ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2013

Problem Solving of graph correspondence using Genetics Algorithm and ACO Algorithm

Alireza Rezaee,[1], Azizeh Ajalli [2]

Assistant professor ,Department of Mechatronics Engineering, Faculty of New Sciences and Technologies, University of Tehran, , P.O.Box 143951374,Tehran, Iran[1]

Member of Young Researchers Club, Abhar Branch, Islamic Azad University, Abhar, Iran[2]

Abstract: In this paper, new genetics and Ant colony optimization algorithm for solving the problem of graph correspondence is presented. When using the genetics technique for the problem of graph correspondence, it is not easy to define the crossover operator. our attempt will be to present a definition holding the integration of the population graph in a one-to-one correspondence. we present new and suitable definitions for the target function and a function giving score to a solution at the end of any cycle. We compare both algorithms and try to find their advantages and their shortcomings.

Keywords: Graph Correspondence, Genetics Algorithm, AOC Algorithm

I. INTRODUCTION

The problem of graph correspondence has been noticed in a wide range of various fields and has become an important instrument for modeling and solving different problems. This problem is a NP-Complete one and the time required for solving it increases exponentially in terms of the number of the input graphs. Recently, many researches are conducted for improvement of the efficiency of graph correspondence algorithms, both in time and in memory consumption. Some of these algorithms reduce temporal complexities through applying limitations over the structure of the input graphs. Another method for reducing the complexity of the process of correspondence is to present a special display for seeking and deleting useless routes of the search space. In these methods, no limitation is applied on the structure of the input graphs. So they can be used in a larger domain of applications. One of the first articles in this context [1] suggests an algorithm applying some distortion on the input graphs and builds an intermediate display on which the action of correspondence is done much rapider. But [2] showed that the assumptions used by this method are not always true and this limits the domain of using this algorithm.

Another method reducing the search space considerably is Ullmann algorithm that uses backwards technique [3] and is still widely used. Messmer in [4] compared this algorithm with other algorithms and showed its high efficiency in a one-to-one correspondence.

Another algorithm using backwards techniques is Schmidt& Druffel algorithm stated in [5]. This algorithm uses the information existing in the distance matrix. Another important algorithm in graphical correspondence is Mckay's Nauty algorithm [6]. This algorithm is based on a series of transformations. These transformations transform the graphs to a focused form and correspondence test is much rapider in this focused form.

Another algorithm for the problem of graphical correspondence is presented in [7]. In this algorithm, instead of trying to reduce temporal complexity of the algorithm, it is tried to reduce the complexity in a case there is a constant model graph and the graphs are constantly passed in front of this model graph and the samples of the model graph are extracted from any target graphs. Since the model graph is deemed constant in this algorithm, so a table of information related to the model graph and the various states of having it in the target graph can be built in a preprocessing manner and it can be used in the process of correspondence.

Using techniques inspired from the nature in the problem of graph correspondence is examined in several articles. [8] suggested a method for solving the problem of graph correspondence using the genetics algorithm. In his suggestion,

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2013

the suggested crossover function may not work correctly for a one-to-one correspondence. The example he used for testing the algorithm is the examination of the resemblance between the solar system and an atom. In his work, no comparison between the suggested algorithm and other methods has been done. In [9], using an ACO technique, a method for solving the problem of graph correspondence is suggested. In this article, definitions of the target function and also a function giving scores to a solution found at the end of any cycle are stated. [10] has also proposed an algorithm using ACO.

