Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

On the Minimum of True Matches in Exact Graph Matching with Simulated Annealing

Hashem Ezzati*[1], Mahmood Amintoosi†[2] and Hashem Tabasi‡[3]

[1]Faculty of Mathematics and Computer science, Amirkabir University of Technology
[2]Faculty of Mathematics and Computer science, Hakim Sabzevari University, Sabzevar, Iran
[3]Faculty of Mathematics and Computer science, Damghan University, Damghan, Iran

ABSTRACT

Graph matching is one of the most important problems in graph theory and combinatorial optimization, with many applications in various domains. Although meta-heuristic algorithms have had good performance on many NP-Hard and NP-Complete problems, but for graph matching problem, there were not reported superior solutions by these sort of algorithms. The reason of this inefficiency has not been investigated yet. In this paper it has been shown that Simulated Annealing (SA) as an instance of a meta-heuristic method is unlikely to be even close to the optimal solution for this problem. Mathematical and experimental results showed that the chance to reach to a partial solution, is very low, even for small number of true matches. In addition

Keyword: Graph matching, Simulated Annealing, Meta-Heuristic, Stochastic Optimization

AMS subject Classification: 05C60, 05C70

ARTICLE INFO

Article history:
Research paper
Received 14, October 2020
Received in revised form 17, April 2021
Accepted 11 May 2021
Available online 01, June 2021

*h.ezzati@aut.ac.ir
†Corresponding author: M. Amintoosi. Email: m.amintoosi@hsu.ac.ir
‡tabasi@du.ac.ir

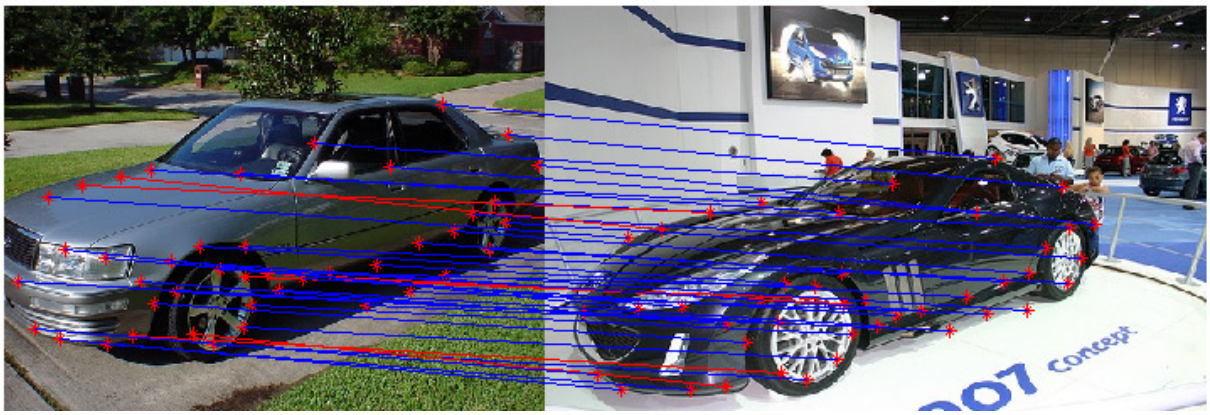

Figure 1: Two car images' key points are considered as graphs and a partial solution of graph matching is demonstrated by blue/red lines; incorrect matches are shown with red lines.

1 Abstract continued

to theoretical discussion, the experimental results also verified our idea; for example, in two sample graphs with 10000 vertices, the probability of reaching to a solution with at least three correct matches is about 0.02 with simulated annealing.

2 Introduction

Graph matching, which is the problem of finding a similarity between graphs [3] is one of the oldest problems in Graph theory and combinatorial optimization. The case of exact graph matching is known as the graph isomorphism problem [3] which has various applications such as image matching in computer vision [16, 30]. Several important applications in computer vision, such as 3D reconstruction from stereo images, object matching, object category and action recognition [25] require the ability to efficiently match features of two sets. Image features are considered as graph vertices for modeling the matching problem as a graph matching problem.

