A PRACTICAL OVERVIEW.

POLYNOMIAL REDUCTIONS.

May 10, 2019

Jeffrey Manuel Pallares Nunez

Cadiz University
Superior Engineering School
manuel.pallanunez@alum.uca.es

1 Abstract

In this paper, i will present the exercises proposed in the assignment three of the subject Computational complexity. This exercises are about to demonstrate if the *Petersen graph*[1] is *n-colorable graph*[2], and the relations of the colorable problem with the *clique problem*.To do this i will use a (polynomial) reduction of this problem to *SAT problem*[3].

2 Introduction

To Start, I will develop a data structure to represent the graph, using a adjacency list in the programming language C++, so its necessary have a deep knowledge of this language to understand the logic of the attached files that encode the logic of the problem.

Its necessary too, learn about the *DIMACS*[4] format, because we will use a file in this format to feed the *PicoSAT*[5] program to know if our graph is n colorable. So its essential to know about this two topics if you want to get a deep understanding of the way that it works and to manipulate yourself the codes and to do your own experiments.

3 Methodology

The entire project its developed in a GNU/Linux distribution in specific Kubuntu. To replicate the results its necessary to have any distributions of GNU/Linux installed because its necessary both to run the applications and to install the necessary programs.

To do the experiments necessaries to answer the questions, I decided to code a data structure based in adjacency list to represent the graph, in the file called *Graph.hpp*. In this file there are auxiliary functions too, as factorial, or the reduction of the Graph to SAT problem, that resolve the entire problem. I decide to put it together to make easier the steps of compilation and execution without force the user of this research deal with makefiles and installs more applications. So to execute the main program, you only have to deal with one file, the *main.cpp* in which there is the declaration of the variables. This is the only file that you have to compile and execute with:

```
clang++ main.cpp -o main && ./main
```

To do that es necessary install the compiler *clang++*. This files are compiled with the compiler clang in his version 6.0, so I am not responsible for possible version failures if you use other compilers or versions.

The output of execute this program is a file called *dimacsformatSAT.txt* in this files is the reduction of the the n-colorable problem(2,3) of the Petersen Graph to SAT problem. To process this file I used the program *picoSAT*. Its necessary install this program. In my case I used the 960 version. To process the files use the follow command:

```
picosat dimacsformatSAT.txt --all
```

Or to process and get more information:

```
picosat dimacsformatSAT.txt --all -v
```

The Petersen graph is represented by a text file called *the_petersen_graph.txt*.

To finish, It's necessary to have all of this files mentioned before in the same directory. The specific format of some files, and his behaviors will be explained in detail in the following sections that I divided by file.

Graph.hpp

In this file, I coded the representation of a graph using a adjacency list. This class has member functions that are not implemented or used, because aren't necessary to to deal with our problem, but are declared to including it as future work, so you may ignore them if you study the code to get a more comprehension of my work. The important function here is the constructor of the class:

```
Graph::Graph(const char* name_file)
```

To construct the graph you only have to give him a file that represent the graph in a specific format, you can construct every graph not only the Petersen Graph. I overloaded the « operator:

```
std::ostream &operator<<(std::ostream& os, const Graph& g)
```

so you can print the graph with std::cout«, and check if is well built.

main.cpp

This file is the responsible of the execution of the program. In it, change the variable **ncols** we can generate the reduction of the problem with the numbers of the colors that you want. To replicate the results, use the values 2 and 3.