At first, we try to improve genetics and ACO algorithms suggested for the problem of graph correspondence. For genetics algorithm, we present a new definition of crossover operator. [11]For ACO algorithm, we try to present exact definitions fitting the problem of graph correspondence for the target function and a function giving scores to a solution at the end of any cycle. Finally, we do a comparison between the two proposed methods. [12] The structure of this article is as follows: first, we propose the genetics algorithm suggested for solving the problem of correspondence. [13,14]Then, we describe the proposed method using ACO technique. In the next section, the results are examined and both algorithms are compared. Finally, conclusions are derived from the works. [15,16]

II. GENETICS ALGORITHM

Genetics algorithm is an optimizing algorithm based on using evolutionary methods. The main idea of this algorithm is that instead of a solution, a population of solutions is held for a problem of optimization and doing operations over this population, the new population is generated. Any solution is composed of a set of genes and any gene shows one or more properties fitting that solution. Genes can be of any type of data, but they are often considered as binary ones. Any solution forms a vector of binary values although except the vector, other data structures can be used (as it is done for solving the problem of graph correspondence and instead of using a vector, we use a two dimensional matrix). Any member of this population can pair based on the closeness to a target function and produce some offspring. These offspring can be a part of the new population of the solutions. After all reproductions are performed, individuals (solutions) being more beneficial to the target function than other solutions are returned as the final solution of the problem. [17]

Genetics algorithm has good capabilities in solving problems of discrete optimization. Naturally, genetics algorithms are suitable for solving those problems having many optimal points and in addition, having this property that sub-solutions can be locally combined and a better general solution is obtained. This is done using the crossover function. [18]

In the problem of graph correspondence, a correspondence function is defined identifying the relation between nodes of the target graph and those of the model graph. This correspondence function is shown as a two dimensional binary matrix M when used in Genetics algorithm. Any row of this matrix shows a node of the target graph (t_i) and any of its columns shows a node of the model graph (b_j). If the element M_{ij} of this matrix is one, then $f(t_i)$ will be equal to b_j . Only an element of a row can have a non-zero value and other elements must be zero, because any node of the target graph can only correspond to a node of the model graph.

Now we need to define mutation and crossover operators in this representation of the problem of graph correspondence. These definitions must perform such that the integration of the matrix holds, i.e. after applying them; any node of the target graph only corresponds to a node of the model graph. For mutation, we use the shift of two rows of the matrix, i.e. in mutation of a solution (a matrix), we replace nodes of the model corresponded to two target nodes. [19]

Definition of crossover function is not as easy as that of mutation. Our definition from crossover function is that first, we determine a position between one to the number of the nodes of the target graph (the number of rows of the matrix). This position divides any of the two matrixes and also the matrix must be divided to two parts. The new matrix takes any parts from one of the two available matrixes. A problem we face is that the integration of the resulting matrix may not hold and nodes from the target graph correspond to some nodes from the model graph. In this case, the crossover function replaces some non-zero values of the column having more than one non-zero value with zero values of the columns not having non-zero values. [20]

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2013

III. ACO ALGORITHM

One of the other algorithms inspired from the nature we have selected for solving the problem of graphical correspondence is Ant Colony Optimization (ACO) algorithm. [21] In any cycle of this algorithm, any ant forms a complete correspondence and the effect of pheromone emerges. This algorithm uses a construction graph in solving the problem of graph correspondence. The construction graph is a graph the artificial ants leave their pheromone traces on its sides. The vertexes of this graph are various parts of the solution that every vertex can be selected by an ant and added to the solution constructed by then (vertexes and sides selected by then). [22] Suppose two graphs their resemblance must be determined are as $G=(V,r_V,r_E)$ and $G'=(V',r'_V,r'_E)$. A construction graph is a non-directed complete graph in which there is a vertex for any $(u,u')\in V\times V'$.

a. Pheromone

As we know, pheromone is the means of communication among ants leaved on the edges of construction graph by them. The amount of pheromone on an edge $<(u,u'),(v,v')>$ is denoted by $T(u,u'),(v,v')$ and shows the utility of the correspondence u with u' and v with v'. When selecting a new vertex by an ant, a side lead to the selection of that vertex takes a value that shows the content of the pheromone leaved by the ant on that side. When we construct a correspondence graph, those vertexes corresponded to each other in the correspondence graph are also more likely to correspond to each other in the new correspondence graph, of course if the new correspondence graph has taken vertexes from the previous correspondence graph. In other words, the more correspondent vertexes the new correspondence graph has, the higher the utility and the probability of the selection of the remaining vertexes of the previous correspondence graph that correspond to each other but have not appeared in the new correspondence graph.