2D Image matching problem is an example of exact matching problem. Figure 1 shows a pair of car images along with their corresponding feature (key) points. The number of key points in two images are identical. The set of key points in each image indicate the vertices of a graph. The graph edges were not shown here. Graph matching problem for these images, means finding the corresponding points of two images. With a perfect solution, each point in the left image should be related to its correspondence point in the right image. Figure 1 represents a partial solution, in which the true/false matches are demonstrated by blue/red lines. For this problem, there exisits only one exact solution, hence the problem is an exact graph matching problem.

Graph matching is an NP-hard problem [30] with complexity of $O(n!)$ in its general form. Consider two graphs with n vertices in each graph, the number of all possible matching

cases between them is $n!$. Each permutation of numbers $\{1, \ldots, n\}$ shows a point in the solution space. In many applications such as figure 1 only just one permutation among $n!$ possible solutions is acceptable. Meta-heuristic algorithms which are mainly based on random search are used on many NP-Hard and NP-Complete problems successfully. For example, they are efficient on many famous problems such as: traveling salesman problem, N-Queen, facility location, graph coloring, max-cut problem, bin-packing and time tabling [24, 26, 12, 21, 14]; but the reported works on graph matching based on meta-heuristic algorithms which handle the problem efficiently, are rare.
We divided the published related works to journal and non-journal papers:

Journal Papers: Egozi et. al. [8] reported an spectral approach for graph matching. Qiao et. al. [19] also used spectral graph matching in remote sensing. Finch et.al. [9] and Luo and Hancock [18] reported expectation maximization and singular value decomposition methods for dealing the problem. Almohamad and Duffuaa [1] used linear programming. Gradient descent approach was used in [28]. Shapiro and Brady [23] and Umeyama [27] solve the matching problem by eigen decomposition. Brown et.al. [4], Auwatanamongkol [2], Cross et. al. [7, 6] and Fröhlich and Košir [10] used genetic algorithm (GA).

Non-Journal Papers: Leordeanu and Hebert [17] reported spectral method. Using simulated annealing (SA) [11], ant colony optimization [22, 29] and tabu search (TS) [16] also reported by the researchers. In [5, 13] also GA has been used for the problem.

As can be seen, meta-heuristic approaches – except GA – were only reported in non-journal published papers. In this paper we will discuss why this sort of approaches are not efficient to solve the graph matching problem.
Some meta-heuristic approaches like SA and TS are based on defining a neighboring structure and move from one point in solution space to another point. The published journal papers about graph matching show that stochastic optimization methods like SA are not attractiveness for researchers. This fact and the authors failure to solve the matching problem by SA, makes this intuitive result that methods like as SA are not effective for matching problem. GA in contrast to SA, which only moves to neighborhood point, uses crossover and mutation on a population of state space points, hence here the performance of this algorithm is not investigated.
Simulated annealing is a kind of an stochastic optimization methods; these methods start from a random point in the solution space and produce neighbors by applying random operators. In this study, we will show that why SA is not suitable for graph matching, by discussing that the probability of reaching the optimal solution (success rate) is close to zero, with SA. The experimental results verifies theoretical discussion.
A main reason that SA is not successful for handling graph matching is due to the nature of exact graph matching. In some problems , there are many optimal points in the solution space, which makes finding the solution, easy. Finding the optimal point, takes more time, when the optimal points in solution space are rare. N-Queen problem [15] is a problem

Table 1: Number of solutions of N-Queen problem on various number of Queens (N) [15].

N	Number of Solutions
1	1
2	0
3	0
4	2
5	10
6	4
7	40
8	92
9	352
10	724
11	2,680
12	14,200
13	73,717
14	365,596
15	2,279,184
16	14,772,512
17	95,815,104
18	666,090,624

with so many correct solutions. For example there are about 100 millions correct answers for $n = 17$. As can be seen in table 1, the number of solutions rapidly grows, when n increases. But in some problems such as exact graph matching, only one correct solution exist. Our goal is to prove that the ratio of partially good solutions decreases, when the number of graph vertices increases, for graph matching. The goodness of the solution is related to the number of true matches in the solution.
The rest of the paper organized as follows: In section 3 some basic definitions are mentioned. Section 4 devoted to theoretical aspects of an upper bound on success rate in SA. Section 5, shows experimental results on various graphs for inspecting the theoretical results and the last section is about to conclusion remarks.

