**A Quick General Guidline for executing the PHP Program**

The algorithm in this repository is called 'Partial Circuit Heuristic Algorithm\_1' or 'PCHA\_1'. The algorithm in the next repository is given the title of 'Partial Circuit Heuristic Algorithm\_2 or 'PCHA\_2'. This repository gives an explanation about 'PCHA\_1'.

The general guideline is given below in 9 steps. These steps may vary according to the different type of computers and the different versions of XAMPP Webservers , PHP etc.. that are used in each type of computer.

1) Download 'XAMPP' for Windows from the internet and install 'XAMPP' Web server.

2) Create a folder for eg:- 'tsp\_programs' in "C:\xampp\htdocs" folder.

3) Copy the php files ( ie :- 'tsp\_1\_db\_PCHA\_1.php' , 'tsp\_2\_db\_PCHA\_1.php' , 'tsp\_3\_db\_PCHA\_1.php' and 'tsp\_4\_db\_PCHA\_1.php' ) and paste the php files into ' C:\xampp\htdocs\tsp\_programs' folder.

4) Open 'XAMPP' Control Panel and start Apache first and then start MySql.

5) Open Google Chrome browser and in the URL address, type 'localhost/phpmyadmin' and press enter.

The phpmyadmin page will be shown.

6) From phpmyadmin , create a database named 'tsp\_data\_for\_program\_database'.

( VERY IMPORTANT. PLEASE NOTE. When creating the new database, the database name should be exactly 'tsp\_data\_for\_program\_database'. If the name is different, the php files will not work. )

7) Import the example tables into "tsp\_data\_for\_program\_database" database. An example table is the file 'tsp\_vertex\_15\_table.sql'.

8) Open a new tab in Google Chrome browser and in the URL address, type 'localhost/tsp\_programs' and press enter.

9) A listing of the php files will be shown for eg:- one of the files shown will be 'tsp\_1\_db\_PCHA\_1.php'. From 'tsp\_1\_db\_PCHA\_1.php', the execution of the program can start, by clicking on 'tsp\_1\_db\_PCHA\_1.php'.

( IMPORTANT. PLEASE NOTE. When the execution starts, please select a table from the select box containing the tables in the beginning page of the execution. )

**A General Note about the PHP Program.**

The main advantage of this heuristic program, is that the program takes a very small amount of time to get the result.

Some results which are not got in PCHA\_1 may be got in PCHA\_2. Also some results which are not got in PCHA\_2 may be got in PCHA\_1. So both algorithms are useful. But the main difference is that PCHA\_2 takes much less time for getting a result than PCHA\_1.

"https://en.wikipedia.org/wiki/Christofides\_algorithm" is the website address for an article about Christofides Algorithm. Also in the article it mentions that the results are got within 3/2 of the optimal value. Also Triangle inequality condition needs to be satisfied for Christofides Algorithm to work.

A short form for 'Partial Circuit Heuristic Algorithm\_1' can be written as 'PCHA\_1'. In PCHA\_1 , the Triangle inequality condition need not be satisfied and the results that are got are better than 3/2 of the optimum solution length.

This PHP program finds the 'polynomial time' heuristic solution to the Travelling Salesman Problem for a 'Complete' graph using a database, so that data does not have to be typed in always.

This is a general heuristic program in 'PHP' for solving the Travelling Salesman Problem for any number of vertices.

The input data during the execution of the program is only the name of each vertex and the weight of each edge present in the graph.

( PLEASE NOTE :- If for an example graph, if there is no edge between two vertices, then assign a weight to the imaginary edge connecting the two vertices. The weight can be equal to the sum of the weights of all the real edges present in the example graph. This imaginary edge with a weight is necessary because of the condition that this program works on 'Complete' graphs only. A 'Complete' graph is the exactly the same mathematical object used in Graph Theory. Also please note that no edge in the example graph should have a weight of 0. )

To work with PHP files , 'XAMPP' Web Server has to be installed in the Computer. 'XAMPP' Web Server is available on the internet.

