
A GENETIC ALGORITHM FOR GRAPH MATCHING USING NODE CHARACTERISTICS

VIDYA M SHETTAR, SHANMUKHAPPA A ANGADI

Abstract: Graph Matching or graph isomorphism checking is a popular problem in graph theory and has varied applications. Many approximate and exact solutions have been proposed for the different variants of the graph matching problem. In this paper a new two stage methodology for graph matching using node characteristics and a novel genetic algorithm for finding vertex correspondences is proposed. The methodology makes use of a hypothesis relating the vertices of two supposedly similar graphs and finds the correspondences between the vertices of the two graphs using a novel genetic algorithm. The paper presents a new chromosome structure and defines the genetic operators on the same. The proposed methodology is implemented using MATLAB and the results obtained are robust, efficient and satisfactory. All the graph pairs tested have provided consistent and 100% results and the methodology can be used for graph matching applications in all the domains.

Keywords: Graph Matching, Genetic Algorithm, eccentricity, degree invariance

Introduction: Graph matching, in the sense used here, refers to the process of matching or finding similarity of two graphs. Graph matching has found applications in various domains such as pattern recognition, computer vision, chemical structure analysis, bioinformatics etc. Due to the large number of applications spawned by it, researchers are exploring the applicability of new computing paradigms for efficient graph matching solutions. Soft computing and evolutionary computing have tremendous potential for offering efficient solutions to the graph matching problem. Soft computing techniques mimic the approach employed by human beings and different techniques such as neural networks, fuzzy system, support vector machines etc, are available and have been used for various applications including graph matching [1]. Similarly evolutionary computation techniques such as genetic algorithms, genetic programming, ant colony optimization etc, have found applications in many problem domains. Evolutionary computational techniques are fundamentally search techniques which search for efficient optimal solutions in the state space. By posing the problems in different domains as search problems the evolutionary computational techniques can be applied to almost any problem area. A novel application of evolutionary computational paradigm namely the genetic algorithm for exact graph matching/ graph isomorphism is described in this paper.

Genetic Algorithm was first introduced by John Holland in 1960's at University of Michigan [2] and its power was published to the world in 1975, through the seminal work of Holland, "*Adaptation in Natural and Artificial Systems*", [3]. The paradigm of genetic algorithms was made popular by David Goldberg who presented its various applications in his book, "Genetic Algorithms in Search, Optimization, and Machine Learning" in 1989 [4].

Given a problem to solve; the genetic algorithm finds an optimal solution by using a set of randomly generated solutions and evolving better solutions through iterations using operators which mimic the natural evolution process. The operators that imitate the natural evolution process are derived from the theory of evolution proposed by Charles Darwin (Survival of the fittest) [5].

Genetic algorithms also have been used in conjunction with graph theory to solve many problems. Genetic algorithms and graphs have been used for solving machine scheduling problem [6]. One of the most fundamental problems in graph theory is verifying graph isomorphism or graph matching. It has found applications in various domains. A brief description of the graph matching problem including the theoretical aspects and a brief survey of graph matching algorithms is presented in [7]. A comprehensive survey of graph matching algorithms applied to pattern recognition and image processing for the thirty year period preceding 2004 is found in [8,9].

A few representative approaches of evolutionary computing algorithms to graph matching are enumerated here. A framework for performing relational graph matching using genetic search on attributed graphs is described in [10]. There are three novel ingredients to the work. Firstly, the optimization process is cast into a Bayesian framework by exploiting the global consistency measure of Wilson and Hancock as a fitness measure. The second novel idea is to realize the crossover process at the level of sub graphs, rather than employing string-based or random crossover. Finally, convergence is accelerated by employing a deterministic hill-climbing process prior to selection. Different genetic algorithm solutions to the inexact graph matching variants have been discussed in [11]. Recently researchers from around the world have

used genetic algorithms/ evolutionary computing techniques for solving different types of graph matching problems [12-14]. In this paper a novel genetic algorithm (GA) based technique is proposed for solving the exact graph matching (graph isomorphism) problem using the graph node properties.

