21st International Research/Expert Conference
”Trends in the Development of Machinery and Associated Technology”
TMT 2018, Karlovy Vary, Czech Republic, 18th – 22nd September, 2018

SOLVING THE PROBLEM OF THE SCHEDULE CONCERNING CONFERENCE SESSIONS BY USING THE GRAPH THEORY

Hermina Alajbegović
Univerzitet u Zenici
Zenica, BiH

Almir Huskanović
Univerzitet u Zenici
Zenica, BiH

Štefko Miklavič
Univerza na Primorskem
Koper, Republika Slovenija

Dževad Zečić
Univerzitet u Zenici
Zenica, BiH

ABSTRACT
Let us suppose there is a scientific congress where scientists show their results in several sessions. A session schedule should be arranged so that two sessions in which the same scientist takes part, do not occur simultaneously. This problem is usually solved with the help of the graph theory. The aim of this paper is to show engineers how graph theory can be applied in solving problems related to the organization. And then using the Mathematica software we will show that we can easily get to the solution to the problem.
Keywords: *schedule of sessions, graph theory, chromatic index of a graph*

1. INTRODUCTION
A graph $G = (V, E)$ is an ordered pair, which consists of the vertex set V, whose elements are called vertices or points, and the edge set E, whose elements are called edges. In an undirected graph, an edge is an unordered pair of vertices. Otherwise, we have directed graph. A simple graph is an undirected graph without multiple edges (e.g. two or more edges connecting the same two vertices) or graph loops (edge which joins a vertex to itself). Multigraph is a graph which is permitted to have multiple edges or loops. Two vertices x and y are called adjacent or neighbors in G if $\{x, y\}$ is an edge of G. We write $x \sim y$ to denote this fact. If $e = \{x, y\}$ is an edge in G, then e is said to be incident with vertices x and y. The degree or valency of a vertex $x \in V$ is the number of edges that are incident with that vertex.

A graph colouring is an assignment of colours to the vertices of a graph such that no two adjacent vertices have the same colour. A graph G is called k –colourable if there exists a graph colouring of G with k colours.
The chromatic number $\chi(G)$ of a graph G is the minimal number k such that G is k –colourable. Such colouring of vertices is called minimal vertex colouring. For a graph G, let P(G,k) denotes the number of its colourings with k colours. It is known that there exists a unique polynomial P_G, which evaluated at any integer $k \geq 0$ coincides with P(G,k). Polynomial P_G is called the **chromatic polynomial** of G. The degree of the polynomial P_G is equal to the number of vertices of a graph G. The chromatic

number $\chi(G)$ of a graph G is the smallest natural number k such that $P_G(k) > 0$. The reader can find out more about chromatic polynomials in [2], [3], [4] and [6].

2. PROBLEM OF THE SCHEDULE CONCERNING CONFERENCE SESSIONS

Let us suppose there is a scientific congress where scientists show their results in several sessions. A session schedule should be arranged so that two sessions in which the same scientist takes part, do not occur simultaneously. Let us denote by N the set of all scientists, and by S the set of all sessions. For each session x, we denote by N_x the set of all scientists which should participate in session x. Then we can conclude that if $N_x \cap N_y \neq \emptyset$, then sessions x and y must occur in different times.

Let us now define a graph G, such that S is the vertex set of G. Vertices x and y are adjacent in G if and only if $N_x \cap N_y \neq \emptyset$.

Let us now consider a colouring of vertices of G. Note that if vertices x and y of G are of the same colour, then they are not adjacent, which implies that sessions x and y can occur simultaneously. It follows that the chromatic number $\chi(G)$ of G is the minimal number of time slots that we need to schedule the sessions in such a way, that two sessions in which the same scientist takes part, do not occur simultaneously.

