# Using Harmony Search Algorithm to Solve the N-Region Four Color Map Problem

[1]Romie B. HORCA, [2]John Paul T. YUSIONG
[1,2]Division of Natural Sciences and Mathematics,
University of the Philippines Visayas Tacloban College
Tacloban City, Leyte, 6500, Philippines
[1]ya10_romie07@yahoo.com, [2]jpyusiong@gmail.com

***Abstract*-The Harmony Search (HS) algorithm is a relatively recent addition to the family of optimization algorithms. It imitates the behavior of musicians when composing their music such as random playing of notes, previous composition-based play, and pitch-adjusted play. The algorithm was used in solving the Four Color Map Problem. In this problem, it is said one can color the regions of a map using a maximum of four colors only with the constraint that no neighboring regions should share the same color. Neighboring regions mean two regions having a common boundary, not just a common point. Simulation results proved the feasibility of HS as the primary optimization technique in solving sample maps.**

***Keywords:* Optimization, Harmony Search Algorithm, Four Color Map Theorem, Map Coloring**

## I. INTRODUCTION

Nowadays, computers have become an essential tool in our society. It was first intended for arithmetic computing, but now it became a tool for the advancement in every field of science and technology. Its innovations and development made human lives easier. The usage of computers had become so vast that almost anything we know is run or made by computers.

Computer science is a field that deals with finding solutions to problems that we encounter [3]. It has wide variety of applications ranging from abstract analysis of algorithms and grammars, to areas like programming languages, hardware and software.

In computer science, problems can be categorized in two main groups: P problems and NP problems. A problem is considered a P problem if there exist at least one algorithm to solve that problem such that the number of steps of the algorithm is bounded by a certain length of input [4]. An algorithm is an important tool that helps computers process information. Without an algorithm, the performance of a computer will not be efficient. However, certain programs can do better than other programs in accomplishing the same kind of problem using a different algorithm.

On the other hand, NP stands for nondeterministic polynomial time. Naturally, it means that a solution to any search problem can be found and verified in polynomial time (time required to solve a problem on a computer) by a special sort of algorithms [12].

NP complete problems can be solved using optimization algorithms. Optimization means solving problems by selecting the best element from set of available options. It tries to minimize or maximize a real-valued function by systematically selecting the values of variables from a given range of allowable values [13].

In this study, the researchers aim to solve The Four Color Map Problem, an example of an NP complete problem [10] by using an optimization algorithm called Harmony Search Algorithm. It is an optimization algorithm inspired from the behavior of musicians when they are composing music.

Since Francis Guthrie raised the question about the Four Color Map Conjecture to De Morgan last 1852, mathematicians and scientists were challenged in finding the proof of this conjecture. The NP-complete nature of this problem compelled some mathematicians and scientists to devote their life in finding a proof to this conjecture [1] [2].

The succeeding pages are arranged as follows. Section 2 provides a discussion on the Four Color Map problem. Section 3 explains the Harmony Search Algorithm followed by its implementation in Section 4. Results of the experiments are discussed in Section 5 and finally the conclusion in section 6.

## II. FOUR COLOR MAP THEOREM

In mathematics, the four color map theorem states that, given some division of a plane that has neighboring regions (connected without a break), which forms a figure called a *map*, at most four colors are needed to color the regions of the map such that no two neighboring regions have the same color. Two regions are said to be *adjacent* if and only if a common boundary is shared, not just a common point [7].

Fig. 1 illustrates the adjacency of regions in the given map sample. Regions A and B have a common boundary, same goes with regions C and D. But regions A and D and regions B and C only share a common point. Therefore, by definition, regions A and B are adjacent, as well as regions C and D, but not regions A and D and regions B and C.

The proof of Kenneth Appel and Wolfgang Haken on the theorem was met with both triumphs and dismay. They have provided a solution to the challenging problem raised by Francis Guthrie by using a computer that bothered many mathematicians because it was feared that the correctness of the proof would only be checked with the aid of a computer [1] [2].

The Four Color Problem is said to be one of the great mathematical problems because the problem is understood even by a non-mathematician. Just like the map in Fig. 2, this research focuses on coloring a map with the aid of an algorithm known as Harmony Search Algorithm.


Fig. 1. Adjacency of regions.