The methodology makes use of the vertex invariance, degree invariance and summation of shortest distance invariance for ascertaining the exact graph matching. Once the graphs are proved to be isomorphic, a novel genetic algorithm is employed for finding the vertex correspondence between the two isomorphic graphs. A new chromosome structure using real numbers for describing the various characteristics of the nodes is devised. The chromosome describes the proposed vertex correspondences between the two graphs. A new fitness function that computes the fitness of each chromosome (to be precise the optimality of vertex correspondence) is devised and employed. Further the crossover and mutation operators for the proposed chromosome representation are newly defined. Using the above representations and the operators defined, the optimal vertex correspondence for isomorphic/ similar graphs is found using a simple genetic algorithm [4]. The methodology is tested on a large number of pairs of synthetic graphs and the results are very encouraging as the accurate results are obtained in all (100%) of the cases.

The paper is organized into five sections. Section 2 gives the complete description of the two stage methodology devised to find the matching between two undirected unweighted graphs. Section 3 describes the genetic algorithm and its various components and elaborates how it is employed for finding vertex correspondence. The experimentation conducted and the analysis of the results is presented in section 4. Section 5 gives the conclusions and future directions.

Genetic Algorithm based Methodology for Graph Matching

The graph matching (graph isomorphism) problem is approached in two stages in the proposed methodology. First the two graphs are processed for finding whether they are isomorphic. The isomorphism is verified by employing the invariance in terms of the; The number of vertices; The degree invariance, (the vertices of same degree are equal in number in both the graphs); The sum of shortest distance invariance (The number and value of sum of shortest distances from a vertex to other vertices are the same in the two graphs). Using the above invariants the two graphs are verified for isomorphism. The two graphs are represented as adjacency matrices; the degree and shortest distance sum from each vertex to all other vertices are

computed and used for proving the similarity of graphs. If the two graphs are not isomorphic the methodology concludes that the two graphs are not similar. But if the two graphs are isomorphic then the newly devised genetic algorithm is employed for establishing the correspondence between the vertices of the two graphs.

For this purpose, the eccentricity of each node is computed and used along with the vertex degree and sum of shortest distances to other vertices for generating an initial population of chromosomes of the genetic algorithm (described in a subsequent section), which represents the initial estimates to the vertex correspondence between the two graphs. A newly defined fitness function estimates the fitness of each individual in terms of correspondence of vertices. This fitness function is based on the relation brought out in the hypothesis given below.

Once the initial population is generated then the next generation of the population is generated using the simple GA (generational GA). The newly defined selection, crossover and mutation operators are employed for generating the offspring's (new solutions) to the vertex correspondence problem. After iterating through the genetic algorithm, the fitness function is employed to select the most-fit individual as the proposed optimal correspondence between vertices of the graphs. The complete methodology is brought out in the two stage algorithm 1, given below.

Hypothesis

Given two isomorphic graphs G_1 and G_2 , if there is a relation R between vertices of G_1 to vertices of G_2 such that $R: G_1 \rightarrow G_2$ or $R: G_1(v_i) \rightarrow G_2(v_j)$ then if

$$Val = \frac{1}{\sum(\Delta_i \sigma_i e_i - \Delta_j \sigma_j e_j) + k} = \frac{1}{k}$$

Then the vertex i of G_1 corresponds to vertex j of G_2 .

Algorithm 1: Graph Matching GA (G_1, G_2)

Stage1

1. Read the edges of G_1, G_2 and construct $X(G_1), X(G_2)$.
2. Find the degrees of vertices, of both graphs and store in vectors, say D_1 and D_2 .

3. Find sum of shortest distances from a vertex to all the other vertices in both the graphs and store in Sp_1, Sp_2 .
4. Find the eccentricity of vertices of both graphs say E_1 and E_2 .
5. Check for invariance between (D_1, D_2) , and (Sp_1, Sp_2) respectively. If invariance does not hold, display the two graphs are not isomorphic and **stop**. Otherwise display the two graphs are isomorphic and go to step 6. (Stage 2: Genetic Algorithm)

Stage 2

6. Construct initial population of chromosomes representing various initial solutions of vertex correspondence.
7. Find the fitness of fittest individual after the termination of the iterations using simple GA
8. Display the vertex correspondence, using the fittest individual obtained from step 7.

9. Stop

The components of genetic algorithm are described in detail in section 3.

3.0 The Genetic Algorithm for Vertex Correspondence:

The genetic algorithm for correspondence match between vertices of the two graphs requires various components for its functioning viz; A chromosome representation, that gives a proposed solution to the problem. The crossover and mutation operators, that are reproduction operators, employed for producing off springs during the functioning of the GA. The selection operator helps in selection of parents for reproduction. The fitness function that evaluates the fitness of individual solutions and generally is the objective function that is to be optimized by the candidate solutions.