b. Construction of a correspondence by an ant

In any cycle, any ant constructs a correspondence and this correspondence states that which vertexes correspond to which vertexes of the second graph.[23] Any ant starts from a null correspondence $m=\Phi$ and adds pairs iteratively to it. These pairs are selected from the set $\{(u,u')\in V\times V'-m\}$, where m is the set of the pairs selected before. The probability of selecting the pair $(u,u')\in cand$ is as follows:

$$\frac{[T_m(u,u')]^{\alpha}.[h_m(u,u')]^{\beta}}{\sum_{(v,v')\in cand}[T_m(v,v')]^{\alpha}.[h_m(v,v')]^{\beta}} \qquad (1)$$

Where:

- $T_m(u,u')$ is the pheromone factor and is equal to the set of all traces of pheromone that are between the new candidate (u,u') and any pair (v,v') selected before and added to m, i.e:

  $$T_m(u,u')=\sum_{(v,v')\in m}T<(u,u'),(v,v')> \qquad (2)$$

  If m is equal to 0, i.e. when we want to select the first pair, we ignore the effect of pheromone factor. For this, we put $T_m(u,u')$ equal to 1. In this state, the probability is only related to the heuristic factor.

- $h_m(u,u')$ is the heuristic factor that causes those pairs to be selected that increase the score function more.

- α and β are two parameters determining the importance of any above parameters.

Ants stop selection and adding new pairs to the correspondence m when no pair increases the score function directly or when the score function is not increased in the three last iterations. [24,25]

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2013

c. The stage of updating pheromone

For simulating vaporization of pheromone, any trace of pheromone $T<(u,u'),(v,v')>$ is multiplied by (1- ρ), where ρ is the rate of vaporization and is in the range of 0 and 1.

The best ant of the cycle leaves its pheromone. Suppose m_k is the best correspondence compared to the score function that is constructed during this cycle. If there are some correspondences with identical scores, one of them is randomly selected. Suppose m_k is the correspondence of the current cycle and m_{best} is the best correspondence constructed from the beginning of the program to now (consider also the current cycle). The content of the pheromone leaved has an inverse relation with the difference between the score m_k and the score m_{best} , i.e. it is equal to: $1/(1+score(m_{best})-score(m_k))$

This content of pheromone is leaved on any edge $((u,u'),(v,v'))$ linking two different pairs (u,u') and (v,v') of m_k .

IV. RESULTS

ACO algorithm highly depends on its parameters and among the parameters, pheromone factor (α) and the factor of the rate of vaporization (ρ) have the highest effects.

In this algorithm, when α or ρ increase, the ants reach a correspondence more quickly although this decreases the score of correspondence. For example, consider the following figures:

