Problemi, mümkün olan en büyük eşleştirme kümesini bulmaya çalışır. Kombinatoryal algoritma ile çözülebilen literatürde sıklıkla rastlanan bir problemdir. Bununla birlikte, azami döngüsüz eşleştirme problemi için, sadece azami eşleştirmeyi değil, aynı zamanda doymuş düğümlerin indüklenmiş alt çizgelerinin herhangi bir döngüye sahip olmamasını istiyoruz. Bu amaçla, döngüsüzlük kısıtına ihtiyaç vardır. Belirli çizge sınıfları için oluşturulmuş bazı kesin ve yaklaşık algoritmalar olmasına rağmen, genel çizgeler için azami döngüsüz eşleştirmeyi bulan bir algoritma yoktur. Bu çalışmada dört algoritma oluşturulmuştur. Farklı yoğunluk seviyelerine ve boyutlara sahip rastgele oluşturulmuş çizgeler, bunların performanslarını analiz etmek için kullanılır. Algoritmalardan ikisi, Kompakt ve Ayrıştırma formülasyonları olan kesin algoritmalardır. Bunların verimliliğine bağlı olarak, Ayrıştırma daha iyi performans sergilemiştir. Diğer iki algoritma ise Modifikasyon ve Konstrüksiyon yaklaşımı olan buluşsal yöntemlerdir. Hem optimale yaklaşma hem de zaman verimliliği açısından Konstrüksiyon yaklaşımının Modifikasyon yaklaşımından daha iyi performans gösterdiği gözlemlenmiştir. Kesin ve buluşsal yöntemlerin en iyileri karşılaştırıldığında Ayrıştırma formülasyonu optimallik yönünden en iyi performansı göstermiştir ancak büyük boyuta sahip çizgeler için uygulanabilir değildir. Konstrüksiyon yaklaşımı daha kısa zamanda optimala yakın makul sonuçlar verir. Ayrıca çizge boyutu ve yoğunluğu büyüdükçe döngüsüzlük kısıtının etkisinin arttığı görülmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a_{ij}	Adjacency matrix
C	Cycle set
c	Constant
d	Degree
E	Edge
e'	Edge set among unsaturated vertices
e''	Edge set among unsaturated vertices that has minimum degree in the graph
G	Graph
n	Number of nodes in the graph
p	Density of the graph
r_{ij}	Binary variable indicating whether the edge among nodes i and j is in subgraph induced by saturated vertices or not
s	Subset
$\boldsymbol{S}$	Set of subset
V	Vertices
x_{ij}	Binary variable indicating whether the edge among nodes i and j is as matching or not
y	Binary variable indicating whether the vertex is saturated or not
i)	Saturated vertices set

LIST OF ACRONYMS/ABBREVIATIONS

DFS	Depth-first search
M	Matching
MaxMac	Maximum matching

1. INTRODUCTION AND LITERATURE REVIEW

Graph theory originated more than 28 decades ago when mathematician Leonhard Euler solved the Königsberg bridge problem. The problem was a long-standing puzzle involving the possibility of crossing all seven bridges that span a forked river running past an island without crossing them twice. Euler claimed that it is impossible to find a patlı that travels along all the vertices of a given polygon. His proof of the first theorem in graph theory consisted of only the physical arrangement of the bridges without proving the theorem itself. [1, 2]

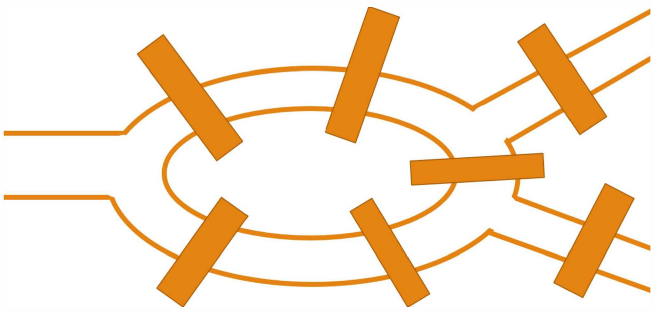


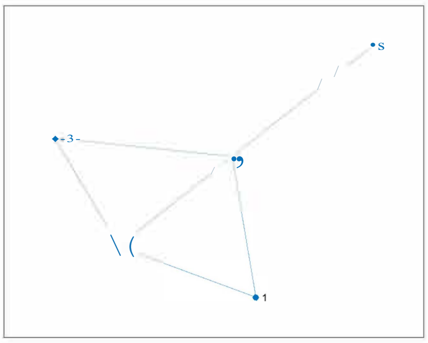
Figure 1.1: The Königsberg's Bridge.

Graph theory is the study of relationships in the sense of vertices and edges, which can help us understand and simplify the complex dynamic system. Also, graphs have many useful applications such as for finding shortest route to home at traffic [3] or in the recent days for possible spread of Covid-19 in the community through contacts [4,5]. Graph theory can provide answers to many questions about how things are connected, how to optimize networks, and how to match people or resources.