The genetic algorithm literature presents two basic types of genetic algorithms, namely the simple or generational GA and the second is the Steady State GA [4]. The steady state GA replaces the least fit individual from the population by the newly generated chromosomes if their fitness is better than the least fit individual's fitness in the population, whereas the simple GA generates a completely new generation of the population that replaces the older one. The different components of the genetic algorithm for graph matching are described in the following sub sections.

3.1 The Chromosome Representation

Generally the candidate solution for any problem that is solved using genetic algorithms is represented as a chromosome. The chromosome consists of different parts of the solution which are referred in literature as genes. For the problem of graph matching a new chromosome representation that employs the real values of the degree of the node, the summation of shortest distance to all the vertices from a given

vertex and the eccentricity of the vertex is employed. The chromosome contains n genes, where n is the number of vertices/ nodes of the graph. Every gene contains eight alleles that specify the possible correspondences and their characteristic values. The first four alleles stand for the vertex number, degree and summation of shortest distance to every other vertex and the eccentricity of the vertex from graph G_1 . The second set of four alleles stands for the same information from the second graph G_2 which is corresponding to the vertex in the first graph. The pictorial representation of the chromosome is given in the Figure 1.

$$\{V_1^1, D_1^1, Sp_1^1, E_1^1, V_2^1, D_2^1, Sp_2^1, E_2^1\}$$

$$\{V_1^2, D_1^2, Sp_1^2, E_1^2, V_2^2, D_2^2, Sp_2^2, E_2^2\}$$

$$\{V_1^3, \dots, \dots, \dots\} \dots\dots\dots$$

Figure 1: The Chromosome Representation

3.2 The Selection Operator: The literature has references to many selection operators such as elitism, round robin, greedy, stochastic and roulette wheel selection [4]. Each of the selection operators has its advantages. The roulette wheel selection operator which is one of the popular selection operators used in the genetic algorithms, is employed in this work. The simple GA also uses the elitism selection along with roulette wheel selection; it involves selecting the fittest individual from the population.

3.3 The Crossover Operator: The crossover operator is the reproduction operator involving two parents. The two parents are selected using the selection operator and are subjected to crossover operation based on a crossover probability. In this work the crossover probability is assumed to be 0.9. The crossover operation defined is a single point crossover [4]. The single point crossover involves randomly selecting crossover points in the two parents and then exchanging the two genes. If the gene so exchanged has alleles already present in the other genes of the chromosome, the alleles of such genes are replaced with alleles in the outgoing gene of this parent.

3.4 The Mutation Operator: The mutation operator is introduced to add variety to the population by randomly varying the genes of the chromosome. The mutation operator is applied with lesser mutation probability (A probability of 0.6 is used in this work). The mutation operator involves randomly selecting the gene for mutation. Once the gene for mutation is selected the first vertex (first allele) is modified by randomly selecting the vertex and finding the corresponding values for the next three alleles. If the newly introduced vertex is present in any other gene of the chromosome then it is replaced by the alleles of the mutated vertex.

3.5 The Fitness Function: The fitness function is a very important component of the genetic algorithm. It decides on the fitness of the individual chromosomes which are the probable solutions to the problem. The fitness function is also used as an important condition for terminating the GA. In most of the cases the fitness function is the objective function of combinatorial optimization. In the present problem of graph matching the fitness of the proposed solution/ chromosome is obtained by the summation of difference between the products of the three parameters of the corresponding vertices in the chromosome. More particularly the difference between the products of the degree, summation of shortest distances to all other vertices in the graph and the eccentricity of the corresponding vertices in a gene is considered and a small value delta is added to this difference (empirically taken as 0.01). The reciprocal of this term is the fitness value of the chromosome and is based on the hypothesis described above). The fitness function is depicted in equation 1.

$$f = \frac{1}{\left(\sum_{i=1}^n (D_{v1g1}^i SP_{v1g1}^i ECC_{v1g1}^i - D_{v2g2}^i SP_{v2g2}^i ECC_{v2g2}^i)\right) + \delta}$$

Where δ is 0.01

...(1)

This fitness function returns a value of 100 when the chromosome gives the required solution. The genetic algorithm employed for vertex correspondence matching is described in ensuing sub section.

