Unifying Graph Matching Problems with a Practical Solution

Chao-wen Kevin Chen[1] and David Y. Y. Yun[2]
Laboratory of Intelligent and Parallel Systems
Department of Electrical Engineering
University of Hawaii at Manoa
Email: cwchen@spectra.eng.hawaii.edu[1] or dyun@spectra.eng.hawaii.edu[2]

ABSTRACT

Recent advances in computer vision, information retrieval, and molecular biology amply demonstrate the value and importance of graph matching. However, graph matching algorithms developed to date are either application domain-specific or severely limited by the problem size. This paper presents a unified solution for the broad range of major graph matching problems. Particularly, the new algorithm goes beyond matching given pattern graphs and effectively tackles two very difficult graph matching problems -- Maximal Common Subgraph and Maximal Overlap Sets. By mathematical proofs, these graph matching problems can be shown to be transformable into the *maximum clique* problem. This new algorithm actually alternates between clique-finding and graph coloring by utilizing the information generated from the counterpart to either solve the maximum clique problem or, alternatively, the graph coloring problem as well. The effectiveness of this algorithm to solve all these graph matching problems is then evaluated with an extensive set of benchmark graphs for clique finding. The solutions show consistent superior solution quality.

Keywords: Graph Matching, Maximum Clique, Maximal Common Subgraph, Maximal Overlap, Coloring.

1. INTRODUCTION

Graph matching is one of the most important techniques in pattern recognition. It has been studied in the simpler (but already computationally difficult) forms of Graph Isomorphism (GI), Subgraph Isomorphism (SGI) [1, 2], and in the most complex forms of *Maximal Common Subgraph* (MCS) Isomorphism and *Maximal Overlap Set* (MOS) [3, 4], for both theoretical and practical interests.

Though recent advance in the SGI research has led to efficient solutions for certain practical applications [2], MCS and MOS problems have largely remained intractable due to the added complexity from combinatorial possibilities of patterns. All published MCS/MOS algorithms are still limited to solving small graphs. Nevertheless, the need for an effective MCS algorithm is clearly called for by many real-world problems [3, 4, 5, 6].

Levi [7] used a more restrictive definition of MCS that requires isomorphism between vertex-induced sub-graphs. McGregor [3], later, pointed out that Levi's definition is not appropriate as a measure of the similarity of graphs, and adopted Velduts's Maximal Overlap Set definition by relaxing edge correspondence. To our knowledge, McGregor's algorithm is the only one existing that solves MOS problems. However, our experiments show that it can hardly handle graph sizes beyond 30 nodes due to its branch-and-bound search scheme and its lack of good heuristics.

In search of the solution for MCS, Barrow et al. [8] first introduced a transform to convert MCS into the Maximum Clique problem. We recently showed [9], independent from Kann's recent work [10], that the MOS problem can also be transformed into an MC problem. Consequently, devising a good maximum clique algorithm may provide solutions to several related graph matching problems.

This paper proposes a unified solution to these graph matching problems by first providing the corresponding transformation to MC, in Section 2 after a brief introduction. Then in Section 3, a constructive algorithm CRP-MC that couples MC and Graph Coloring (GC) in an iterative loop will be presented. A series of benchmark testing that evaluates the effectiveness of the proposed algorithm is shown in Section 4. Finally, Section 5 asserts some conclusions.

2. GRAPH MATCHING

2.1 Problem Definitions:

Graph Matching often utilizes the concept of isomorphism, which simply means that two graphs are topologically identical. (*Sub)graph Isomorphism*, then, means that one graph contains an isomorphic copy of another.

To define the MCS and MOS problems, the concept of *induced graph* need to be introduced,

Definition 1: Given a graph **G(*v,e*)**, and a subset $V' \in V$ (or $e' \in e$), the *vertex* (or *edge*) *induced subgraph* **G|*v'*** (or **G|*e'*)** is the subgraph formed by ***v'*** (or ***e'***) and all of their adjacent edges (nodes).

Definition 2: The *MCS* problem [7] is defined as finding a common *vertex-induced* subgraph of two graphs with the maximal number of nodes.

Definition 3: *Maximal Overlap Set (MOS)* [3] is defined as finding a common *edge-induced* subgraph of two graphs with the maximal number of edges.