There are several problems that can be reduced to finding the chromatic number of the graph. We talked about these problems in [1]. In order for the reader to have a better insight into the application of graph theory in the organization's problems, we will present another example. Suppose we have several types of materials that should be stored, but some materials cannot be deposited in the same warehouse due to possible turbulent chemical reactions between them. The question arises: How many warehouses are least needed to put materials away safely? And we can present this problem with graphs. Materials that need to be stored are the vertices of a graph, and two vertices are adjacent if and only if materials cannot be stored in the same warehouse. The chromatic number of this graph is the smallest number of warehouses required for safe storage of materials.

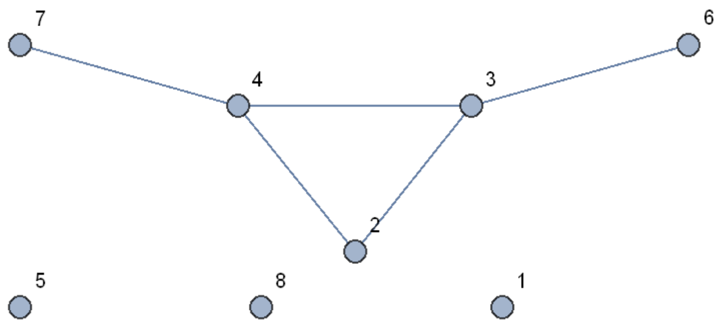
We have already said in the introductory section that the chromatic number of the graph can be determined from the chromatic polynomial. The chromatic polynomial can be calculated by a recursive formula. In order to state this formula, we need the following definitions. Let G be a simple graph and let $e = \{x, y\}$ denote an edge of G. Then $G - e$ is a graph obtained from G by deleting the edge e. Graph G/e is a graph obtained from G by deleting vertices x, y and replacing them with a new vertex z, such that the set of neighbours of z is equal to the union of the sets of neighbours of x, y. We could now state the above mentioned recursive formula.

Theorem 1 [2]. Let G be a simple graph and let e be an edge of G. Then $P_G(k) = P_{G-e}(k) - P_{G/e}(k)$.

Below we give a short code written for the Mathematica software that calculates the chromatic polynomial. This code, with the addition of several short lines, allows you to print graphs that are obtained in the steps of the recursive formula. ChromaticPolynomial[*g*] is an embedded command in the same software that calculates the chromatic polynomial of the graph *g*.

```
In[2]:= chrpol[vertex_, edge_] := Module[{v1, v2, e1, e2, v, n},
   n = Length[vertex];
   If[Length[edge] == 0,
    pol[x_] := x^n;
    chrpol[vertex, edge] = pol[k]];
   If[Length[edge] >= 1,
    v1 = vertex; e1 = Take[edge, {1, Length[edge] - 1}];
    v = edge[[Length[edge]]][[2]]; v2 = Sort[Complement[vertex, {v}]]; e2 = {};
    For[i = 1, i ≤ Length[edge] - 1, i++,
     If[MemberQ[{edge[[i]][[1]], edge[[i]][[2]]}, edge[[Length[edge]]][[2]]],
      AppendTo[e2, Sort[{edge[[i, 1]], edge[[Length[edge]]][[1]]}]],
      AppendTo[e2, Sort[{edge[[i]][[1]], edge[[i]][[2]]}]];
     ]];
    e2 = DeleteDuplicates[e2];
    pom1 = chrpol[v1, e1]; pom2 = chrpol[v2, e2];
    chrpol[vertex, edge] = chrpol[v1, e1] - chrpol[v2, e2]
   ]]
```

In order to implement the chromatic number search procedure, we assume that we have a conference with 8 sessions and that after the knowledge about the exhibitors we get the following graph:



Applying the above code to our graph, we get a chromatic polynomial.

```
In[6]:= graf = Graph[{1, 2, 3, 4, 5, 6, 7, 8}, {{2, 3}, {2, 4}, {3, 4}, {3, 6}, {4, 7}}];
   vertex = Sort[VertexList[graf]];
   edge = Sort[EdgeList[graf]]; chrpol[vertex, edge]
   chrpol[vertex, edge] == ChromaticPolynomial[graf, k]
```

Out[8]= $2k^4 - 7k^5 + 9k^6 - 5k^7 + k^8$

Out[9]= True

By analysing the sign of the chromatic polynomial, what we can do again using the Mathematica software package tools, we conclude that the chromatic number of the graph is equal to 3.