3 Basic Definitions

Definition 3.1 (Graph Matching)**.** Consider two graphs G_1 and G_2, a one-to-one mapping fromG_1 to G_2 represents a matching between the two graphs [24].

Definition 3.2 (Isomorphism)**.** Consider two graphs G_1 and G_2, they are isomorphic if and only if there is a mapping $\phi : V(G_1) \longrightarrow V(G_2)$ as each vertex of G_1 exactly corresponds to a vertex of G_2 (Any two vertices x and y of G_1 are adjacent in G_1 if and only if $\phi(x)$ and $\phi(y)$ are adjacent in G_2)[24].

Simulated Annealing Algorithm: SA algorithm starts from a random point, and picks a random move to go to a new point in the neighbor of the previous point. If the new

point improves the solution (decreases energy), then it is always accepted. Otherwise, it is accepted with some probability. This probability decreases exponentially over time and with the badness of the move.

Feasible Solution: A feasible solution or a point in the solution space is a permutation of $\{1, \ldots, n\}$, in matching problem.

True Solution: In this paper it is supposed that the true solution is numbers $\{1, \ldots, n\}$ in ascending order.

Energy (Error): The number of elements that are not in their true positions, is the solution energy (error), which should be minimized. This measure is normalized by dividing by n. For example suppose that the true solution of a matching problem be {1,2,3,4,5} and a random solution be {3,2,1,4,5}. In this permutation $1, 3$ are not in their right places, hence its corresponding error is 2/5. In stochastic optimization methods like SA, in each run, the starting point is selected randomly; the mean error is computed by averaging the errors over multiple runs.

Precision: Precision is 1 minus error, which is equal to the rate of correct matches.

Derangement: a derangement is a permutation of the elements of a set, such that no element appears in its original position. The number of derangements of a set of size n, usually written D_n, d_n, or $!n$, is called the "derangement number".

Generally the rate of correct matches is reported as the precision of graph matching algorithm. In the following section we will show that with SA for exact graph matching, the precision, approaches zero, when the graph size increases.

4 Theoretical Discussion

Each feasible solution is a random permutation of {1, ..., n}, in stochastic optimization approaches like SA, the new solution is produced by perturbing the current solution. The common methods for permutation perturbation are swapping, insertion, inversion and scramble operators. All of these operators make a new vector (solution) from the current vector by altering some part of the vector. In fact, moving to the new point is just moving to a new random point, around the previous point. Here we discuss the probability of "goodness" of a random point.
In a solution space of permutations vectors, there are $n!$ points. Remmel [20] showed that the number of permutations of $\{1, \ldots, n\}$, that no number is located in its true position (known as derangement) is:

$$\begin{aligned} &D(n) = (n-1)(D(n-1) + D(n-2)), n \geq 1 \\ &D(0) = 1, D(1) = 0 \end{aligned} \tag{1}$$

which is equivalence to following non-recursive summation [20]:

$$D(n) = n! \sum_{k=0}^{n} \frac{-1^k}{k!} \tag{2}$$

For demonstrating that the SA move operator is weak to produce good results, it is sufficient to show that the probability of producing good solutions is very low with this manner. In the following it is shown that the expected value for having a solution with only 3 correct matches is about 0.02. Before that, some preliminary theorems should be explained.

Lemma 4.1. *The number of permutations of* $\{1, \ldots, n\}$*, that exactly* $m < n$ *elements are located in their true positions is:*

$$G(n, m) = \begin{pmatrix} n \\ m \end{pmatrix} D(n - m)$$

Proof. The proof is obvious, if m elements are located correctly, $n - m$ elements are not on their true positions. Derangement number of these elements is $D(n - m)$. Since the number of m-combinations of n elements is $\begin{pmatrix} n \\ m \end{pmatrix}$, the total number of the mentioned permutations is:

$$G(n, m) = \begin{pmatrix} n \\ m \end{pmatrix} D(n - m)$$

□

In the following, the probability of being $m < n$ elements in their true positions, denoted by P_m^n.