2.2 Transform to Maximum Clique Problem:

The graph matching problems mentioned above can all be transformed to *Maximum Clique* (MC):

Definition 4: A *clique* is a complete graph. The *Maximum Clique Problem* is to find the clique with the maximum number of nodes in a given graph.

[Transform from SGI (or GI) to MC] The transform from SGI into *k*-clique problem (i.e. maximum clique size is know to be *k*) was first introduced by Cook [11]. However, it's never been applied to the SGI problem probably due to the lack of efficient MC algorithm.

[Transform from MCS to MC] Barrow et al. [8] proposed a transform to convert MCS into the MC problem by the following procedures:

Given a pair of labeled graphs $\mathbf{G_1}$ and $\mathbf{G_2}$, create a *correspondence graph* **C** by,

1) Create the set of all pairs of same labeled nodes, one from each of the two graphs.

2) Form the graph **C** whose nodes are the pairs from (1). Connect any two node pairs $\mathbf{N_1(A_i, B_x)}$, $\mathbf{N_2(A_j,B_y)}$ in **C** if the labels of the edges from $\mathbf{A_i}$ to $\mathbf{A_j}$ in $\mathbf{G_1}$ and $\mathbf{B_x}$ to $\mathbf{B_y}$ in $\mathbf{G_2}$ are the same.
3) Maximal common subgraphs then correspond to the maximum cliques of **C**

[Transform from MOS to MC] Recently, Kann [10] showed in his dissertation work that the MOS problem can also be transformed into the MC problem. However, this theoretical result is less practical since it generates a large correspondence graph. Independent from his work, we [9] devised a similar transform with a two-stage reduction process that results in much smaller correspondence graphs in practical cases:

1) An Edge Unit (EU) [2] is defined as an edge and its two end nodes, together with their labels.
2) Remove all edge in $\mathbf{G_1}$ and $\mathbf{G_2}$ whose corresponding EUs do not have any match on the counterpart ➔ $\mathbf{G_1}'$ and $\mathbf{G_2}'$ **(First Reduction Process)**
3) Examine each EU pair (u_1, u_2) from $\mathbf{G_1}'$ and $\mathbf{G_2}'$, if u_1, u_2 have the same label, and,

[Ambiguity Condition]

If (a) u_1, and u_2 are in a *triangle-fork dual subgraph pair* (see Figure 1(f)), where the two edges other than u_1 in one subgraph match with the edges other than u_2 in the other, and,
(b) all the end nodes of the edges in the subgraph pair have the same label,
Then generate two corresponding directional nodes $N(u_1^+, u_2^+)$, $N(u_1^-, u_2^+)$ in the correspondence graph **C**. (where direction is treated as an extra label to be matched)
Else generate one undirectional correspondence nodes $N(u_1,u_2)$. **(Second Reduction Process)**

4) In the correspondence graph, connect any two node pairs $N_1(u_{11}, u_{21})$, and $N_1(u_{12}, u_{22})$ if either,
(a) u_{11} connects to u_{12} , u_{21} connects to u_{22} , and they both connect at nodes of the same labels, or,
(b) u_{11} does not connect to u_{12} and u_{21} does not connect to u_{22}
5) The maximal overlap sets then correspond to the maximum cliques of **C**

In other words, this transform process is basically similar to the MCS-to-MC transform except that the edge (or edge unit) is considered instead of node. The other major difference is that the Ambiguity Condition used in Step 3 requires establishing two correspondence nodes in certain cases.

Theorem 1: In the above MOS to MC transform, the necessary and sufficient condition to cause ambiguity is u_1, and u_2 are contained in a *triangle-fork dual subgraph pair* (where the edge labels matched). In addition, all the end nodes of the six edges in the subgraph pair have the same label (Details in [9]).

Proof: (1) It's trivial to show that there is no ambiguity if there are less than 3 connected EU pairs.