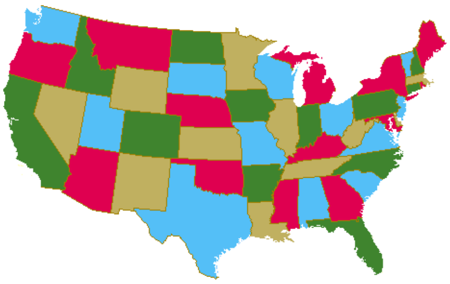

Fig. 2. A colored USA map using four colors.

## III. THE HARMONY SEARCH (HS) ALGORITHM

There is a relationship between music and optimization. In the Harmony Search Algorithm, the two interesting fields of studies, music and computer science are combined. The behavior of musicians when they are creating their music has a resemblance in the optimization process: Each musical instrument represents a decision variable; a musical note corresponds to the value of each variable; and the harmony corresponds to a solution [5][6].

Jazz musicians, when they are composing their music, either play notes randomly, play notes based on experiences, or adjusting the pitch in order to find a fantastic harmony. In order to find an optimal solution, variables in the harmony search algorithm are assigned with values that are either random or are taken from previously-memorized good values [6].

To better understand the harmony search model it is important to study first the inspiration that led to the creation of the said algorithm. It is believed that when musicians create their music, they use three techniques to achieve harmony. These are (1) playing using randomly selected notes, (2) playing music from their experiences, and (3) adjusting the tone to better harmonize the music [5] [6].

Geem et al [5] [6], noticed the similarity of this behavior in achieving the optimal solution to a problem. Thus in 2001, they proposed three methods corresponding to the three techniques namely the use of (1) random selection, (2) memory consideration, (3) and pitch adjustment. These became the elements of the newly developed meta-heuristic optimization algorithm called the Harmony Search Algorithm [9] [14].

Just like when musicians play a random pitch within the instrument range, in random selection, random values are picked from the range of possible values of a certain variable [12] [14].

Also, similar to a musician that plays any preferred pitch from his previous composition or memory, in memory consideration, values are chosen from the vectors stored in harmony memory [12] [14].

Once a pitch is obtained from memory, a musician can further adjust the pitch to the neighboring pitches to obtain a better harmony. In pitch adjustment, the value is adjusted with a certain probability. This value may or may not move to neighboring values with a definite probability [12] [14].

The flow of the harmony search algorithm can be summarized by the following code sequence: [14]

Harmony Search Algorithm

**Begin**
  Define objective function $f(x)$, $x = (x_1, x_2 \ldots x_d)^T$
  Define harmony memory accepting rate ($r_{accept}$)
  Define pitch adjusting rate ($r_{pa}$) and other parameters
  Generate Harmony Memory with random harmonies
  **while** *(t < max number of cycles)*
    **while** *(i < number of variables)*
      **if** ($rand<r_{accept}$) choose a value for var i
        **if** ($rand<r_{pa}$) adjust the value
        **end if**
      **else**
        choose a random value
      **end if**
    **end while**
    accept the new memory if better
  **end while**
  pick the best solution
**End**

The overall process of the harmony search algorithm can be illustrated in Fig. 3, where it can be generalized into three main steps [14]:

1. **Initialization:** it is where parameters are defined and the harmony memory is being filled with random harmonies or candidate solutions.
2. **Improvisation:** in this process, a new solution is created using the three methods of the harmony search algorithm. This step is repeated until a termination condition is met.
3. **Selection:** after improvisation, the best harmony is selected in the harmony memory to represent the solution to the problem.

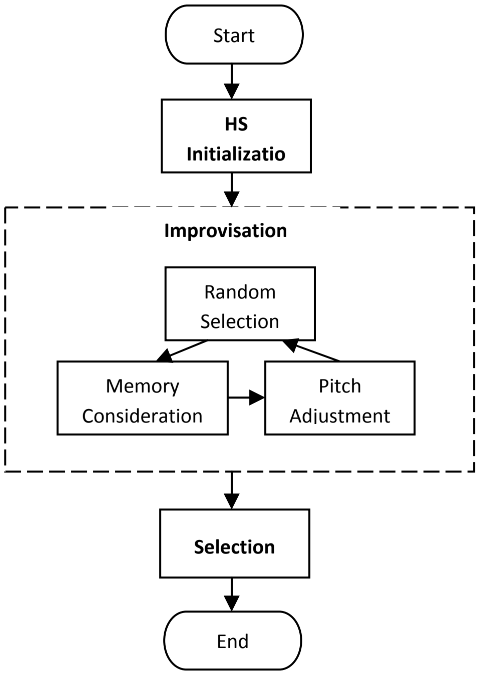


Fig. 3. The Harmony Search algorithm.