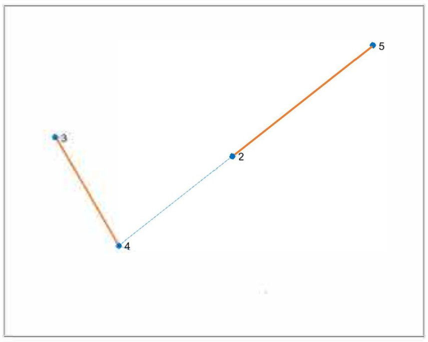
Matching problems involve a set of members where each member has a capacity and a subset of members ranks the other subset in order of preference. Matching implies the attempt to match each member to one (or more) acceptable member in a way that does not overburden their capacity. Examples of matching include assigning new doctors to hospitals [6], students to schools, and transplantation of human organs to recipients [7,8]. Although other one-to-many matching is exist such as matching of students to school, for the rest of this thesis, one-to-one matching will be referenced by the means of matching.

The classical matching problem is the maximum matching problem that searches for maximum number of nodes that are saturated by matchings [9]. Finding maximum matching is solvable in polynomial time in general graphs using Edmond's Augmenting Patlı Algorithm [10]. There are different variants of matching problems. Acyclic matching problem is a type of subgraph-restricted matching [11, 12]. Acyclic matching is a matching where the subgraph induced by the saturated vertices is acyclic [11]. lnduced matching [13] and uniquely-restricted matchings [14] are the other types of subgraph-restricted matching. The former one is a matching that satisfies the following: there are no two edges in the matching that are connected by any other edge in the graph [11, 15]. The latter one is defined by Golumbic et al. as a matching whose subgraph induced by saturated vertices has only one perfect matching where a matching is perfect matching, if and only if every vertex of it is saturated by a matching edge [11, 16].

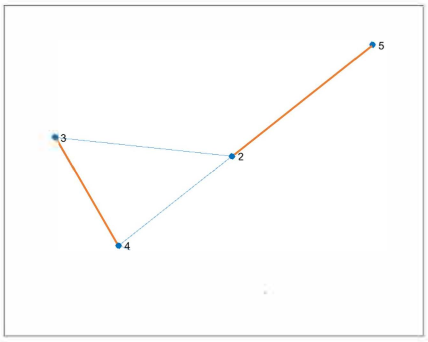
Another matching problem is maximal matching that is also known as inclusion-wise maximal matching and can be found easily by a greedy algorithm [11, 17]. If there are no additional edges can be added to matching set while preserving the matching property, the maximal matching on the graph is a inclusion-wise maximal matching. On the other hand, the minimum maximal matching problem is NP-hard in general which can be introduced as maximal matching in smallest size [18]. As minimum maximal matching, minimum weighted maximal matching is also an NP-hard problem which aims to obtain maximal matching which has lowest weight.



(a) Example graph with 5 node 6 edges.



(b) Example subgraph.



(c) Example induced subgraph.

Figure 1.2: Examples of the concept of subgraph and induced subgraph.

Before reviewing the literature on the maximum acydic matching, it will be helpful to introduce some specific types of graph dasses which are the subject of these studies:

- Based on Goddard et al., a graph is r-degenerate, if there exists a vertex in the every subgraph of the graph which has order 1 at least and the vertex's degree is at most r [12].
  - A matching is r-degenerate matching if the subgraph induced by the set of vertices incident to an edge in the matching is r-degenerate [12, 19].
  - 1-degenerate matchings are defined as acydic matching [12].
- A graph is bipartite if and only if it does not involve an odd eyde [20].
- A graph whose vertices can be divided as a complete graph and independent set is called as split graph [21].
- If every eyde of a graph, whose length is more than three, has a chord, then the graph is chordal [22].

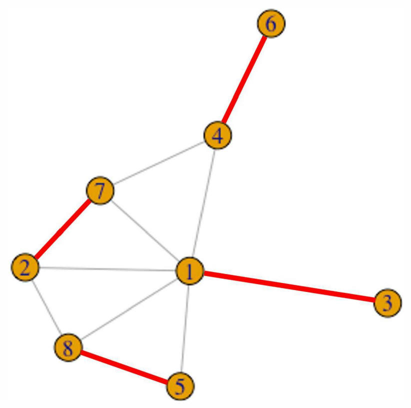
- Dually-chordal graphs are the clique graphs of chordal graphs [23].

Panda and Chaudhary indicate that the decision version of maximum acyclic matching is NP-complete for some specific subclasses of graphs which are comb-convex bipartite graphs and dually-chordal graphs [24]. Fürst and Rautenbach suggested that the decision version of maximum acyclic matching for bipartite graphs whose maximum degree is less than or equal to four is NP-complete [25].

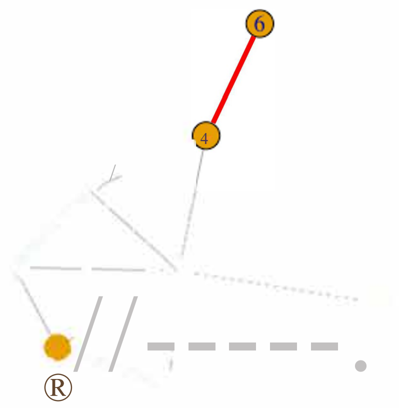
On the other hand, according to Panda et al., maximum size of an acycling matching in split graphs can be found in polynomial time by an approximate algorithm. Its complexity is declared as $O(n^7)$ [24]. it is also proposed by Furst et al. that maximum acycling matching can be found in polynomial time for $\bar{P}_4$-free graphs and $2P_3$-free graphs [25]. Baste and Rautenbach declared that maximum acycling matching can be found in polynomial time for chordal graphs by the help of r-degenerate matching [19]. By the studies of Fürst and Rautenbach, acycling matching number for connected subcubic graphs cannot be lower than $\frac{3}{n_1}{}^{G)}$ - c where n(G) is order of graph G and c is the constant [26]. Another study carried on by Baste et al. provides a lower bound which is (1 - o(l)) ! · fora graph which has n non-isolated vertices and maximum degree at most - They also suggest that maximum acycling matching can be found by 3/2-factor approximation and $\overline{(2}$(f ~~1))~~ - factor approximation algorithms for cubic and -regular graphs, respectively [27].