(2) When given 3 connected EU pairs, there are 6 possible combinations as in Figure 1. In case (a)-(c), the two subgraph match can be reflected in the correspondence graph. Whereas in case (d)-(e), the connection counts of the two subgraphs differ, so it is detectable in the transform. The only exception in case (f), where $\mathbf{G_1}$ and $\mathbf{G_2}$ might be mistaken to be matched since the inter-connections among the edge in both graphs are identical. **[Condition 1]**

(3) Case (f) is considered as a 3-edge match only when a_1, a_2, b_1, and b_2 have the same label so that the connections among u_1, A_1, A_2 are consistent with u_2, B_1, B_2. In addition, a_3 has to be the same as b_2 so that A_1, A_2 connect to the same label as B_1 and B_2. Consequently, b_3 and b_4 need to have the same label as b_2. Therefore, all seven end nodes in the subgraph pair need to have the same label. **[Condition 2]**

(4) By mathematical induction, it's easy to see that the conditions sustain for cases with more than three nodes.

Therefore, with the appropriate transform, these graph matching problems can then be solved by developing an efficient maximum clique algorithm.

2.3 Previous Work:

The research on graph matching began in the 70's. Early work focused mainly on the graph isomorphism [12], and sub-graph isomorphism problems [1], and the graph sizes reported were usually about 20 nodes. Recently, Ling and Yun [2] have presented a new SGI algorithm that showed to be able to handle large graph sizes in reasonable time given that the graphs have bounded-degree and their labels are locally distinguishable in some sense. Nevertheless, these conditions and the 'center' concept employed in that algorithm can hardly apply to the MCS/MOS cases. There have been several MCS and MOS algorithms [13] devised in the past decade due to the increasing interest in the chemical and molecular biology fields. However, most of them used certain domain-specific knowledge, such as the particular conditions for atom bonds, and the bounded-degree characteristics observed in the chemical molecules. These features are usually not available in the other cases.

Few algorithms have been designed to tackle the general MCS and MOS problems. One exception is the one proposed by McGregor [3], which is a branch-and-bound, pair-wise comparison type of algorithm. Although his algorithm is intended for MOS problem (and it's the only one up to now), it can be easily modified for the MCS problem. However, as noted by the other researchers' and our own experiments, this algorithm can hardly handle graphs with more than 30 nodes. There have been several attempts on applying the maximum clique algorithms to solve the MCS problem [4, 14]. All these algorithms use the same maximum clique algorithm, **Algorithm 457** [15], which is basically a branch-and-bound

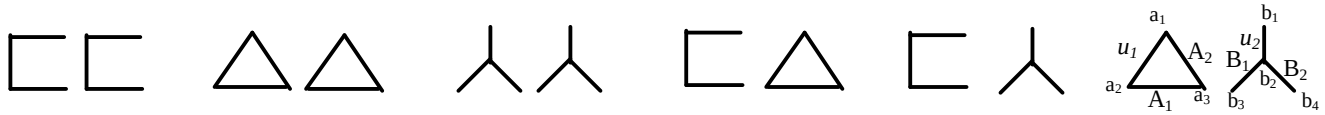


Figure 1 All six possible connected graph pairs with three edges

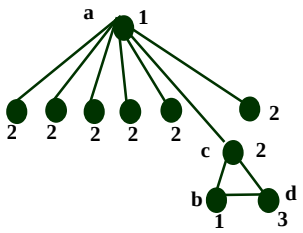


Figure 2: Minimum Coloring

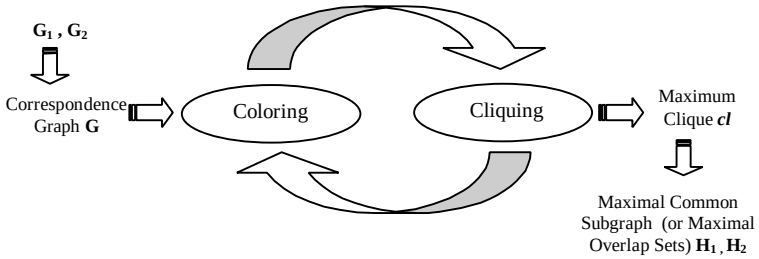


Figure 3: Framework of MCS/MOS Algorithm

search algorithm. It is observed in [4] that using **Algorithm 457** usually solves the MCS problem in the transformed domain faster the pair-wise comparison algorithms such as McGregor's. In our own experiments, we also experienced a hundred fold of speed-up for large test graphs. Based on these advantages and the universal properties as discussed above, we propose to tackle these graph matching problems with the following maximum clique algorithm with appropriate transform as discussed earlier.