In order for the Harmony Search Algorithm to start, certain factors must be considered first. Some parameters must be defined first before the optimization process begins.

1. **Number of decision variables:** each harmony is composed of several decision variables.
2. **Number of cycles of iteration:** one of the termination conditions of the optimization process.
3. **Harmony Memory Size:** refers to the number of solutions that will be stored in the harmony memory.
4. **Harmony Memory Consideration Rate ($r_{accept}$):** the rate at which the value of the decision variables from the harmony memory is picked as elements of the New Harmony that will be created.
5. **Pitch Adjustment Rate ($r_{pa}$):** the probability that the decision variable picked from the harmony memory be altered by some certain amount.

The capability of Harmony Search Algorithm in solving problems has been proven effective in various studies such as the Traveling Salesman problem and Sudoku puzzle. In this study, the researchers aim to determine the feasibility of the said algorithm in solving one of the challenging problems in the field of mathematics, the Four Color Map problem [1] [2].

## IV. Solving the Four Color Map Problem Using HSA

In this section, we will discuss the approach that the researchers took in achieving a four colored map solution. Fig. 4 illustrates the overall flow of the proposed approach.

Each candidate solution is represented by vectors whose variables correspond to the regions of the map. Vectors are array of integer variables that serves as the solution to the problem. Regions have values 1, 2, 3, or 4 that represent its four colors. The size of the solution is equivalent to the number of regions present in the map. Fig. 5 shows the representation of the solution. The general formula of each candidate solution is as follows:

$$R = <r_1, r_2, r_3 ..., r_n> \quad (1)$$

where $\boldsymbol{r_n}$ represents the regions having values 1, 2, 3, or 4 and n is the number of regions of the map.

The optimization process starts by generating random solutions in the harmony memory. Equation (1) illustrates the characteristics of each candidate solution in the memory. Each possible solution is evaluated for their respective objective value. Then the optimization process begins and ends if the process generates an optimal solution or the maximum improvisation is reached.

After initializing the harmony memory, it is followed by the improvisation process. This is where a new harmony or a possible solution is generated through random selection, harmony memory consideration or pitch adjustment.

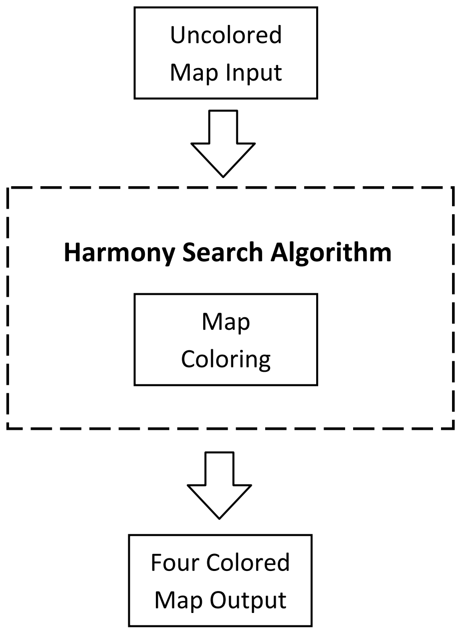


Fig. 4. HS-based approach to solving the map coloring problem.

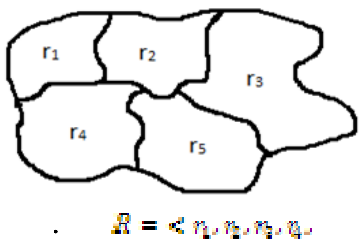


Fig. 5. Vector Representation of the map.

*A. Random Selection*

Random Selection is where a variable of a harmony takes a random value; in this problem it will select one random color from the available four options as illustrated in Fig. 6.

*B. Memory Consideration*

Values or colors in the harmony memory have a chance to be selected when performing memory consideration as shown in Fig. 7.