Additionally, an exact algorithm (AM-PIG(G)) is provided by Panda and Chaudhary which can compute maximum acycling matching size in polynomial time for proper interval graphs which is subclass of chordal graphs. Its complexity is the same as computing for split graphs [24].

For a given graph, $G = (V, E)$, V stands for vertices, and E denotes the set of edges between vertices. When there is a common vertex between two edges, it can be declared that the edges are adjacent to each other [28]. Matching in graph G is a set of pairwise independent edges such that no two edges are adjacent [11, 24]. Matching



(a) The maximum matching.



(b) The maximum acyclic matching.

Figure 1.3: Examples of matching.

can be thought as node pairing with existing edges [29]. In other words, matchings cannot have common endpoints [19]. Vertices incident to edges in a matching M are called saturated [28].

In this study, we are interested in maximum acyclic matching problem which is another variant of maximum matching problem. If every induced subgraph incident to saturated vertices is acyclic, then the matching is acyclic [24].

There is an example with 8 vertices and 0.4 density shown in the Figure 1.3. While dealing with the maximum matching problem, the aim is to saturate all possible nodes as can be seen in the Figure 1.3a. On the other hand, while finding the maximum acyclic matching, an additional restriction is needed to avoid any eyde created by edges that are induced by saturated vertices. In the Figure 1.3a, the matching between the nodes 1 and 3 saturates its endpoints. With the existence of other saturated vertices, which are the nodes 2, 4, 5, 7 and 8, the subgraphs induced by the node 1 and the others have cycles. Therefore, while seeking maximum acyclic matching, the edge between the nodes 1 and 3 is not selected as matching so that the graph induced by saturated vertices in the Figure 1.3b is a patlı. These vertices are unsaturated and shown as grey in the Figure 1.3b. In other words, the graph given in the Figure 1.3b shows the maximum matching which also have the property of acyclicity.

The main focus of the study is to find a matching of maximum size such that the graph induced by saturated vertices has no cycle. Depending on the literature, it is seen that there is no such comprehensive study to get the maximum acyclic matching in general graphs. it is known that the problem's challenging part is the acyclicity. However, in different areas, acyclicity constraint can be solvable in polynomial time. For instance, traveling salesman problem is a very well-known and studied problem that also deals with acyclicity constraint while trying to avoid subtours and there are several algorithms to solve in polynomial time. Therefore, the aim of the this study is to apply an polynomial time algorithm to the maximum acyclic problem by the help of literature.

As far as we know, there is no integer programming formulation or mixed integer programming formulation created for maximum acyclic matching problem. Nevertheless, it exists for the other types of matching problems such as the minimum weighted maximal matching problem [30, 31]. In this study, exact algorithms and heuristics are developed to achieve maximum acyclic matching. The former consists of one integer programming formulation which is the extensive algorithm and mixed integer programming formulation which is the cutting plane algorithm. The latter one contains modification and constructive approaches.

The outline of the thesis is ordered as follows. In Chapter 2, integer programming formulation for the maximum matching problem and, exact algorithms and heuristics for the maximum acyclic matching are given and explained. In Chapter 3, experimental results of all algorithms are compared in the terms of time efficiency, optimality, graph density, and derivation from maximum matching. In Chapter 4, significant outcomes of the study are explained.

2. PROBLEM FORMULATION

In this chapter, problem formulation for the maximum matching problem is formulated and explained in Section 2.1. Formulations of exact algorithms for the maximum acyclic matching problem are developed in Section 2.2. Heuristics for the maximum acyclic matching problem are explained in Section 2.3.

2.1. Problem formulation for the Maximum Matching Problem

In the model below, a_{ij} is adjacency matrix that takes value 1 if there exist an edge between i and j, otherwise it takes 0. $N(i)$ denotes the neighbours of vertex i. x_{ij}, y_i and r_{ij} are binary decision variables. x_{ij} represents the edges in the matching set. If edge $< i, j >$ is selected in the matching, x_{ij} takes value 1.

MaxMac Model:

$$\max \sum_{(<i,j>\in E)} x_{ij} \qquad (2.1)$$

$$\text{subject to} \sum_{(j\in N(i))} x_{ij} \leq 1 \qquad \forall i \in V \qquad (2.2)$$

$$x_{ij} \in \{0, 1\} \qquad (2.3)$$

As an objective, Model Max:Mac tries to maximize selected edges and Eq.(2.2) enforce it to be a matching. In other words, for every vertex i, at most one edge incident to i can be selected in the matching. Therefore, the model ensures obtaining maximum matching.