3 MAXIMUM CLIQUE ALGORITHM (CRP-MC)

Most MC algorithms employ heuristics based on node degrees [16,17]. The idea is that a node with large degree has a higher probability of belonging to a large clique. It is easy to see how this heuristic tends to fail. As show in Figure 2, although node **a** has the highest degree (6), it does not belong to the largest clique, namely, {**b,c,d**}. Instead of using heuristics on node degrees, we found that graph coloring provides more accurate information for finding large cliques. (node **a** is adjacent to only one color, while **b,c,d** to two colors each).

3.2 Algorithm CRP-MC:

Definition 5*:* Given a graph **G**, a (*vertex*) *Graph Coloring* (GC) is an assignment of *k* colors, 1, 2, …, *k*, to the vertices of **G** such that no two adjacent vertices use the same color. The *Minimum Coloring* of **G** is an assignment using the minimum number of colors (*k*). This minimum number, $l(G)$, is called the *chromatic number* of **G**. For example, a valid graph coloring is given in Figure 2, which is also a minimum coloring with a chromatic number of 3.

***Lemma 1*:** Given a graph **G**, the relation between the size of the maximum clique, $w(G)$, and the chromatic number, $l(G)$, is $w(G) \le l(G)$.

Definition 6: Given a graph **G**(***v, e***) and a coloring, the *color-degree* of vertex v_i, **cdeg**(v_i), is defined as the number of different colors of the adjacent nodes.

Lemma 2*:** Given a graph **G**(v, e***), for any vertex v_i , let the size of the maximum clique that includes v_i be $w(G \mid v_i)$. Then, $w(G \mid v_i) \le (\text{cdeg}(v_i)+1)$. (The proof follows immediately from *Lemma 2*.)

By utilizing the concepts of Lemma 1 & 2, we developed a maximum clique algorithm (Figure 3) called **CRP-MC** that couples two sub-algorithms, **COLORING** (Figure 4) and **CLIQUING (**Figure 5**)**. The two algorithms are executed alternatingly in an iterative loop, as shown in Figure 3, and controlled by a tested, efficient resource management technique, *Constraint Resource Planning* (CRP) [18]. (1) **COLORING** takes the cliques found by **CLIQUING** as input, and minimizes the number of colors needed for the correspondence graph. (2) **CLIQUING** initializes with the nodes partitioned according to the colors from the previous **COLORING** and operates on the fact that no two nodes of the same color partition can belong to the same clique.

At the **COLORING** stage, described in Figure 4, the node that has the least number of valid colors is chosen for processing first because of its lack of flexibility. The color that has the least influence (chance to restrict the color set of others) on the other uncolored nodes is then assigned to the chosen node to maximize the available color sets to the nodes still to be colored.

On the other hand, the **CLIQUING** stage (see Figure 5) uses the dual concept. Since for any clique in the graph, each color partition provides at most one node, the least flexible yet important color partition should be considered first. Then a node from that color partition with the highest potential (based on the color degrees) of being in a large clique is chosen.

In general, the iterative process of **CRP-MC** terminates when (1) the clique size equals the number of colors used, since no larger clique can be possible (Lemma 1), or (2) the