After installing XAMPP and when XAMPP Web Server is running , it is better not to connect to the internet , because of 'XAMPP' itself being a Web Server and so when XAMPP is running and if the computer is connected to the internet, other insecure connections can become connected to the computer. So as far as possible, once XAMPP is fully installed , and before starting to run XAMPP, it is better to disconnect the computer from the internet. After php program has finished executing, and Apache and Mysql has been stopped from the XAMPP control panel , and when XAMPP has been quit completely, then the computer can be connected to the internet.

The site from where data for the example tables is taken is 'https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html'. The website contains bigger data problems which can be used as input data when running this program.

Name of datasets and their correct minimum weights according to website are as given below.

1) DANTZIG42 ( Dataset of 42 vertices ). Correct minimum weight according to website equals 699.

2) FRI26 ( 26 vertices ). Correct minimum weight is 937.

3) GR17 ( 17 vertices ). Correct minimum weight is 2085.

4) PO1 ( 15 vertices ). Correct minimum weight is 291.

I am giving below the results of the program execution that I had obtained while running 'PCHA\_1'. The results consists of the dataset name, the minimum weight obtained by 'PCHA\_1' and the time taken for 'PCHA\_1' to execute. I executed 'PCHA\_1' on a Toshiba Satellite C640 Notebook computer. The results are as given below.

1) DANTZIG42 ( 42 vertices ). Minumum weight obtained is 819. Time taken is 1 minute and 27 seconds.

2) FRI26 ( 26 vertices ). Minimum weight is 1010. Time taken is 5 seconds.

3) GR17 ( 17 vertices ). Minimum weight is 2183. Time taken is 3 seconds.

4) PO1 ( 15 vertices ). Minimum weight is 291. Time taken is 1 second.

**Important details about database program**

In this program , a 'Mysql' database is used , so that input need not always be typed in when executing the program. There are four php files which contain the php code for executing the program with a database. 'tsp\_1\_db\_PCHA\_1.php' , 'tsp\_2\_db\_PCHA\_1.php' , 'tsp\_3\_db\_PCHA\_1.php' and 'tsp\_4\_db\_PCHA\_1.php' are the four php files. The order of execution of the php files is 'tsp\_1\_db\_PCHA\_1.php' , 'tsp\_2\_db\_PCHA\_1.php' , 'tsp\_3\_db\_PCHA\_1.php' and then 'tsp\_4\_db\_PCHA\_1.php'. The execution need only to be started from 'tsp\_1\_db\_PCHA\_1.php'.

The 'XAMPP' Web Server is used for executing the php files and for processing the table data present in the database. All the four 'php' files ( ie:- 'tsp\_1\_db\_PCHA\_1.php' , 'tsp\_2\_db\_PCHA\_1.php' , 'tsp\_3\_db\_PCHA\_1.php' and ' tsp\_4\_db\_PCHA\_1.php' ) can be put in 'C:\xampp\htdocs\tsp\_programs' folder. The execution need only to be started from 'tsp\_1\_db\_PCHA\_1.php'.

The database files are as follows :-

1) "tsp\_vertex\_15\_table.sql" is the sql import file for the 15 vertex graph.

2) "tsp\_vertex\_17\_table.sql" is the sql import file for the 17 vertex graph.

3) "tsp\_vertex\_26\_table.sql" is the sql import file for the 26 vertex graph.

4) "tsp\_vertex\_42\_table.sql" is the sql import file for the 42 vertex graph.

If any of the example tables is imported into the database , it should always be imported into the database named " tsp\_data\_for\_program\_database". The 'phpmyadmin' database interface can be used for importing the table sql files.

An example of a table is as follows. A table 'vertex\_15' is present as an example of data of a fifteen vertex graph in the "tsp\_vertex\_15\_table.sql" table. For running the example of a 'fifteen' vertex graph, the 'vertex\_15' can be selected from the drop down list of the tables , when the file 'tsp\_1\_db\_PCHA\_1.php' is executed.