2.2. Exact Algorithm for the Maximum Acyclic Matching Problem

In this section, extensive and cutting plane formulations are interpreted.

2.2.1. Extensive Formulation

Starting from the description of the maximum acyclic matching, the subgraph induced by saturated vertices should be acyclic. For this purpose, the set C of all cycles in the graph is formed. In the Extensive Model below, y_i takes value 1 if node i is saturated by a matching. r_{ij} is a decision variable that is used to avoid any cycle in the subgraph induced by saturated vertices. Although y_i is defined as continuous, it is guaranteed that it can only take binary values by the Equation (2.5). As well as y_i, r_{ij} takes value 1 only if both y_i and y_j are 1 otherwise it takes 0.

Extensive Model:

$$\max \sum_{(<i,j>\in E)} x_{ij} \tag{2.4}$$

$$\text{subject to} \sum_{(j\in N(i))} x_{ij} = y_i \qquad \forall i \in V \tag{2.5}$$

$$x_{ij} \le y_i \qquad \forall i \in V, j \in N(i) \tag{2.6}$$

$$x_{ij} \le y_j \qquad \forall j \in V, i \in N(j) \tag{2.7}$$

$$r_{ij} \le y_i \qquad \forall i \in V, j \in N(i) \tag{2.8}$$

$$r_{ij} \le y_j \qquad \forall j \in V, i \in N(i) \tag{2.9}$$

$$r_{ij} \ge y_i + y_j - 1 \qquad \forall i,j \in V, i \le j \tag{2.10}$$

$$\sum_{(i\in S, j\in S)} r_{ij} \le |S| - 1 \qquad \forall S \subset C, S \neq \emptyset, |S| \ge 3 \tag{2.11}$$

$$x_{ij} \in \{0,1\} \qquad \forall i,j \in V \tag{2.12}$$

$$y_i \ge 0 \qquad \forall i \in V \tag{2.13}$$

$$r_{ij} \ge 0 \qquad \forall i,j \in V \tag{2.14}$$

There are some constraints added to Model MaxMac to get the maximum acyclic matching. By the definition of maximum acyclic matching, nodes saturated by the edges of the matchings are taken into account. Equation (2.5) forces that if no edge incident to a vertex i is selected in the matching then i is unsaturated, that is $Y_i = 0$. Equations (2.6) and (2.7) guarantee that if an edge is selected in the matching, its both end points i and j should be identified as saturated.

Equations (2.8) and (2.9) provide that if Y_i or Y_j is unsaturated, then the corresponding r_{ij} should be zero. Equation (2.10) satisfies that when both end points are saturated, r_{ij} can only take value 1 since it is a binary decision variable. As such, r_{ij} takes value 1 only if ij is an edge induced by two saturated vertices. The Equation (2.11) ensures that the algorithm concludes with an acyclic solution by forcing summation of r_{ij} to be less than the order of the corresponding subgraph for every possible subgraph.

Since all cycle subsets of vertices are taken into account, the Equation (2.11) provides potentially exponentially many constraints to the formulation.

2.2.2. Cutting Plane Formulation

Another algorithm is established by cutting plane formulation to get optimum solutions for larger graphs. Extensive Model has an exponential number of constraints which is why it can only be applicable for small graphs. A cutting plane algorithm is developed so that the constraints related to acydicity can be added when needed. 1f the graph induced by saturated vertices is acydic, then the solution is optimal. 1f not, a new constraint expressing that the obtained solution should be changed to avoid detected cydes is added to the formulation.

Because of the exponential number of constraints (2.11), the constraint is removed from the master problem and generated as needed by cuts. Master Problem aims to obtain maximum matching. The feasibility of the problem is checked by the subproblem which is a depth-first search(DFS) algorithm. Detecting cydes is briefly shown with pseudo-codes in Algorithm 2. 1f DFS algorithm cannot find any eyde in the induced subgraph by saturated vertices, then the solution is feasible and optimal. 1f else, vertex sets leading cydes are detected and related cuts are added by Equation (2.11). After adding cuts related to cydes, the master problem works again to find maximum matching. The algorithm stops when there is no eyde found by DFS. Hence, the process continues by adding cuts one by one until the solution becomes acydic.

To devise a cutting plane algorithm, Equation (2.11) is relaxed. By this relaxation, the problem can be solved with cutting plane. in the subproblem, cydes in the subgraph induced by saturated vertices are detected by depth first search algorithm. 1f there is no eyde detected, then the solution is optimal. 1f any eyde is detected in the subproblem which means current solution is not feasible. Based on detected cydes, eyde set C is updated and cuts are generated. After cut generation, algorithm returns to master problem and procedure continues as given in Algorithm 1.