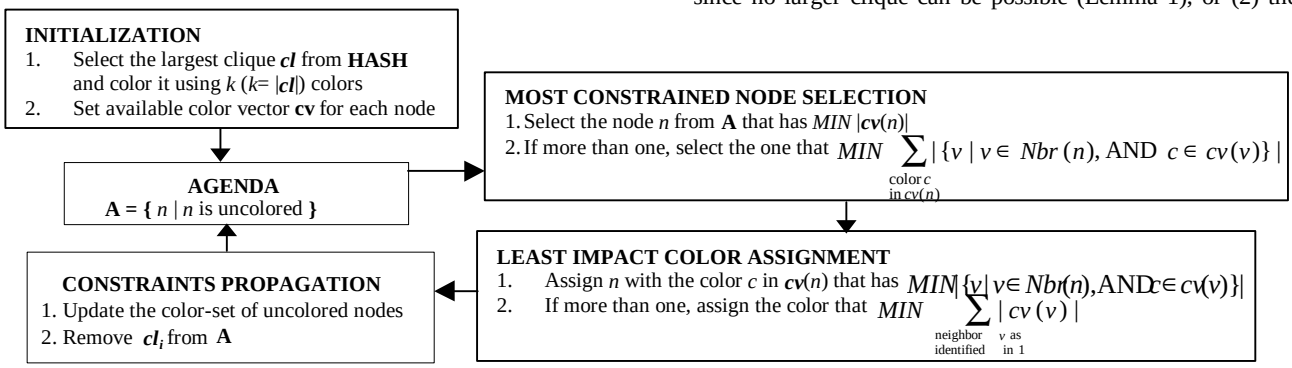


Figure 3 COLORING Algorithm

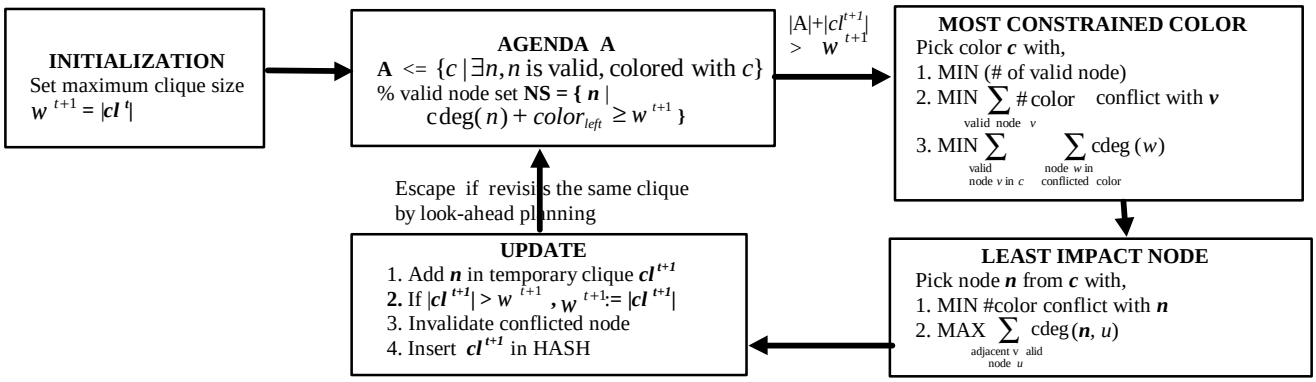


Figure 5 CLIQUING Algorithm

distribution of colors in COLORING is not changed anymore, since no more improvement is possible from this iterative refinement process.

However, for the case of Graph (or Subgrah) Isomorphism, if the to-be-matched graph pattern is of size *k*, then there has to be a *k*-clique in the correspondence graph if there is an isomorphism. Therefore, the CRP-MC can be modified to terminate when (1) A *k*-clique is found, (➔ isomorphic) or, (2) A coloring uses less than *k* colors is obtained (➔not isomorphic). The advantage of using maximum clique for solving SGI (or GI) problem is that when the chromatic number is detected to be less than the to-be-matched graph pattern size, then there is no graph or subgraph isomorphism and the process can be terminated much earlier, rather than having to exhustively search the whole search space.