Theorem 4.2. *The probability of being* $m < n$ *elements of* n *elements on their true positions in a random permutation – denoted by* P_m^n *– is* $\frac{D(n-m)}{m! \times (n-m)!}$.

Proof. The total points in the solution space is $n!$; by using Lemma 4.1 we have:

$$\begin{aligned} P_m^n &= \frac{G(n, m)}{n!} \\ &= \frac{\begin{pmatrix} n \\ m \end{pmatrix} D(n - m)}{n!} = \frac{n! \times D(n - m)}{n! \times m! \times (n - m)!} \\ &= \frac{D(n - m)}{m! \times (n - m)!} \end{aligned}$$

□

Corollary 4.3. $\lim_{n\to\infty} P_m^n = \frac{e^{-1}}{m!}$.

Proof.

$$\begin{aligned}\lim_{n\to\infty} P_m^n &= \lim_{n\to\infty} \frac{D(n-m)}{m! \times (n-m)!} \\ &= \frac{1}{m!} \times \lim_{n\to\infty} \frac{D(n-m)}{(n-m)!}\end{aligned}$$

Substituting $n-m$ by L and using eq. (2) yields:

$$\lim_{n\to\infty} P_m^n = \frac{1}{m!} \times \lim_{L\to\infty} \frac{L!}{L!} \sum_{k=0}^{L} \frac{-1^k}{k!}$$

since:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

hence:

$$\lim_{n\to\infty} P_m^n = \frac{e^{-1}}{m!} \tag{3}$$

□

Corollary 4.4. $\lim_{n\to\infty} P_0^n$ *is* e^{-1}.

Corollary 4.5. *For large* n*, the probability of being* ***at most*** m *elements of* n *elements in their true positionsis:*

$$e^{-1} + \frac{e^{-1}}{1!} + \frac{e^{-1}}{2!} + ... + \frac{e^{-1}}{m!}$$

Corollary 4.6. *For large* n*, the probability of being* ***at least*** m *elements of* n *elements in their true positions– denoted by* $P_{1:m}^n$ *– is:*

$$P_{1:m}^n = 1 - (e^{-1} + \frac{e^{-1}}{1!} + \frac{e^{-1}}{2!} + ... + \frac{e^{-1}}{m!}) \tag{4}$$

If the rate of true matches, which is related to the number of elements lying in their true positions (m), regarded as the goodness of the solution, from the above Corollary it is obvious that, increasing the m, reduces this probability (goodness of the solution). The bad news is that this probability is close to zero, even for small values of m.

Proposition 4.7. *The probability of being at least* 3 *of* n *elements on their correct positions is* 0.02*:*

Table 2: The probability of being at least 3 elements on their true positions for different values of n. In each row 100000 permutations were generated randomly.

Number of Vertices (n)	Number of permutations with at least 3 elements in their true positions	$P^n_{1:3}$
20	1827	0.0183
50	1856	0.0186
100	1874	0.0187
300	1943	0.0194
500	1872	0.0187
1000	1939	0.0193
10000	1862	0.0186

Proof. By using Corollary 4.6, we will have:

$$\begin{aligned} P^n_{1:3} &= 1 - (e^{-1} + \frac{e^{-1}}{1!} + \frac{e^{-1}}{2!} + \frac{e^{-1}}{3!}) \\ &= 1 - (e^{-1} \times (1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!})) \\ &= 1 - (e^{-1} \times (2.6666)) \approx 0.02 \end{aligned}$$

□

Suppose that the goodness of a solution (its precision) is the total number of its elements (m) that are located on their true positions. The above inference showed that, as m increases, the probability to have a partially good solution decreases. In addition, the probability to have at least m true matches in a solution vector is bounded by a small number, which makes SA non-effective to solve graph matching problem.