*C. Pitch Adjustment*

Selected value from the memory can be altered to obtain a variation through pitch adjustment. Fig. 8 illustrates how pitch adjustment is done.

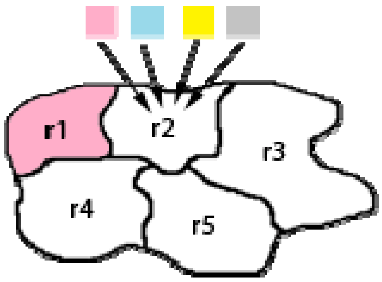


Fig. 6. Random Selection for $r_2$.

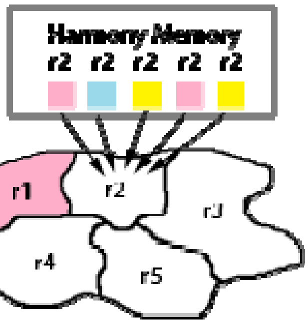


Fig. 7. Memory Consideration for $r_2$.

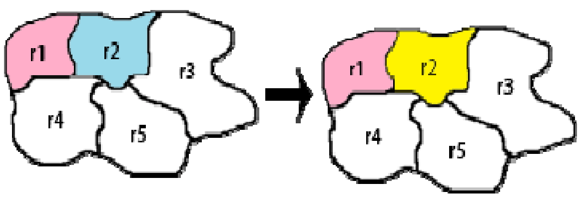


Fig. 8. Pitch Adjustment for $r_2$.

*D. Fitness Function*

Solutions in the memory and the newly generated solution are tested using equation (2).To assess the fitness of a harmony vector, it should undergo a fitness evaluation using the evaluation function. The following equation will be used as evaluation function to accomplish this objective:

$$Minimize\ R = \sum_{i=1}^{n} \sum_{j=i+1}^{n} m_{ij}\, r_{conflict} \qquad (2)$$

where $r_{conflict} = \begin{cases} 1 & if\ r_i = r_j \\ 0 & otherwise \end{cases}$, and $m_{ij}$ refers to the value of the adjacency information of the map. The adjacency information of each region $\mathbf{r_n}$ in an adjacency matrix $\boldsymbol{M}$ is given as follows:

$$M = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1j} \\ m_{21} & \ddots & \cdots & \cdots \\ \vdots & \cdots & \ddots & \cdots \\ m_{i1} & \cdots & \cdots & m_{ij} \end{bmatrix} \qquad (3)$$

where: $m_{ij} = \begin{cases} 1 & if\ r_i\ adjacent\ r_j \\ 0 & otherwise \end{cases}$.

The evaluation function will return a value that will represent the fitness of a harmony vector. The function aims to minimize the result, wherein, the smaller the fitness value, the better the vector is. The value also represents the number of errors present in the solution. If the fitness value reaches zero, then that is an optimal solution and the optimization process stops.

*F. Illustrating the HS-based Process to Map Coloring*

Using the map in Fig. 5, we will run the harmony search algorithm and show how it sequentially solves the map. We assign color pink = 1, light blue = 2, yellow = 3, and gray = 4.

**Step 1:** Initialize random harmony vectors that will represent the solution for the colored map using equation (1). In this example, we set our harmony memory to three.

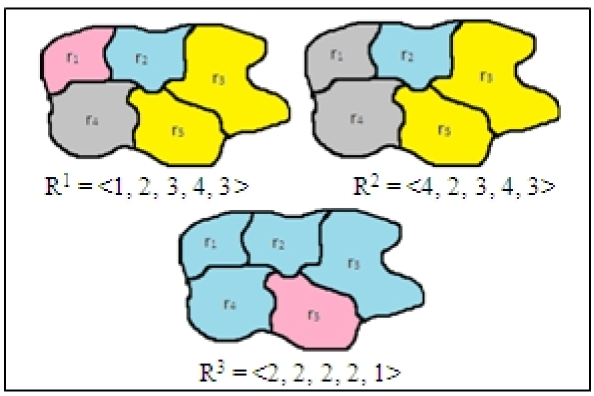


Fig. 9. A Harmony Memory with HMS=3.