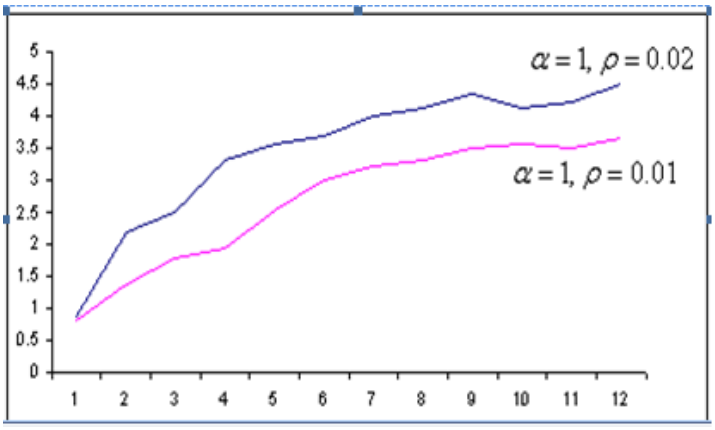


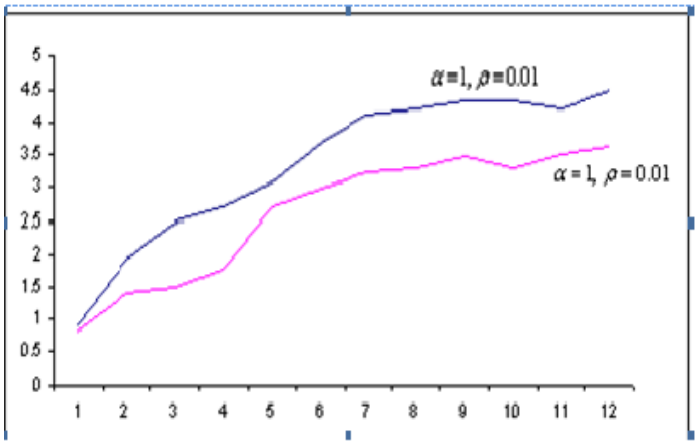
Figure 1: we have changed the values of ρ , where α is constant. The result is that increasing ρ , correspondence is obtained more quickly, but with a lowers score.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

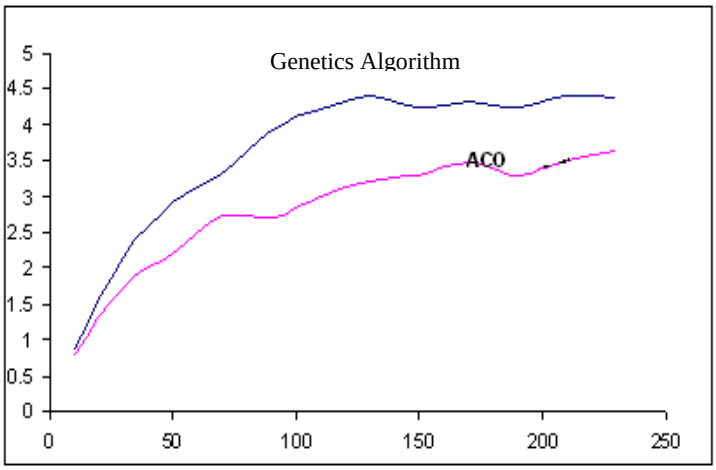
Vol. 2, Issue 12, December 2013



The number of cycles

Figure 2: we have changes the values of α , where ρ is constant. The result is that increasing α , the correspondence is obtained more quickly, but with a lower score.

In comparison between genetics algorithm and ACO algorithm, the output results highly depend on the parameters of ACO algorithm. For example, consider that ρ =0.1 and α =1. In these conditions, ACO algorithm finds the answer much quicker than genetics algorithm. The following figure shows this result.



The number of nodes

Figure 3: Given ρ =0.1 and α =1, ACO algorithm works quicker than genetics algorithm

On the other hand, for example suppose that ρ =0.05 and α =0.5. In these conditions, Genetics algorithm works quicker than ACO algorithm. The following figure shows this result.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2013

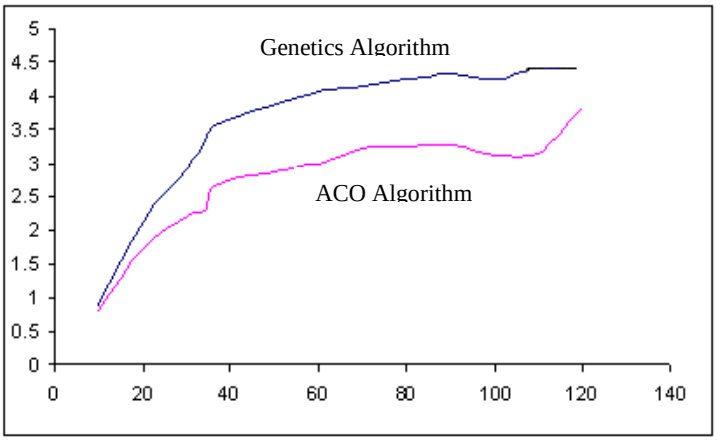


Figure 4: given ρ =0.05 and α =0.5, genetics algorithm works quicker than ACO algorithm.