5 Experimental Results

The previous theoretical discussion is verified by experimental results in this section. The experiments performed in this section are divided into two parts. In the first part, a large number of random permutations is generated and the correctness of proposition 4 is shown. In second part, this is verified by applying SA algorithm on random graphs.

5.1 Random permutations results

In order to have some experimental evidence for aforementioned theorems, some artificial graph matching problems, with known solutions are generated. Each sequences $1, 2, \ldots, n$ is considered as the vertices of the target (model) graph, and each of the $n! - 1$ permutations is the corresponding graph that should be matched with the target graph. The experiments are done for n =20, 50, 100, 300, 500, 1000, 10000. Investigation of all

Table 3: Simulated annealing results on graphs with various sizes. As the number of vertices increases, the value of the second column approaches to 0.02, which verifies propos. 4.

Number of vertices	$P^n_{1:3}$
20	0.409
50	0.182
100	0.095
300	0.041
500	0.032
1000	0.025
10000	0.02

solutions is not tractable, hence 100000 random permutations for each value of n are generated. For example for $n = 100$, 100000 random permutation is generated and the number of those permutations that at least 3 elements are located on their true positions is counted. The results is shown in table 2. The last column of the table shows $P^n_{1:3}$, which according to proposition 4 should be equal 0.02. As can be seen, the experimental results confirms that the probability of being 3 of n elements on their correct positions is 0.02 (proposition 4).

5.2 Simulated Annealing Results

Table 3 shows simulated annealing results. The graphs are produced according to [16]. For each number of verices (n), we have 100000 runs of SA. The average value of $P^n_{1:3}$, as the expected value for having a solution with at least 3 correct matches, are reported. As can be seen, as the number of vertices increases, the value of the last column approaches to 0.02, which verifies proposition 4.
All experiments using MATLAB R2018a were carried out on an Intel 2.4 GHz with 4 GB RAM and also were spent 6 hours rather. Our database and MATLAB codes are accessible from Mendeley[1].

6 Conclusions

In this study, an upper bound for minimum true matches in graph matching problem using simulated annealing has been proposed. Graph matching problem for applications like image stereo matching, has just one optimal solution. We argued that finding the optimal solution with stochastic optimization methods are not feasible, due to the low probability of the number of true matches. Mathematical derivation showed that the success rate is very low, in a random based search method. For example the probability

[1] http://dx.doi.org/10.17632/5vtn4h49p3.1

that at least 3 nodes are matched correctly is about 0.02. Experimental results verified theoretical discussion.

Conflict of interest The authors declare that they have no conflict of interest.

References

[1] Almohamad, H. A. and Duffuaa, S. O. A linear programming approach for the weighted graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 15(5):pp. 522–525, May 1993.

[2] Auwatanamongkol, S. Inexact graph matching using a genetic algorithm for image recognition. *Pattern Recognition Letters*, 28(12):1428 – 1437, 2007.

[3] Bengoetxea, E. *Inexact Graph Matching Using Estimation of Distribution Algorithms*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, Paris, France, Dec 2002.

[4] Brown, R. D., Jones, G., Willett, P., and Glen, R. C. Matching two-dimensional chemical graphs using genetic algorithms. *Journal of Chemical Information and Computer Sciences*, 34(1):63–70, 1994.

[5] Choi, J., Yoon, Y., and Moon, B.-R. An efficient genetic algorithm for subgraph isomorphism. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 361–368, New York, NY, USA, 2012. ACM.

[6] Cross, A. D., Myers, R., and Hancock, E. R. Convergence of a hill-climbing genetic algorithm for graph matching. *Pattern Recognition*, 33(11):1863 – 1880, 2000.

[7] Cross, A. D., Wilson, R. C., and Hancock, E. R. Inexact graph matching using genetic search. *Pattern Recognition*, 30(6):953 – 970, 1997.

[8] Egozi, A., Keller, Y., and Guterman, H. A probabilistic approach to spectral graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol 35(1):pp. 18–27, January 2013.