Depth first search algorithm is also known as backtracking relies on labeling principle. Initially, all nodes are unlabelled. Starting from the initial nodes, successive

Algorithm 1 Cutting Plane Formulation

```
Input: $G = (V, E),\ S = 0$

1: Solve Extensive Model with relaxed constraint (2.11). Let $(x,y)$ be the optimal
   solution and G[y] be subgraph induced by $y$.
2: Detect cycles in G[y] by DFS if exist;
3: if There is no cycle in G[y] then
4:     $x$ is a maximum acyclic matching, STOP ;
5: else
6:     Generate cuts (2.11) for all cycles detected in Step 2 and add it to the Extensive
       Model;
7:     Go to the Step 1;
8: end if
```

nodes are discovered through the edges and marked. If successive vertex is already marked as visited, then there is a cycle. [32]

Algorithm 2 Depth-first Seareh Algorithm to Find Cydes.

```
  Input: fi, ai,j, S = 0 , s = O
  Output: The set S of all eydes indueed in G by saturated vertiees fi
DFS(s, i)
Marki as "visited" and add to s;
for j in N(i) do
  if j is not marked as "visited" then
    DFS(s, j)
  else
    if eard(s)    3 then
      There is a eyde. Add s to S
  end if
end for
```

2.3. Heuristics for the Maximum Acyclic Matching Problem

In this seetion, further strategies will be explained to aehieve the maximum aeydie mateching. These are the heuristies that developed by modifieation and eonstruetion approaehes.

2.3.1. Modification Algorithm

Modifieation algorithm is one of the heuristies. In this heuristie, maximum mateh-ing is given as input whieh is found by Model MaxMae. Thus, the maximum number of vertiees are saturated at first. If the subgraph indueed by saturated vertiees eauses any eyde, the modifieation algorithm removes one of the edges from the mateching set until there is no eyde exists. We remove edges depending on their deereasing degree where the degree of an edge is the summation of the degrees of its end vertiees in the original graph. After the removing proeedure is eompleted, the modifieation algorithm eheeks if there is any edge whose end vertiees do not form a eyde with existing ones;

such an edge is added to the matching. in other words, the acyclic matching is made maximal in a greedy fashion by adding edges as long as they do not form new cycles.

Algorithm 3 Modification Algorithm.

```
Input: $G = (V, E)$
Take a maximum matching $M$ using MaxMac;
if The subgraph induced by saturated vertices is acyclic then
  $M$ is optimal ;
else
  while The subgraph induced by saturated vertices is cyclic do
    Pick an edge $e$ of $M$ with highest degree in the original graph where degree
    of an edge is the sum of its endpoints;
    Remove $e$ from $M$;
  end while
  while There is an edge which does not lead to a cycle with existing saturated
  vertices do
    Add edge to $M$ ;
  end while
end if
```

2.3.2. Constructive Algorithm

Another heuristic is developed with constructive approach. As opposed to the Modification Algorithm, it starts with an empty matching set. While selecting the edge to be added to the matching set, the process starts with the edge that has the lowest degree. Then, the algorithm continues selecting edges with respect to ascending degree of edges that incident to unsaturated vertices. Here the intuition is that edges with low degree are less likely to create cycles. Adding proceeds one by one as long as end nodes of the newly added edge do not lead to cycle with existing ones.

In Algorithm 4, *ff* represents saturated vertices and *d(e)* symbolizes the degree of edge *e* which is the sum of the degrees of its endpoints. *G(V/f))* implies the subgraph induced by the unsaturated vertices and *e'* is an edge between unsaturated vertices. Finally, *e''* is the edge that has lowest degree in *G(V/f))*.

Algorithm 4 Constructive Algorithm.

```
Input: $G(V, E)$
Initialize M = 0;
  Find $e'' = min\{d(e')\} > 0$where $e'$ E $G(V/f))$;
  if M U$e''$ is acyclic then
    M := MUe";
  else
    Find the next $e'$ such that $e' / e''$;
    Update $e'' = min\{d(e')\}$;
    if $e'' = 0$ then
      STOP
    end if
  end if
```

3. EXPERIMENTAL RESULTS

Random graphs are generated by Erdos-Renyi to test the performance of all algorithms. For accuracy of the performance comparison, 10 samples are produced for each density and size level and their averages are reported in the given tables below. Densities are set to 0.2, 0.5 and 0.8. Through higher level of density, interaction between nodes increases. Exact algorithm and the maximum matching parts of study are conducted based on IBM ILOG CPLEX Optimization Studio 20.1 with Optimization Programming Language(OPL). Heuristics are implemented on RStudio. The performance of algorithms is compared and reported based on their efficiency. in the result tables, '1h+' means that the algorithm is stopped when the timer hits 1 hour. The result of the objective function near the '1h+' is collected at that point. Additionally, when the algorithm exceeds its limits or needs more than an hour to process, results are given as a dash.

3.1. Performance of Exact Algorithms on Maximum Acyclic Matching

Table 3.1: Comparison of Exact Algorithms wrt Number of Nodes

		Extensive		**Cutting Plane**	
n	p	**Obj**	**Time [s]**	**Obj**	**Time [s]**
	0.2	3.50	1.81	3.50	1.75
10	0.5	2.70	1.76	2.70	2.50
	0.8	2.00	2.09	2.00	1.92
	0.2	5.50	4.38	5.50	1.90
15	0.5	3.50	10.50	3.50	1.84
	0.8	2.00	35.31	2.00	4.07
	0.2	6.20	78.86	6.20	3.09
19	0.5	3.60	1267.63	3.60	7.72
	0.8	2.67	1036.07	2.67	6.01

Our two exact algorithms are given in the the first table below. Extensive formulation is formed by direct definition of maximum acyclic matching. it can be said that cutting plane algorithm reaches the maximum acyclic matching in less time. Since they are both exact algorithms there is no difference observed in the values of the objective function. The limit of extensive algorithm is much smaller than cutting plane one since in the extensive algorithm all possible subsets of nodes are created at one which leads to huge memory load and makes it impossible to use this algorithm for larger graphs.