To construct the graph we use the call:

```
Graph g("the_petersen_graph.txt");
```

In which the parameter of *g* es the name of the file, to do this we need a file in the same directory with this name. To do the reduction we call the function:

```
ncol2SAT(g,ncols);
```

Where *g* es the graph constructed before and *ncols* the number of colours of the colorable problem that we will reduce. Generating the file *dimacsformatSAT.txt* that will be process with *picoSAT*.

the_petersen_graph.txt

In this file we have the presentation of the Petersen graph in this format:

```
0:1:4:5;
1:0:2:6;
2:1:3:7;
3:2:4:8;
4:0:3:9;
5:0:7:8;
6:1:8:9;
7:2:5:9;
8:3:5:6;
9:4:6:7;
```

Every row represent one vertex. In this case from 0, to 9 to represent the 10 vertices that form the Petersen graph. This vertices are represented by the first number of the row, following by ":" symbol .The next numbers represent the vertices that are connected with the vertex represented by the first number of the row. At the end we add ";" character to represent that the connection sequence is finished. Every row have 4 numbers, because in the Petersen graph, every vertex are connected with another 3 vertices.

dimacsformatSAT.txt

In this file is the reduction of the problem of n-colorable of the Petersen graph to SAT problem. The format of the reduction is done following the way described in the conference given by *Richard M. Karp* called *"NP-Complete problems"* [6]. In witch we have strings representing that a specific vertex could be colored by the numbers of colors that are represent by numbers $1 \dots n$ and a set of string that represent that a vertex could be coloured only with one colour at the same time. This file is written following the *DIMACS*[4] format.

This reduction is constructed by:

$N_v = n \times Num\_vertices$, being n the number of colours and N_v the number of variables

$N_c = 1 + \frac{n!}{2(n-2)!} \times Num\_Vertices + n \times Num\_Edges$, being n the number of colours and N_c the number of clauses.

This is not a polynomial reduction, but we are resolving the instances of the problem with only three, and two colours so the number of variables remains equals, but the number of clauses is determined int he case of three colors by:

$N_c = 4 \times Num\_vertices + 3 \times Num\_Edges$, being n the number of colours and N_c the number of clauses.

And for two colors :

$N_c = 2 \times Num\_vertices + 2 \times Num\_Edges$, being n the number of colours and N_c the number of clauses.

Being the reduction possible in polynomial time.

Tests

The tests have consisted in, once generated the necessary files, compiling and executing the program with *ncols* variable with the values two and three, use the *picoSAT* problem to process the *dimacsformatSAT* file, to know if the Petersen graph is colorable or not through the reduction of the problem to *SAT*. The output of the *picoSAT* problem give to us if the equation of the *SAT* is satisfiable with all the possible combinations of coloring the graph with the *n* number of colors and the number of solutions. Giving to us the necessary information to answer the questions of the assignment.

4 Results

I will put the results of the executions of the *picoSAT* program over three and two colors, the number of possible solutions and if are satisfiable or not, answering the question proposed in the assignment.

Reducing 3COL to SAT

In this section i will answer the questions of the first part of the assignment, which is about the reduction about the 3 coloring problem to SAT, and if the Petersen Graph is 3 colorable using the reduction before mentioned.

4.1 What kind of reduction is obtained?

First I will talk about the reduction. The reduction of the problem is a polynomial reduction as we see before. The order of the construction of the reduction its defined by the equation:

$N_c = K \times (3 + 3 \times 3 \times X \times 3)$, being N_c the number of clauses.

That is the "time" to write all the clauses in the file. K is the number of vertices and X is the number of edges. But we know that $X < K$ and its constant (equals to 3). So we can say that $\in O(n \times k)$.The *picoSAT* application works as a black box and we cant analyze his behavior, beyond of reading the file in which case its about reading all the lines/clauses, so this part $\in O(n)$. Doing the maximun rule, we can say that *ncols* function $\in O(n \times k)$. So the reduction its a polynomial reduction.

