

Approximation Algorithms for the Maximum Happy Vertices Problem: User Guide

This document contains descriptions on how to compile and use a program for finding upper bounds and achieving approximate solutions to the maximum happy vertices colouring problem. The algorithms used in this program are described in the following paper:

- Lewis, R., Thiruvady, D. and Morgan, K. "Finding Happiness: An Analysis of the Maximum Happy Vertices Problem" *Computers and Operations Research*, doi: 10.1016/j.cor.2018.11.015

This program has been written in C++ (V11) and is available at <http://rhydlewislew.eu/resources/happyAlgs.zip>. It has been successfully compiled in Windows using Microsoft Visual Studio and in Linux using the GNU Compiler g++. Instructions on how to do this now follow.

Compilation in Microsoft Visual Studio

To compile and execute using Microsoft Visual Studio the following steps can be taken:

1. Open Visual Studio and click **File**, then **New**, and then **Project from Existing Code**.
2. In the dialogue box, select **Visual C++** and click **Next**.
3. Select the subdirectory containing these files and click **Next**.
4. Finally, select **Console Application Project** for the project type, and then click **Finish**.

The source code can then be viewed and executed from the Visual Studio application. Release mode should be used during compilation to make the program execute at maximum speed.

Compilation with g++

To compile the source code in Linux, please use the included **makefile**.

Usage

Once generated, the executable file should be run from the command line. If the program is called with no arguments, the following usage information will be printed to the screen.

```
Algorithm for the Maximum Happy Vertices Problem
USAGE:
-----
<InputFile>      (Required. File must be in DIMACS format)
-a <int>          (Algorithm (1/2). 1 = Greedy-MHV (assign all free vertices to same colour, take the best of the k options);
                  2 = Growth-MHV (Algorithm described by Li and Zhang).
                  Default = 1)
-r <int>          (Random seed. Default = 1)
-----
```

This provides the user with information needed to produce valid commands. Note that random number generation is only used with the Growth-MHV algorithm. Here are some example commands.

```
happy_graph.txt
```

This produces a solution to the problem specified in the file **graph.txt** using the default Greedy-MHV algorithm. The file **graph.txt** is included with this resource. Note that an upper bound for the number of happy vertices is generated using the method described in the above paper.

```
happy_graph.txt -a 2 -r 123
```

This produces a solution to the problem specified in **graph.txt** using the Growth-MHV algorithm together with the random seed 123.

On termination, the solution is written to the screen along with various run details. The same information is also written to the file **out_***, where ***** is the name of the input file.

In addition, the solution is also written to the file **sln_***, where ***** is the name of the input file. The first line of this file contains the number of happy vertices. The remaining lines then contain the colour of each vertex in order, using colour labels 1 to k.

Input Format:

Problem instances for this algorithm should be in the DIMACS format. We recommend using the problem instance generator available at <http://rhydLewis.eu/resources/happyGen.zip>.

For ease of reading, the vertices within the file are labelled in descending order of degree. An input file contains the following information.

- Lines beginning with a **c** contain comments and can be ignored;
- The single line beginning with **p edge** contains, respectfully, the number of vertices, the number of edges, and the number of vertices preassigned to a colour;
- Lines beginning with **e** state the graph adjacencies (i.e., the endpoints of each edge);
- Lines beginning with **n** state the precolourings by giving, respectively, the vertex number and its colour.

For reference, an example graph, **graph.txt**, is included with this resource.

Copyright notice

Redistribution and use in source and binary forms, with or without modification, of the code associated with this document are permitted provided that citations are made to the publication mentioned at the start of this document. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. This software is provided by the contributors “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage. This software is supplied without any support services.

Please direct any queries/comments to Rhyd Lewis: web: www.rhydLewis.eu, email: LewisR9@cf.ac.uk

R. Lewis (Last updated Friday, 23 November 2018)