3.2. Effect of Starting Point on Cutting Plane Algorithm

To enhance the efficiency of the cutting plane algorithm, outputs of the construction approach are used as the initial solution since the output is feasible and close to the optimal. in Tables 3.2 and 3.3, the performing time of the algorithm with initial solution is slightly higher for small graphs because of additional input reading. On the other hand, while the graph getting denser, the benefit of initial solution is noticeable. in other words, when graph size is higher than 30, given input decreases process time.

Table 3.2: Effect of Starting Point on Cutting Plane Algorithm wrt Number of Nodes

		Cutting Plane		**Cutting Plane with Initial**	
n	p	**Obj**	**Time [s]**	**Obj**	**Time [s]**
	0.2	3.50	1.75	3.50	3.57
10	0.5	2.70	2.50	2.70	3.78
	0.8	2.00	1.92	2.00	4.05
	0.2	5.50	1.90	5.50	3.69
15	0.5	3.50	1.84	3.50	6.00
	0.8	2.00	4.07	2.00	8.49
	0.2	6.20	3.09	6.20	7.57
19	0.5	3.60	7.72	3.60	12.24
	0.8	2.67	6.01	2.67	14.08
	0.2	6.50	5.33	6.50	7.32
20	0.5	4.10	6.55	4.10	12.28
	0.8	2.60	13.34	2.60	15.28
	0.2	8.50	12.07	8.50	10.25
30	0.5	4.60	30.66	4.60	23.12
	0.8	2.90	51.02	2.90	30.13
	0.2	9.60	353.74	9.60	146.86
50	0.5	4.70	909.42	4.70	475.15
	0.8	3.00	3182.12	3.00	1041.06
	0.2	10.70	2823.77	10.70	2010.65
75	0.5	5.80	1+	5.80	3079.31
	0.8	-	-	-	-

Table 3.3: Effect of Starting Point on Cutting Plane Algorithm wrt Graph Density

		Cutting Plane		**Cutting Plane with Initial**	
p	n	**Obj**	**Time [s]**	**Obj**	**Time [s]**
	10	3.50	1.75	3.50	3.57
	15	5.50	1.90	5.50	3.69
	19	6.20	3.09	6.20	7.57
0.2	20	6.50	5.33	6.50	7.32
	30	8.50	12.07	8.50	10.25
	50	9.60	353.74	9.60	146.86
	75	10.70	2823.77	10.70	2010.65
	10	2.70	2.50	2.70	3.78
	15	3.50	1.84	3.50	6.00
	19	3.60	7.72	3.60	12.24
0.5	20	4.10	6.55	4.10	12.28
	30	4.60	30.66	4.60	23.12
	50	4.70	909.42	4.70	475.15
	75	5.80	1+	5.80	3079.31
	10	2.00	1.92	2.00	4.05
	15	2.00	4.07	2.00	8.49
	19	2.67	6.01	2.67	14.08
0.8	20	2.60	13.34	2.60	15.28
	30	2.90	51.02	2.90	30.13
	50	3.00	3182.12	3.00	1041.06
	75	-	-	-	-

3.3. Effect of Acyclic Contraint on Algorithms

Table 3.4 is given to analyze how acyclicity constraint affects our best algorithms. From the wider point of view, dramatic increase in process times are observed when number of nodes hits 75, 30 and 50 for MaxMac, Cutting Plane and Construction algorithms, sequentially. However, due to lack of acyclicity constraint, increase in span time for MaxMac model is not as steep as the others. Deviation from the objective value of MaxMac problem is also expected for the other models who aims to achieve the maximum acyclic matching configuration.

Table 3.5 shows that when the density of graphs increases deviation from the maximum matching is also arising as expected. This is directly the result of acyclicity constraint. More precisely, when the graph becomes denser, acyclicity constraint becomes more binding constraint. That causes disclaiming from the number of matching achieved by MaxMac algorithm. Denser graphs contain large number of cycles, that is why induced graphs of saturated vertices tends to have eyde. Hence, number of subsets and also number of inequalities depending on acyclicity constraint leads to an increase in time which can be observed for both cutting plane and construction algorithms.

Table 3.4: Comparison with MaxMac wrt Number of Nodes

		MaxMac	Cutting Plane with Initial		Construction Approach	
n	p	Obj	Obj	Time [s]	Obj	Time [s]
	0.2	3.40	3.50	3.57	3.50	0.15
10.00	0.5	5.00	2.70	3.78	2.60	0.28
	0.8	5.00	2.00	4.05	1.90	0.44
	0.2	6.50	5.50	3.69	5.30	0.31
15.00	0.5	7.00	3.50	6.00	3.20	0.69
	0.8	7.00	2.00	8.49	1.80	1.91
	0.2	8.90	6.20	7.57	6.00	0.49
19.00	0.5	9.00	3.60	12.24	3.40	1.83
	0.8	9.00	2.67	14.08	2.20	4.06
	0.2	10.00	6.50	7.32	5.80	0.52
20.00	0.5	10.00	4.10	12.28	3.30	2.36
	0.8	10.00	2.60	15.28	1.80	6.04
	0.2	15.00	8.50	10.25	7.50	2.00
30.00	0.5	15.00	4.60	23.12	3.70	9.35
	0.8	15.00	2.90	30.13	2.30	28.66
	0.2	25.00	9.60	146.86	9.00	12.08
50.00	0.5	25.00	4.70	475.15	4.30	92.68
	0.8	25.00	3.00	1041.06	2.40	288.92
	0.2	37.00	10.70	2010.65	9.70	72.10
75.00	0.5	37.00	5.80	3079.31	4.50	613.95
	0.8	37.00	-	-	2.50	1704.85
	0.2	50.00	-	-	10.80	275.06
100.00	0.5	50.00	-	-	4.80	2165.99
	0.8	50.00	-	-	-	-