4.2 Using PicoSAT to determine if the Petersen graph is 3-colorable

According to *picoSAT* the Petersen graph is 3-colorable. In a reduction with 30 different variables and 130 clauses, we get 120 different solutions in which the SAT equation is satisfiable for 3 colors. Giving two examples,coded by triplets by vertex in which the positions represent the colors in this order | Blue, Green, Red |. The first example represented for the figure number 1 and the output of the *picoSAT* program:

s SATISFIABLE
v | -1 -2 3 | -4 5 -6 | -7 -8 9 | -10 11 -12 | 13 -14 -15 | -16 17 -18 |
v | 19 -20 -21 | 22 -23 -24 | -25 -26 27 | -28 -29 30 | 0

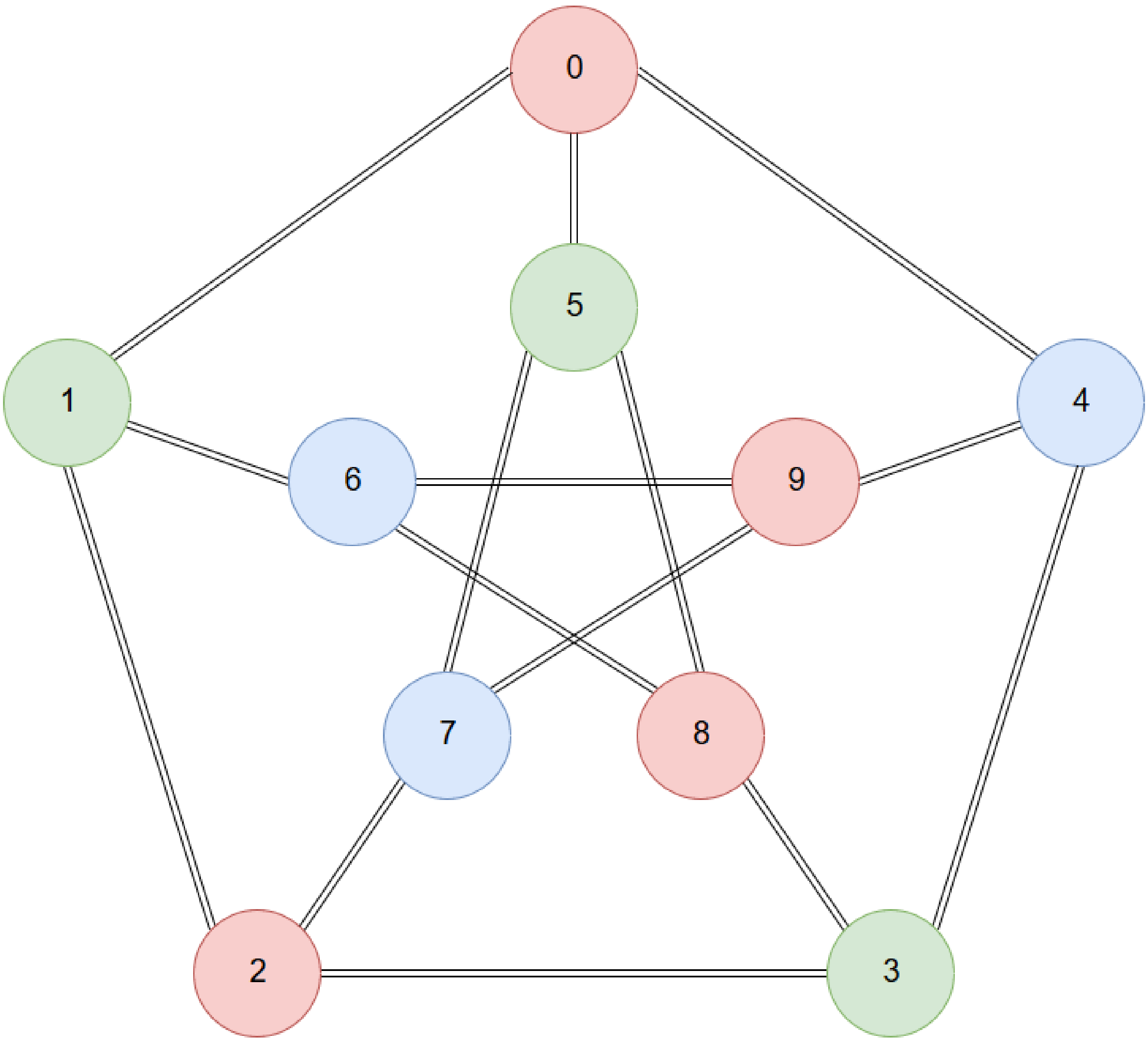

Figure 1. A possible coloring solution for the Petersen Graph.