3.6 The Simple or Generational GA

This technique of the genetic algorithm involves generation of the complete population in a single iteration. The simple GA is historically found to be a very efficient search technique and has been employed for finding solutions to various problems. The simple or generational genetic algorithm to search for vertex correspondence is given as algorithm 2. The algorithm assumes maximum iteration of 15000.

Algorithm 2: Simple or Generational Genetic Algorithm

Input: The adjacency matrices of the graphs G_1 and G_2 , the degrees of all the vertices of G_1 and G_2 (D_1, D_2), the summation of shortest distances from a vertex to all other vertices in G_1 and G_2 (SP_1, SP_2), the eccentricities of all the vertices in G_1 and G_2 (E_1, E_2)

Output: The correspondence of vertices in the graphs G_1 and G_2 .

Step 1: Start, Find the quantity of the initial population. $Init = \text{Fact}(n)/5$ if $n \leq 5$, $Init = \text{Fact}(n)/20$ if $n > 5$ and $n \leq 7$, $Init = 250$ otherwise.

Step 2: Generate the initial random population of size equal to $Init$, let it be CR

Step 3: Find the fitness of all the individuals

Step 4: $iter = 0$;

Step 5: While $iter \leq 15000$ do

Step a: Find the individual with maximum fitness. Let the maximum fitness be M .

Step b: If M is greater than or equal to predefined optimal fitness of 100, break out of the loop.

Step c: $K = 0$

Step d: While ($K < Init$) do

Step i: Copy the fittest individual to the new generation being created, CRN by incrementing the index K . If $K > Init$ Break out of loop

Step ii: Select a parent using roulette wheel selection, say P_1 and P_2 of two graphs

Step iii: Perform cross over operation and generate two offsprings, say CH_1, CH_2 using crossover probability

Step iv: Perform Mutation operation on the offsprings CH_1 and CH_2 using mutation probability

Step v: ADD CH_1 and CH_2 to the new population CRN and correspondingly increment K . If $K > Init$ Break out of the loop

Step e: Copy CRN to CR

Step f: Find fitness of all the individuals of CR

Step 7: Select the chromosome with maximum fitness, and display the correspondence of the vertices between graphs G_1 and G_2 , and STOP

The simple GA has been implemented and tested on a large number of synthetic graph pairs and the results of vertex correspondences are satisfactory. The test results are described in the ensuing section.

4.0 Experimentation and Analysis: The proposed method of graph matching using the invariants such as degree of vertex, shortest distance sum from a vertex to other vertices for checking graph similarity and further a new genetic algorithm for finding node correspondence has been implemented using MATLAB. The methodology has been thoroughly tested using a large number of synthetic graph pairs. The results of the methodology including the number of iterations used up by the GA and the time taken for a typical graph pairs of given number of nodes is brought out in Figure 2.

The time complexity of the genetic algorithm depends on population size and the number of generations. The overall results of graph matching (for isomorphic graph pairs), using the genetic algorithm is 100%. Out of the 39 graph pairs tested each one of them were correctly identified and the nodes were accurately mapped.

Conclusions: In this paper a novel method of graph matching as a two stage process is proposed. In the first stage the graphs are checked for similarity and if they are similar, vertex correspondences are obtained using the newly devised genetic algorithm. Genetic algorithm has been previously employed for bipartite matching applications but not for exact graph matching (graph isomorphism) of simple undirected

graphs. This paper has introduced a novel method for the purpose.