In[10]:= Reduce$\left[2k^4 - 7k^5 + 9k^6 - 5k^7 + k^8 > 0, k\right]$

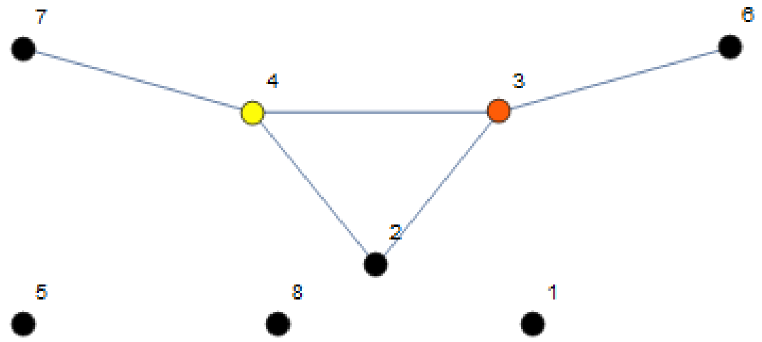
Out[10]= k < 0 || 0 < k < 1 || k > 2

In Mathematica software there are built-in tools for minimum vertex colouring. MinimumVertexColoring [g] is used for graphs with a small number of vertices, and for large graphs it can be used BrelazColoring[g]. Brelaz's heuristic algorithm finds good, but not necessarily minimum vertex colouring. You can find out more about using tools on official website of Mathematica. Ü. Ufuktepe and G. B. Turan also used Mathematica to color the graphs [5].

```
In[41]:= << Combinatorica`
         << GraphUtilities`
         graf = System`Graph[{1, 2, 3, 4, 5, 6, 7, 8}, {{2, 3}, {2, 4}, {3, 4}, {3, 6}, {4, 7}}];
         mvc = MinimumVertexColoring@ToCombinatoricaGraph[graf]
         SetProperty[graf, {System`VertexStyle → Thread[System`VertexList[graf] → (ColorData[3] /@ mvc)],
           VertexSize → Medium, VertexLabels → "Name"}]

Out[44]= {1, 1, 2, 3, 1, 1, 1, 1}

Out[45]=
```



Thus, the minimum number of time slots for session maintenance is 3. The sessions, whose corresponding vertices are colored with the same colour, can be maintained at the same time. The chromatic number in this paper was obtained using the chromatic polynomial and the command MinimumVertexColoring. We did this to confirm the result.

3. CONCLUSION

Many problems in the organization can simply be presented as problems in graph theory for which there are known algorithms that lead to their solution. In order to get solutions for the observed problems in practice, we can use the Mathematica software that has built-in tools for almost all known algorithms in mathematics. The theory of the graphs has been intensively developed over the past decade and we can expect its increasing application in various areas such as information technology, engineering, economics, chemistry, etc.

4. REFERENCES

[1] Alajbegović H., Huskanović A., Miklavič Š., Zečić Dž.: Teorija grafova i primjene, Sedmi međunarodni naučno-stručni skup “Obrazovanje, jezik, kultura: tendencije i izazovi”, 2018.
[2] Fouts C.: The Chromatic Polynomial,
[3] https://www.whitman.edu/Documents/Academics/Mathematics/fouts.pdf
[4] Read R. C.: An introduction to chromatic polynomials, Journal of combinatorial theory 4, 52-71 (1968)
[5] Hubai T., Lovasz L.: The Chromatic Polynomial, Master’s thesis, Eotvos Lorand University, (2009)
[6] Ufuktepe Ü., Turan G. B.: Applying Mathematica and webMathematica to graph coloring, Future Generation Computer Systems 23(2007):716-720 ·
[7] Weisstein E., Chromatic Polynomial, From MathWorld–A Wolfram Web
[8] Resource.http://mathworld.wolfram.com/ChromaticPolynomial.html