**Step 2:** Improvise a new harmony vector through Random Selection, Memory Consideration, and Pitch Adjustment.

**Step 3:** Evaluate the harmony using the fitness function in Equations (2) and (3). According to the formula, only half of the matrix is evaluated since the matrix is symmetric. The smaller the resulting value the better is the result. Let's evaluate the harmony memory and compare it to the New Harmony. $R^1$ is evaluated as shown in Equation 4. Using the same function, the result of $R^2$=2 and $R^3$=4.

$$R^1 = \begin{array}{ccccc} 0\ + & 1x0\ + & 0\ + & 1x0\ + & 0\ + \\ & 0\ + & 1x0\ + & 1x0\ + & 1x0\ + \\ & & 0\ + & 0\ + & 1x1\ + = 1 \\ & & & 0\ + & 1x0\ + \\ & & & & 0 \end{array} \quad (4)$$

**Step 4:** Steps 2 and 3 are repeated until a series of finite number of improvisation is achieved. The smallest value or the best harmony in the harmony memory is selected to become the solution. Assuming that after a certain number of improvisations, an optimal solution is found in the harmony memory as shown Fig. 10.

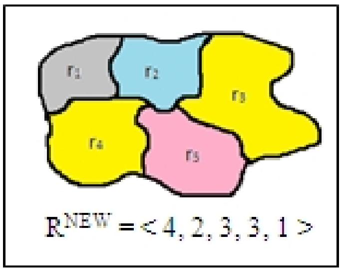


Fig. 10. The Optimal Solution.

After the improvisation the process, the best harmony in the memory will now be selected to become the output. The map will be colored using the corresponding colors of the values in the solution.

## V. Experiments and Results

The experiments were performed with the goal of testing the feasibility of HSA in solving four color map problem and also to assess the effects of varied map inputs in the efficiency of HSA in finding an optimal solution to the problem. The setups used were subjected to 30 runs in order to have satisfactory results. Table I shows the standard parameter values used for the experiments.

The assessment of the feasibility of HSA was done using three test map cases as shown in Fig. 11. These are the China map, USA map, and the 52-Region map. These maps were used as test cases in various researches. Equations (2) and (3) were the basis of evaluation of the fitness.

TABLE I
STANDARD HAS PARAMETER VALUES

| Parameter | Cycles | HMS | HMCR | PAR |
|---|---|---|---|---|
| Value | 1,000,000 | 10 | 0.9 | 0.15 |

Fig. 11. Test map cases

The results of this experiment were tabulated in Table II. Based on the results; it is observed that the Harmony Search Algorithm has successfully solved the three types of map.

The map of China having 32 regions and the map of USA having 48 regions are relatively quicker to solve compared to the map with 52 regions. It is because the 52-region map has a high number of adjacency per region, each region has 5 to 6 adjacent regions. While the China and USA maps have low number of adjacency per region.

This can be attributed to the fact that the 52-region map has a more complex adjacency information compared to the two maps. It was observed that the complexity of the adjacency relation of the map contributes to the difficulty in finding an optimal solution. Fig. 15 to Fig. 17 show sample results on the three test map cases.

Table II shows the average processing time and average number of cycles, respectively. The table reflects similar results in time and cycles in achieving the optimal solution.

TABLE II
HS PERFORMANCE IN SOLVING THE THREE TEST MAPS

| | China Map | USA Map | 52-Region |
|---|---|---|---|
| Average Cycles | 86,217.37 | 148,154.8 | 12,056,040 |
| Average Time (s) | 1.3 | 66.033 | 268.83 |

To further evaluate the performance of HS, another experiment was conducted. It was to test the efficiency of HS with random map inputs. The purpose of this experiment is to know the behavior of HSA when different random maps are applied. The numbers of regions of the maps are set to 30, 40, 50, 60, and 70 with random number of adjacency per region. HSA parameters are also set to their standard values and executed 30 runs. The results of this setup were tabulated in Tables III.

Fig. 15. Optimal coloring of the China map.

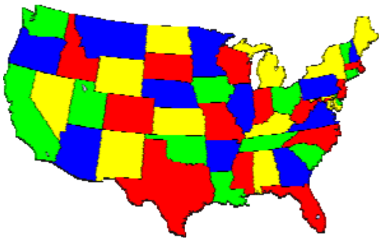

Fig. 16. Optimal coloring of the USA map.