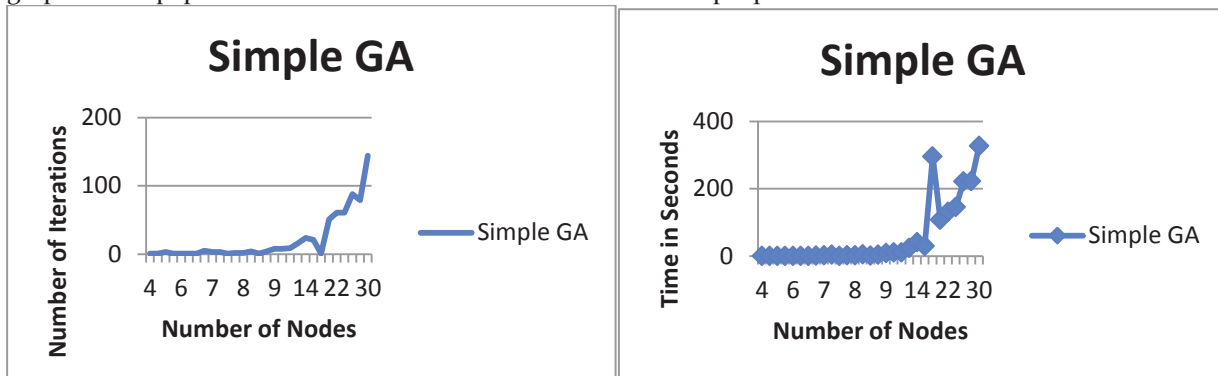


Figure 2: The Plot of Number of Iterations and time in seconds V/s Nodes for Simple GA

The methodology, has made use of the characteristics of the vertices (nodes) of the graph and a new fitness function that finds the optimal vertex correspondences by computing the difference of the products of the corresponding node characteristics namely degrees, shortest distance sum and eccentricity. The methodology proposed in this paper is found to be robust and has performed very well on all types of graph pairs. Some of the typical graphs conventionally employed in graph isomorphism testing (counter examples) have been used and the results are as expected. The results are very encouraging and can be further employed for various applications and can be explored for applicability to other types of graphs.

References:

- Jain Brijnesh J., Fritz Wyszotzki, 2005 "Solving inexact graph isomorphism problems using neural networks", *Neurocomputing*, pp. 45-67
- Mitchell Melanie, 1998, "An Introduction to Genetic Algorithms", MIT Press, ISBN 0-262-13316-4, (HB), 0-262-63185-7 (PB)
- John H. Holland, 1992 "Genetic Algorithms", *Scientific American*.
- Goldberg David E., 1989 "Genetic Algorithms in Search, Optimization, and Machine Learning", ISBN-10: 0201157675
- W15: http://en.wikipedia.org/wiki/Charles_Darwin
- Nirmala G, Mr.D.Ramprasad, 2012, "Application of Genetic Algorithm in Graph Theory", *International Journal of Scientific and Research Publications*, vol. 2, Issue 7, July 2012, ISSN 2250-3153, pp. 1-4
- Bunke Horst, 2000 "Graph Matching: Theoretical Foundations, Algorithms and Applications", In *Vision Interface*, pp. 82-88
- Conte D., P. Foggia, C. Sansone and M. Vento 2004 "Thirty Years of Graph Matching in Pattern Recognition" *International Journal of Pattern Recognition and Artificial Intelligence* vol. 18, No. 3 pp.265-298 @ World Scientific Publishing Company
- Conte D., P. Foggia, C. Sansone and M. Vento 2003, "Graph Matching Applications in Pattern Recognition and Image Processing", *Proceedings of International Conference on Image processing (ICIP)*, pp. II 21-4
- Cross Andrew D J; Wilson Richard C, Hancock Edwin R, 1997, "Inexact Graph Matching Using Genetic Search", *Pattern Recognition* vol. 30, no. 6, pp 953-970
- Rekha V. Joshi, Multiplication Semigroups With Identity; *Mathematical Sciences International Research Journal* ISSN 2278 - 8697 Vol 2 Issue 2 (2013), Pg 302-307
- Ravadanegh Sajad Najafi, 2008 "A New GA - Based and Graph Theory Supported Distribution System Planning", *ICIC 2008, CCIS 15*, pp. 7-14
- Angadi Shanumukhappa A, Sanjeevakumar M Hatture, 2011, "A Novel Spectral Graph Theoretic Approach to user Identification using Hand Geometry", *International Journal of Machine Intelligence*, ISSN: 0975-2927 & E-ISSN: 0975-9166, vol. 3, Issue 4, pp-282-288
- Barecke T. 2007, "Memetic algorithms for inexact graph matching", *Evolutionary Computation*, IEEE pp. 4238 - 4245
- Le Dang Nguyen, Dac-Nhuong Le, Tran Thi Huoang and Le Trong Vinh, 2014, "A New Genetic Algorithm Applied to Inexact Graph Matching", *International Journal of Computer Science and Telecommunications*, Vol. 5, Issue 5, May 2014

Vidya M Shettar/Department of Mathematics/Basaveshwar Engg. College/Bagalkot.
Shanmukhappa A Angadi/Department of CSE/Visvesvaraya Tech. University/ Belagavi