Table 3.5: Comparison with MaxMac wrt Graph Density

		MaxMac	**Cutting Plane**		**Construction Approach**	
p	n	**Obj**	**Obj**	**Time [s]**	**Obj**	**Time [s]**
0.2	10	3.40	3.40	1.75	3.50	0.15
	15	6.50	5.50	1.90	5.30	0.31
	19	8.90	6.20	3.09	6.00	0.49
	20	10.00	6.50	5.33	5.80	0.52
	30	15.00	8.50	12.07	7.50	2.00
	50	25.00	9.60	353.74	9.00	12.08
	75	37.00	10.70	2823.77	9.70	72.10
	100	50.00	-	-	10.80	275.06
0.5	10	5.00	2.70	2.50	2.60	0.28
	15	7.00	3.50	1.84	3.20	0.69
	19	9.00	3.60	7.72	3.40	1.83
	20	10.00	4.10	6.55	3.30	2.36
	30	15.00	4.60	30.66	3.70	9.35
	50	25.00	4.70	909.42	4.30	92.68
	75	37.00	5.80	1+	4.50	613.95
	100	50.00	-	-	4.80	2165.99
0.8	10	5.00	2.00	1.92	1.90	0.44
	15	7.00	2.00	4.07	1.80	1.91
	19	9.00	2.67	6.01	2.20	4.06
	20	10.00	2.60	13.34	1.80	6.04
	30	15.00	2.90	51.02	2.30	28.66
	50	25.00	3.00	3182.12	2.40	288.92
	75	37.00	-	-	2.50	1704.85
	100	50.00	-	-	-	-

4. CONCLUSION

The study focuses on maximum acyclic matching which aims to obtain the largest matching set such that subgraph induced by saturated vertices does not contain any cycle. The largest matching set is referred to as maximum matching, and it is well-known that the maximum matching problem is a polynomial-time solvable problem. However, it is noticed from the literature that maximum acyclic matching problem is just examined on specific graph classes. Methods solving maximum acyclic matching problem are utilized to fulfill the purpose of this thesis. Therefore, two exact and two approximate algorithms are formulated and evaluated.

Üne of the exact algorithms is an extensive one which process all the constraints at once. The other exact algorithm is a cutting plane formulation which contains relaxed acyclicity constraints that is where it differs from extensive formulation. Master problem of cutting plane formulation operates as a version of maximum matching algorithm while subproblem consists of detecting cycles in the subgraph induced by saturated vertices. Both exact algorithms are tested on the graphs which are randomly generated in different level of size and density. Based on the results of these tests, cutting plane formulation performed better than the extensive formulation with respect to running time.

üne of the heuristics is modification approach which takes maximum matching set as an input so that tries to deselect some edges from matching set in order to achieve acyclicity. After that process, if there is an edge which can be selected as matching that does not cause any cycle, modification approach searches for that. The other heuristics, construction approach, pursuits edges depending on their degree and selects as matching while saturate vertices' induced subgraphs does not trigger any cycle. They are also tested with the same samples. it is realized that construction approach predominates with regards to time efficiency and derivation from optimality.

The algorithms are also analyzed to understand how acyclicity constraint affects the derivation from optimal value of maximum matching problem based on graph size and density. it can be concluded that when graphs get denser, optimal value of maximum acyclic matching problem diverge from the optimal value of maximum matching problem, as it is expected. As a conclusion, the cutting plane formulation wins in the sense of optimality in this study. However, from the time point of view, construction algorithm provides near-the-optimal solutions in smaller period of time, and it can process larger graphs.

For future works, these methods can be compared with existing algorithms for specific graph classes.

REFERENCES

1. Euler, L., "Solutio Problematis ad Geometriam Situs Pertinentis", *Commentarii Academiae Scientiarum Petropolitanae,* pp. 128-140, 1741.

2. Biggs, N., E. K. Lloyd and R. J. Wilson, *Graph Theory, 1736-1936,* Clarendon Press, Oxford, 1986.

3. Nazari, S., M. R. Meybodi, M. A. Salehigh and S. Taghipour, "An Advanced Algorithm for Finding Shortest Patlı in Car Navigation System", *2008 First International Conference on Intelligent Networks and Intelligent Systems,* pp. 671-674, IEEE, 2008.

4. Bhapkar, H., P. N. Mahalle and P. S. Dhotre, "Virus Graph and COVID-19 Pandemic: a Graph Theory Approach", *Big Data Analytics and Artificial Intelligence against COVID-19: Innovation Vision and Approach,* pp. 15-34, Springer, 2020.