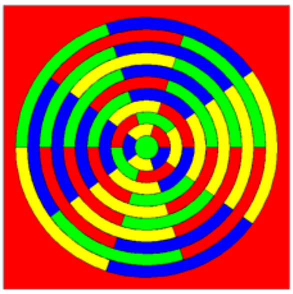

Fig. 17. Optimal coloring of the 52-Region map.

TABLE III
HS PERFORMACE IN SOLVING RANDOM MAPS

| Map Sizes | Average Cycles | Average Time (s) |
|---|---|---|
| Set A 30 | 140, 269.00 | 1.73 |
| Set B 40 | 263,629.60 | 4.10 |
| Set C 50 | 1,025,429.00 | 22.83 |
| Set D 60 | 2,664,367.00 | 76.20 |
| Set E 70 | 5,424, 564.00 | 206.53 |

Tables III indicate a relative increase in improvisation cycles as the map sizes become larger. The results indicate that as the size of the map increases, the time and number of cycles also increases. This is because as the size of the map increases, the more difficult it is to find an optimal solution.

## V. Conclusions

The Harmony Search Algorithm (HSA) has been proven to be an effective underlying algorithm in solving the Four Color Map Problem because based on the experiments results HSA was able to find an optimal solution for all types of sample maps. This is also the case for random maps with varying number of regions. Also, the HAS-based map solver was able to find an optimal coloring even when the adjacency information of the maps becomes more complex.

## References

[1] Appel, K. and Haken W.: Every planar map is four colorable. Part I. Discharging, Illinois J. Math. 21 (1977). p. 429-490.

[2] Appel, K. and Haken W., Koch, J.: Every planar map is four colorable. Part II. Reducibility, Illinois J. Math. 21 (1977). 491-567

[3] Computer Science Department, College of Saint Benedict, Saint John's University. From, http://www.csbsju.edu/computer-science.html (Accessed: November 2010).

[4] Cook, S.: The P versus NP Problem. University of Toronto. From,http://www.claymath.org/millennium/P_vs_NP/Official_Problem_Description.pdf (Accessed: November 2010).

[5] Geem, Z.W., Kim, J.H., and Loganathan, G.V.: A New Heuristic Optimization Algorithm. Harmony Search Simulation (2001).

[6] Geem, Z.W.: Music-Inspired harmony Search Algorithm - Theory and Applications. Studies in Computational Intelligence, Vol. 191 ISBN: (2009). 1-12

[7] Gonthier, G.: A computer-checked proof of the four colour theorem. From, http://en.wikipedia.org/wiki/Four_color_theorem. (Accessed: November 2010).

[9] Holland, J., *Adaptation in Natural and Artificial Systems.* MIT Press, Cambridge, MA, (1992).

[10] Kumar, A.: Using a Genetic Algorithm Approach to solve the N-Region Four Colour Map Problem. First International Conference on Emerging Trends in Engineering and Technology (2008).

[12] Papadimitriou, C.H., Dasgupta, S., and Vazirani, U.V.: NP-complete Problems. Algorithms (2006), 257-258.

[13] Papalambros, P.Y. and Wilde, D.J.: Principles of Optimal Design: Modeling and Computation, Cambridge University Press. ISBN 0-521-62727-3 (2000).

[14] Romero, V.M., Tomes L.L., Yusiong, J.P.T.: Tetris Agent Optimization Using Harmony Search Algorithm. International Journal of Computer Science Issues, Vol. 8, Issue 1. (2011). p. 22 – 31.

**Romie B. Horca** received his B.S. degree in Computer Science from the University of the Philippines Visayas Tacloban College (UPVTC), Tacloban City, Leyte, Philippines in 2011. He is currently employed as an Instructor at the Global School for Technological Studies, Inc., Catarman, Northern Samar, Philippines.

**John Paul T. Yusiong** received his M.S. degree in Computer Science from the University of the Philippines Diliman (UPD), Diliman, Quezon City, Philippines in 2006 and his B.S. degree in Computer Science *(cum laude)* from the University of the Philippines Visayas Tacloban College (UPVTC), Tacloban City, Leyte, Philippines in 2002. He has been teaching for nine years and he is currently an Assistant Professor in Computer Science at the University of the Philippines Visayas Tacloban College, Tacloban City, Leyte, Philippines. His research interests include Artificial Intelligence, Neural Networks and Optimization algorithms.