The second example represented for the figure number 2 and the output of the *picoSAT* program, with the same code and order representing the colors | Blue, Green, Red |:

s SATISFIABLE
v | 1 -2 -3 | -4 5 -6 | 7 -8 -9 | -10 -11 12 | -13 14 -15 | -16 -17 18 |
V | 19 -20 -21 | -22 23 -24 | -25 26 -27 | -28 -29 30 | 0

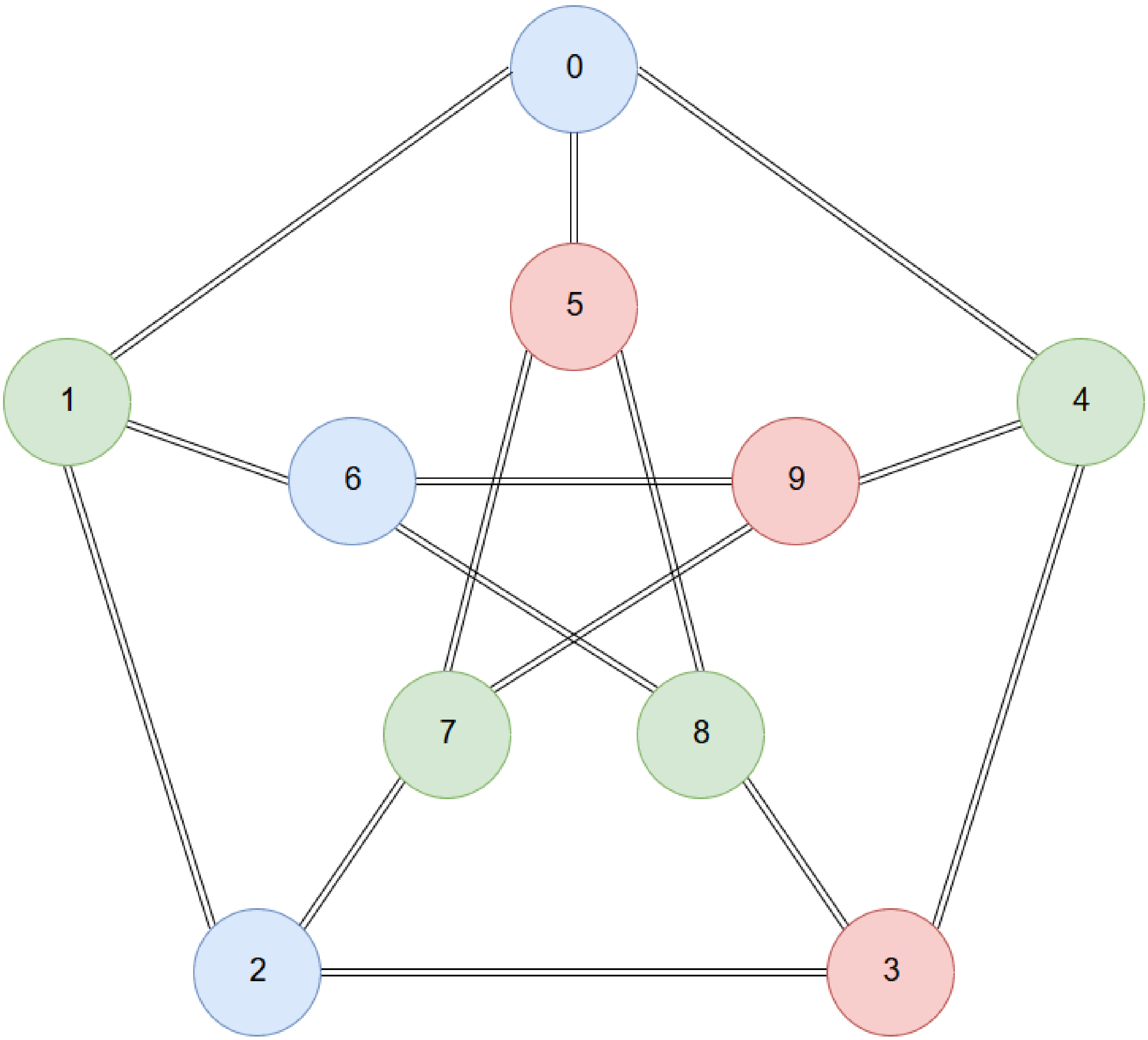

Figure 2. A possible coloring solution for the Petersen Graph.

Cliques and colors

In this section, will be discussed and answered the questions of the last part of the assignment, which is about the relation about the *Clique* problem and the *Coloring* problem.

4.3 Which is the clique number of the Petersen graph?

The clique number of a graph is the maximum numbers of vertices contains in biggest clique that the graph has[7]. In this case is easy to know, all the vertices has three edges, but not form a cycle between them, so the maximum. clique size is two, so the clique number its two.

4.4 Which is it the chromatic number?

The chromatic number of a graph is the minimum numbers of colours needed to color a graph[8]. In this case i got the solution trying to resolve the problem with 2 colors, and it cant, so the minimum numbers of colours needed to color the graph is three, the chromatic number its three.

4.5 Could it have contained a clique of size four? Why? The relation between colors and cliques

The answer its no, if the Petersen Graph had a clique of size four, would be impossible coloring the graph with 3 colors, because would have 4 nodes that are adjacents, being impossible coloring this sub-graph with less than 4 colors, in other words, are a directed relations between the clique problem and the coloring problem. And its it, the numbers of colors needed to color a graph its always greater or equals than the clique number, but it cant be less.