Another problem is the number of the nodes existing in the graph. If the size of the graphs increase (in our experiments, more than 150 nodes), then regardless of the parameters of ACO algorithm, genetics algorithm works better than ACO algorithm. For example, consider the above case (ρ =0.05 and α =0.5). in this case, when the number of nodes is low, ACO algorithm works quicker (as it is shown in figure 4), but when the number of nodes is high, genetics algorithm finds the answer quicker.

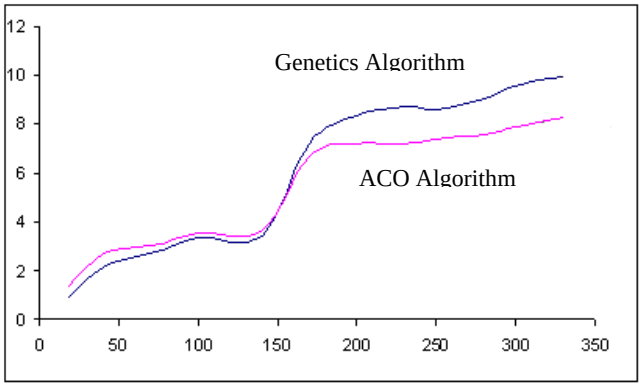


Figure 5: Efficiency of both algorithms with increasing the number of nodes.

The vertical axis shows the operation time. First, ACO algorithm works better, but gradually increasing the number of nodes (more than 150 nodes), the operation time of genetics algorithm becomes less than that of ACO algorithm.

V. CONCLUSIONS

In this paper, we first presented a genetics algorithm and an ACO algorithm for the problem of graph correspondence. We stated a new definition from crossover operator in genetics algorithm that holds the integration of the population graph. In ACO algorithm, we presented new definitions for the target function and a function giving scores to a solution at the end of any cycle.

In the next step, we compared both algorithms presented. ACO algorithm is very sensitive to the parameters ρ and

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 12, December 2013

α , and its operation time and also the quality of the solution it finds (the score it gains) depends on these parameters. This relation is presented in the section, empirical results and the comparison of both algorithms, and also in figures 1, 2, 3 and 4.

If the size of these graphs increases (for example more than 150 nodes), regardless of the parameters of ACO algorithm, genetics algorithm works better than ACO algorithm (it obtains correspondence in less time).