[9] Finch, A. M., Wilson, R. C., and Hancock, E. R. Symbolic graph matching with the em algorithm. *Pattern Recognition*, Vol 31(11):pp. 1777 – 1790, 1998.

[10] Fröhlich, H., Košir, A., and Zajc, B. Optimization of fpga configurations using parallel genetic algorithm. *Information Sciences*, 133(3):195 – 219, 2001. Evolutionary Algorithms.

[11] Herault, L., Horaud, R. P., Veillon, F., and Niez, J.-J. Symbolic image matching by simulated annealing. In *British Machine Vision Conference, BMVC 1990, September, 1990*, pages 319–324, Oxford, Royaume-Uni, September 1990.

[12] Heris, J. E. A. and Oskoei, M. A. Modified genetic algorithm for solving n-queens problem. In *2014 Iranian Conference on Intelligent Systems (ICIS)*, pages 1–5, Feb 2014.

[13] Ishii, Y. W. N. A genetic algorithm and its parallelization for graph matching with similarity measures. In *Artificial Life and Robotics*, volume 2, pages 68–73, 1998.

[14] Jabari, S., Moazzami, D., and Ghodousian, A. Heuristic and exact algorithms for generalized bin covering problem. *Journal of Algorithms and Computation*, 47(1):53–62, 2016.

[15] Knuth, D. E. *Dancing Links*, pages 159–187. Palgrave, 2000.

[16] Kpodjedo, S., Galinier, P., and Antoniol, G. Enhancing a tabu algorithm for approximate graph matching by using similarity measures. In *Evolutionary Computation in Combinatorial Optimization, 10th European Conference, EvoCOP 2010, Istanbul, Turkey, April 7-9, 2010. Proceedings*, pages 119–130, 2010.

[17] Leordeanu, M. and Hebert, M. A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision (ICCV'05),Vol 2*, pages 1482–1489, Oct 2005.

[18] Luo, B. and Hancock, E. R. Structural graph matching using the em algorithm and singular value decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol 23(10):pp. 1120–1136, October 2001.

[19] Qiao, C., Luo, J., Shen, Z., Zhu, Z., and Ming, D. Adaptive thematic object extraction from remote sensing image based on spectral matching. *International Journal of Applied Earth Observation and Geoinformation*, Vol 19:pp. 248–251, 2012.

[20] Remmel, J. B. A note on a recursion for the number of derangements. *Eur. J. Comb.*, 4(4):371–374, 1983.

[21] Samie Yousefi, F., Karimian, N., and Ghodousian, A. Xerus optimization algorithm (xoa): a novel nature-inspired metaheuristic algorithm for solving global optimization problems. *Journal of Algorithms and Computation*, 51(2):111–126, 2019.

[22] Sammoud, O., Solnon, C., and Ghédira, K. *Ant Algorithm for the Graph Matching Problem*, pages 213–223. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[23] Shapiro, L. S. and Brady, J. M. Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, Vol 10(5):pp. 283– 288, 1992.

[24] Shettar, V. M. An exploration of algorithmic and theoretical approaches to graph matching and its applications, 2014. Master Thesis.

[25] Szeliski, R. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[26] Tohyama, H., Ida, K., and Matsueda, J. A genetic algorithm for the uncapacitated facility location problem. *Electronics and Communications in Japan*, 94(5):47–54, 2011.

[27] Umeyama, S. An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol 10(5):pp. 695–703, September 1988.

[28] Williams, M. L., Wilson, R. C., and Hancock, E. R. Deterministic search for relational graph matching. *Pattern Recognition*, Vol 32(7):pp. 1255 – 1271, 1999.

[29] Wu, Y., Gong, M., Ma, W., and Wang, S. High-order graph matching based on ant colony optimization. *Neurocomputing*, 328:97 – 104, 2019. Chinese Conference on Computer Vision 2017.

[30] Yan, J., Yin, X.-C., Lin, W., Deng, C., Zha, H., and Yang, X. A short survey of recent advances in graph matching. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, ICMR '16, pages 167–174, New York, NY, USA, 2016. ACM.