5 Conclusions

In conclusion, you cant do a reduction of the n-colorable problem to SAT problem in polynomial time, the number of combinations gets involved in the construction of the reduction, but you can do a polynomial time reduction of a instances of n-colorable problem as 2-colorable or 3-colorable problem to SAT.

In specific we worked with the Petersen Graph, which his chromatic number is three in spite of have a clique number of two, because his structure in which a vertex has 3 adjacent vertices forming a 5 point starts. Following the rule that the number of colors are lowerbounded by the clique number.

References

1. Weisstein, E. The petersen graph. DOI: http://mathworld.wolfram.com/PetersenGraph.html (2019).
2. Leighton, T. The colorable graph problem. DOI: https://www.geeksforgeeks.org/graph-coloring-applications/ (2010).
3. Baumgartner, A. The sat problem. DOI: https://www3.risc.jku.at/education/courses/ss2013/ars/presentations/presentation_baumgartner.pdf (2019).
4. Dimacs format. *Lab. Math. Log. St.Petersburg Dep. Steklov Inst. Math. Russ. Acad. Sci.* DOI: https://logic.pdmi.ras.ru/~basolver/dimacs.html (2019).
5. Wheeler., D. A. Picosat application. DOI: https://www.systutorials.com/docs/linux/man/1-picosat/ (2019).
6. Karp, R. M. Video of conference of np - complete problems. *Univ. California.* Second Part: from 31:10 to the end. (2019).
7. Weisstein, E. W. Clique number. *From MathWorld-A Wolfram Web Resour.* DOI: http://mathworld.wolfram.com/CliqueNumber.html (2019).
8. Weisstein, E. W. Chromatic number. *From MathWorld-A Wolfram Web Resour.* DOI: http://mathworld.wolfram.com/ChromaticNumber.html (2019).