REFERENCES

[1] D.G. Corneil, C.C. Gotlieb, An efficient algorithm for graph isomorphism, *Journal of the Association for Computing Machinery*, 17, pp. 51-64, 1970.
[2] R. Mathon, Sample graphs for isomorphism testing, *Congressus Numerantium, 21*, pp. 499-517, 1978.
[3] M. Dorigo and G. Di Caro. The Ant Colony Optimization Meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization. McGraw Hill, London, UK, pages 11-32, 1999.
[4] B. T. Messmer, *Efficient Graph Matching Algorithms for Preprocessed Model Graphs*, Ph.D. Thesis, Inst. of Comp. Science and Appl. Mathematics, University of Bern, 1996.
[5] T. St¨utzle and H.H. Hoos. MAX − MIN Ant System. Journal of Future Generation Computer Systems, 16:889-914,2000.
[6] B.D. McKay, Practical Graph Isomorphism, *Congressus Numerantium, 30*, pp. 45-87, 1981.
[7] H. Bunke, B.T. Messmer, Efficient Attributed Graph Matching and its Application to Image Analysis, in *Image Analysis and Processing*. (C. Braccini, L. De Floriani, G. Vernazza, eds), Springer, Berlin Heidelberg, pp. 45-55, 1995.
[8] Tessem, Bjørnar, Genetic Algorithms for Analogical Mapping. In Proc. of Intl. Conf. on Evolutionary Computation (ICEC `98), pp. 762-677, Anchorage, Alaska. IEEE. 1998.
[9] Olfa Sammoud, Sébastien Sorlin, Christine Solnon, and Khaled Ghédira. Ant Algorithm for the Graph Matching Problem. 5th European Conference on Evolutionary Computation in Combinatorial Optimization. April 2005.
[10] Olfa Sammoud, Sébastien Sorlin, Christine Solnon, and Khaled Ghédira. A omparative Study of Ant Colony Optimization and Reactive Search for Graph Matching Problems. 6th European Conference on Evolutionary Computation in Combinatorial Optimization. 2006.
[11] H. Kälviäinen E. Oja, Comparisons of Attributed Graph Matching Algorithms for Computer Vision, Lappenraantra University of Technology, Department of Information Technology, Research Report 18, 1990.
[12] T. Miyazaki, The complexity of McKay's canonical labeling algorithm, in *Groups and Computation, II* (L. Finkelstein and W.M. Kantor, eds.), Amer. Math. Soc., Providence, RI, pp. 239-256, 1997.
[13] X. Jiang, H. Bunke, Including geometry in graph representations: a quadratic time isomorphism algorithm and its applications, in Perner, P., Wang, P., Rosenfeld, A. (eds.): Advances in Structural and Syntactic Pattern Recognition, Springer Verlag, Lecture Notes in Computer Science 1121, 1996, 110 - 119.
[14] S. Sorlin and C. Solnon. Reactive Tabu Search for Measuring Graph Similarity. to appear in 5th IAPR Workshop on Graph-based Representations in Pattern Recognition (GbR 2005), LNCS, Springer Verlag, 2005.
[15] R. Battiti and M. Protasi. Reactive Local Search for the Maximum Clique Problem. Algorithmica, Springer-Verlag, (29), 610-637, 2001.
[16] P. Foggia, C. Sansone, M.Vento, A Database of Graphs for Isomorphism and Sub-Graph Isomorphism Benchmarking, *Proc. of the 3rd IAPR TC-15 International Workshop on Graph-based Representations*, Italy, 2001.
[17] H. Bunke, M.Vento, Benchmarking of Graph Matching Algorithms Proc. 2nd IAPRTC15 Workshop on Graph-based Representations, Handhorf, 1999.
[18] L.P. Cordella, P. Foggia, C. Sansone, M. Vento, Evaluating Performance of the VF Graph Matching Algorithm, *Proc. of the 10th International Conference on Image Analysis and Processing*, IEEE Computer Society Press, pp. 1172-1177, 1999.
[19] P. Foggia C. Sansone, M.Vento, An Improved Algorithm for Matching Large Graphs, *Proc. of the 3rd IAPR-TC-15 International Workshop on Graph-based Representations*, Italy, 2001.
[20] J.R. Ullmann, An Algorithm for Subgraph Isomorphism, *Journal of the Association for Computing Machinery*, vol. 23, pp. 31-42, 1976.
[21] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. International Journal of Pattern Recognition and Artificial Intelligence, 18(3):265-298, 2004.
[22] P. Champin and C. Solnon. Measuring the similarity of labeled graphs. 5th International Conference on Case-Based Reasoning (ICCBR). Lecture Notes in Computer Science - Springer Verlag, 2003.
[23] H. Bunke and X. Jiang. Graph matching and similarity. Volume Teodorescu, H-N, Mlynek, D.Kandel, A. Zimmermann, H-J. (ds.): Intelligent Systems and Interfaces, chapter 1, 2000.
[24] M. Boeres, C. Ribeiro, and I. Bloch. A Randomized Heuristic for Scene Recognition by Graph Matching. WEA 2004, 100-113, 2004.
[25] R. Ambauen, S. Fischer, and H. Bunke. Graph Edit Distance with Node Splitting and Merging, and Its Application to Diatom Identification. IAPR-TC15 Wksp on Graph-based Representation in Pattern Recognition, LNCS, Springer Verlag, 95-106, 2003.