Also the format of the example tables in the database should be followed exactly for creating new example tables , otherwise the program will not work. An example of what is meant by format of the table, is that in the example table, there is no primary key column. So addition , deletion etc.. of data , while in phpmyadmin , cannot be done quickly on the tables, if changes need to be made on the data in the tables. So when creating a new table, create a temporary primary key column in order to quickly insert data into the example tables. After all the data has been entered into the example table, then the primary key column can be removed. ( Please note that a creation of a primary key column is not necessary and need only be created if data needs to be entered faster while working in phpmyadmin. )

**Important properties of the PHP Program**

The program is designed to solve for any number of vertices for the Travelling Salesman Problem for Complete graphs.

The program is an iteration type program and will give 1 or more hamilton circuits which may have the minimum weights. If more that one hamilton circuit is output, take the hamilton circuit with the least weight as the hamilton circuit with the most minimum weight.

The number of vertices to be entered in the program can only be a positive integer which is equal to or greater than 4.

When a 33 vertex graph or a graph containing more than 33 vertices are given for processing to the program, errors showing 'undefined weights' and 'input values of 1000 exceeded' are given as output if the default value of the constant 'max\_input\_vars' in php is 1000.

In a 33 vertex graph, there are 1056 edges ( ie:- 33\*32 = 1056 , since there are 33 vertices in the graph and there are 32 edges attached to each vertex ) or 1056 input values. 1056 is greater than 1000 and so the program will output an error since the number of input values exceed 1000.

For eg:- if a 50 vertex graph is to be processed, then there are 2450 edges ( ie:- 50\*49 = 2450 , since there are 50 vertices in the graph and there are 49 edges attached to each vertex ) or 2450 input values. So 'max\_input\_vars' can be given a value higher than 2450 for eg:- 3000. So the 'max\_input\_vars' constant in php can be changed by uncommenting the line 'max\_input\_vars=1000' and giving a value of 'max\_input\_vars=3000' to the 'max\_input\_vars' constant.

The 'max\_input\_vars' constant can be found in 'php.ini' file which is present in 'C:/xampp/php' folder.

To verify if 'max\_input\_vars' has the required value, the function 'phpinfo()' can be typed in any empty php file and the file executed. This information displayed after execution of the 'phpinfo()' function will contain the value of 'max\_input\_vars' along with other information that the function will display.

Vertex names which are input into the program should begin with an english alphabet. The rest of the name should contain a combination of english alphabets or positive integer numbers or a combination of both english alphabets and positive integer numbers only. No other type of characters are allowed to be used in the vertex name.

Each vertex name should be different from every other vertex name. No two or more vertices should have the same name.

Only positive numbers should be used for the weight values of the edges.

No edge should have a weight with a value of 0.

In "tsp\_4\_db\_PCHA\_1.php" , in line no 3 and line no 4, the following code is given. The code is as follows :-

**Code Begin**

ini\_set('memory\_limit', '1024M');

ini\_set('max\_execution\_time', 10800);

**Code End**

( Note:- '**Code Begin**' and '**Code End**' are just markers to show the beginning and the ending of the code. )

The above code will become necessary when the number of vertices increases. This code is for increasing the processing memory and for increasing the execution time limit. If this code is not there, then sometimes php will stop program execution with error messages. For eg:- if the processing time exceeds 30 seconds, then php will automatically stop execution after 30 seconds, since 30 seconds is the default value given in php. To prevent this, the code 'ini\_set('max\_execution\_time', 10800);' is given.

Also for the code given below as

**Code Begin**

'<?php

session\_start();

?>'

**Code End**

( which is present at the very beginning of a php file ) please note that there should not be a space or even an empty line above this code if any code modification is made in the php file. Even if there is a single space before this line of code, the error of 'header output already sent' is most likely to be shown. So please make sure that this code (ie:- the code included between the single apostrophies ) is present at the very beginning of the code file, without even a single space preceeding this line of code.