5. Davahli, M. R., W. Karwowski, K. Fiok, A. Murata, N. Sapkota, F. V. Farahani, A. Al-Juaid, T. Marek and R. Taiar, "The COVID-19 lnfection Diffusion in the US and Japan: A Graph-Theoretical Approach", *Biology,* Vol. 11, No. 1, p. 125, 2022.

6. Gorman, D., "Matching the Production of Doctors with National Needs", *Medical Education,* Vol. 52, No. 1, pp. 103-113, 2018.

7. Abdulkadiroğlu, A., P. A. Pathak and A. E. Roth, "The New York City High School Match", *American Economic Review,* Vol. 95, No. 2, pp. 364-367, 2005.

8. Sönmez, T. and M. U. Ünver, "Matching, Allocation, and Exchange of Discrete Resources", *Handbook of social Economics,* Vol. 1, pp. 781-852, Elsevier, 2011.

9. West, D. B., *Introduction to Graph Theory,* Prentice Hall, Upper Saddle River,

2001.

10. Edmonds, J., "Maximum Matching and a Polyhedron with O, 1-vertices", *Journal of Research of the National Bureau of Standards B,* Vol. 69, No. 125-130, pp. 55-56, 1965.

11. Lovasz, L. and M. D. Plummer, *Matching Theory,* American Mathematical Soc., North-Holland, 2009.

12. Goddard, W., S. M. Hedetniemi, S. T. Hedetniemi and R. Laskar, "Generalized Subgraph-restricted Matchings in Graphs", *Discrete Mathematics,* Vol. 293, No. 1-3, 2005.

13. Stockmeyer, L. J. and V. V. Vazirani, "NP-completeness of Some Generalizations of the Maximum Matching Problem", *Information Processing Letters,* Vol. 15, No. 1, pp. 14-19, 1982.

14. Golumbic, M. C., T. Hirst and M. Lewenstein, "Uniquely Restricted Matchings", *Algorithmica,* Vol. 31, No. 2, pp. 139-154, 2001.

15. Cameron, K., "Induced Matchings in Intersection Graphs", *Electronic Notes in Discrete Mathematics,* Vol. 5, pp. 50-52, 2000.

16. Lewin, M., "Matching-perfect and Cover-perfect Graphs", *Israel Journal of Mathematics,* Vol. 18, No. 4, pp. 345-347, 1974.

17. Ekim, T., "The Nobel Prize in Economic Sciences 2012 and Matching Theory.", *ICORES,* pp. 5-16, 2020.

18. Demange, M. and T. Ekim, "Minimum Maximal Matching is NP-hard in Regular Bipartite Graphs", *International Conference on Theory and Applications of Models of Computation,* pp. 364-374, Springer, 2008.

19. Baste, J. and D. Rautenbach, "Degenerate Matchings and Edge Colorings", *Discrete Applied Mathematics,* Vol. 239, pp. 38–44, 2018.

20. Bondy, J. A. and U. S. R. Murty, *Graph Theory with Applications,* Macmillan, Landon, 1976.

21. Foldes, S. and P. L. Hammer, "Split Graphs Having Dilworth Number Two", *Canadian Journal of Mathematics,* Vol. 29, No. 3, pp. 666-672, 1977.

22. Golumbic, M. C., "Algorithmic Graph Theory and Perfect Graphs, Acad", *Press, New York,* 1980.

23. Brandstiidt, A., F. Dragan, V. Chepoi and V. Voloshin, "Dually Chordal Graphs", *SIAM Journal on Discrete Mathematics,* Vol. 11, No. 3, pp. 437-455, 1998.

24. Panda, B. S. and J. Chaudhary, "Acyclic Matching in Some Subclasses of Graphs", *International Workshop on Combinatorial Algorithms,* pp. 409-421, Springer, 2020.

25. Fürst, M. and D. Rautenbach, "On Some Hard and Some Tractable Cases of the Ma:x:imum Acyclic Matching Problem", *Annals of Operations Research,* Vol. 279, No. 1, pp. 291-300, 2019.

26. Fürst, M. and D. Rautenbach, "A Lower Bound on the Acyclic Matching Number of Subcubic Graphs", *Discrete Mathematics,* Vol. 341, No. 8, pp. 2353-2358, 2018.

27. Baste, J., M. Fürst and D. Rautenbach, "Acyclic Matchings in Graphs of Bounded Ma:x:imum Degree", *Discrete Mathematics,* Vol. 345, No. 7, p. 112885, 2020.

28. Diestel, R., *Graph Theory,* Springer, New York, 2010.

29. Gerards, A., "Matching", *Handbooks in Operations Research and Management Science,* Vol. 7, pp. 135-224, 1995.

30. Taşkın, Z. C. and T. Ekim, "Integer Programming Formulations for the Minimum Weighted Maximal Matching Problem", *Optimization Letters,* Vol. 6, No. 6, pp. 1161-1171, 2012.

31. Ahat, B., T. Ekim and Z. C. Taşkın, "Integer Programming Formulations and Benders Decomposition for the Maximum Induced Matching Problem", *INFORMS Journal on Computing,* Vol. 30, No. 1, pp. 43-56, 2018.

32. Mehlhorn, K. and P. Sanders, *Algorithms and Data Structures: The Basic Toolbox,* Springer, Berlin, 2008.