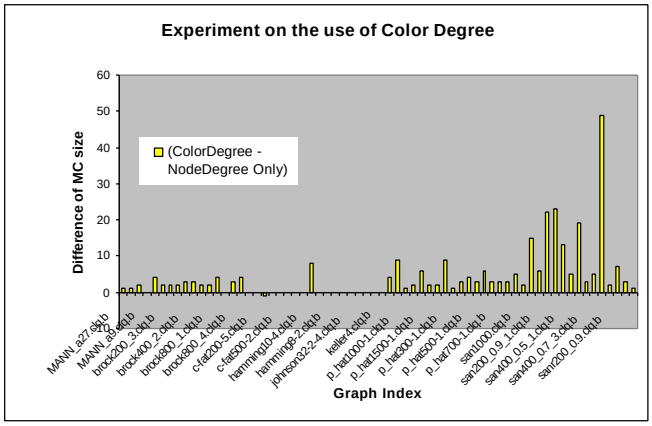


Figure 6 Effects of using color-degree

4 EXPERIMENTS

4.1 Validation of Color-degree Concept:

A major feature that is first proposed in this paper is the use of *color-degree*. In order to verify the effectiveness of this concept, the first experiment is designed to compare it with the one that only uses node-degree, as employed by the other existing maximum clique algorithms. In order to maintain the other factors unchanged, the two compared algorithms are constructed as follows, (1) With Node-degree Only : using the implemened CRP-MC algorithm as backbone, while forcing the **COLORING** module to generate distinct color for each node. Thus removing the information of color degree. (2) With Color-degree : using the CRP-MC without any modification. The test is done by only using one iteration since the node-degree only one will not benefit from the iterative scheme.

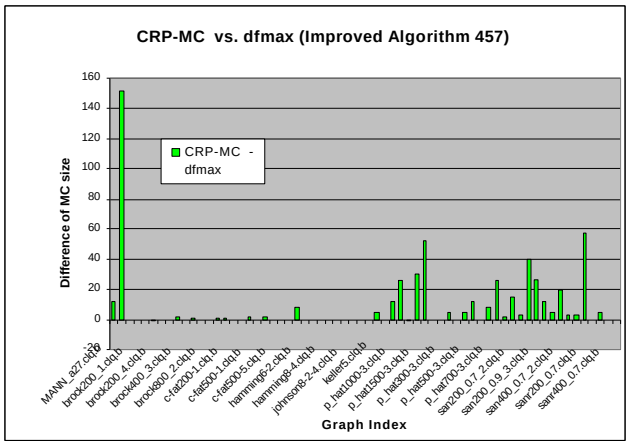
The results are shown in Figure 6, where each vertical bar represents the difference between the maximal clique sizes found by using color-degree, and node-degree only. Out of 66 standard benchmark test graphs, the algorithm using color-degree achieves large cliques that the other in 47 cases, and only loses in one case. Note that the solution quality can be better by up to several tens of nodes.

4.2 Comparison with the other algorithms:

As discussed earlier, the only maximum clique algorithm that has been used for graph matching (MCS) is the **Algorithm 457**, therefore, an improved version of this algorithm, *dfmax*, is chosen for our first comparison. In addition, a neural network based maximum clique algorithm, *gsd0*, recently developed by Jagota [19], reportedly to yield good approximate solution quality, is also used for comparison.

The experiment is done on the same benchmark graphs, and the same chart representation as previous experiment is used. Due to that the *gsd0* algorithm tends to saturate on its solution quality after certain iterations, and the *dfmax* algorithm requires much longer time than the others to complete its search, we decide to compare the performance by terminating all programs when the *gsd0* stops to improve its solution. Figure 7(a) shows the comparison result between CRP-MC and the *dfmax* algorithm. The CRP-MC algorithm consistently finds better (31 out of 66 cases) or equal (35 out of 66 cases) quality than the ones by *dfmax*. Similarly, in Figure 7(b), the CRP-MC algorithm finds 36 better solutions than the ones by *gsd0*, 29 equal quality solutions, and only one that is worse. Also note that the solution quality in the winning cases can often be better by more than five nodes, while the only losing case is differed by merely one node.

The practical use of this CRP-MC algorithm has also been demonstrated with a computer vision problem, the Automation of 3D Object Model Reconstruction from Multiple Line-drawing Views. The details are shown in [20].



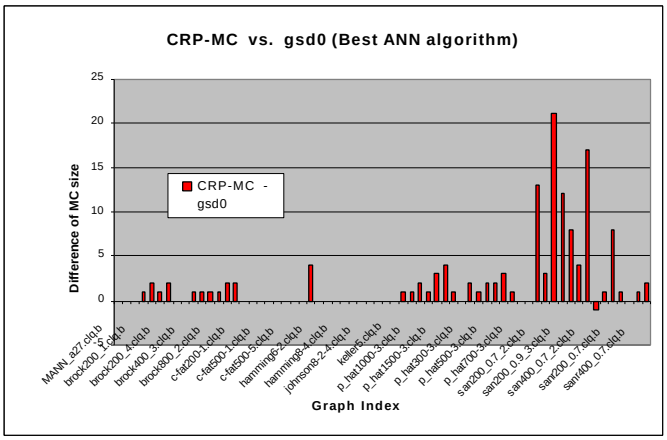


Figure 5 (a) Comparison between CRP-MC and dfmax **(b)** Comparison between CRP-MC and gsd0

5. CONCLUSIONS

This paper presents a unified solution to the graph matching problems by providing algorithmic transformations from MCS and MOS problems to the MC problem. Rigorous mathematical derivations are also provided to show the correctness of these transforms. The result is an extension and completion of previous works on the transformation of various graph isomorphism problems to MC.

In the effort of designing a practical maximum clique algorithm as the core process for the unified solution, a novel algorithm, **CRP-MC**, which tightly couples with the minimum coloring and effectively exploits this duality relationship, has been developed. As pointed out already in this paper, the **CRP-MC** algorithm has a much better chance of tackling large graph matching problems, while conventional MC algorithms based on node-degree tend to fail.

The efficiency of the **CRP-MC** algorithm and the use of proposed *color-degree* are demonstrated with a set of standard benchmark graphs. The experiments show consistently superior solution quality to the other two maximum clique algorithms commonly used for MCS problem. Finally, in order to devise a more robust and universal algorithm, current efforts are focusing on the algorithm effectiveness on different type of graphs, as well as on the relationships and characteristics between the matching graphs and their correspondence graphs.

REFERENCES

[1] J. R. Ullmann, "An Algorithm for Subgraph Isomorphism," *J. ACM*, vol. 23, no. 1, pp. 31-42, 1976.

[2] Z. Ling and David Y. Y. Yun, "An Effective Approach for Solving Subgraph Isomorphism Problem," *Proceedings of the IASTED Int'l Conference on Artificial Intelligence, Expert Systems and Neural Networks*, pp. 342-345, August 1996.

[3] J. J. McGregor, "Backtrack Search Algorithms and the Maximal Common Subgraph Problem," *Software-Practice and Experience*, vol. 12, pp. 23-34, 1982.

[4] A. T. Brint and P. Willett, "Algorithms for the Identification of Three-Dimensional Maximal Common Substructures," *J. Chem. Inf. Comut. Sci.*, vol. 27, pp. 152-158, 1987.

[5] H. Ogawa, "Labeled Point Pattern Matching by Delaunay Triangulation and Maximal Cliques," *Pattern Recognition*, vol. 19, no. 1, pp. 35-40, 1986.

[6] R. Horaud and T. Skordas, "Stereo Correspondence through Feature Grouping and Maximal Cliques," *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 11, no. 11, pp. 1168-1180, Nov. 1989.

[7] G. Levi, "A Note on the Derivation of Maximal Common Subgraphs of the Directed or Undirected Graphs," *Calcolo*, 9, pp.1 –12, 1972.

[8] H. G. Barrow and R. M. Burstall, "Subgraph Isomorphism, Matching Relational Structures and Maximal Cliques," *Information Processing Letters*, vol. 4, no. 4, pp. 83-84, 1976.

[9] Chao-wen Kevin Chen and David Y. Y. Yun, "Toward Solving Maximal Overlap Sets Problem," *Technical Report TR-LIPSC&Y96a*, 1996, LIPS, University of Hawaii.

[10] V. Kann, "On the Approximability of NP-complete Optimization Problems," PhD. Thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, 1992.

[11] S. A. Cook, "The complexity of theorem-proving procedures," *Proc. 3rd Ann. ACM Symp. on Theory of Computing*, Association for Computing Machinery, New York, 151-158.

[12] D. G. Corneil and C. C. Gotlieb, "An Efficient Algorithm for Graph Isomorphism", *J. of Association for Computing Machinery*, Vol.17, No.1, pp.51-64, 1970.

[13] C. W. Crandell and D. H. Smith, "Computer-assisted Examination of Compounds for Common Three-dimensional Substructures," *J Chem. Inform. Comp. Sci.*, v.23, pp.186-197.

[14] H. M. Grindley, P. J. Artymiuk, D. W. Rice and P. Willett, "Identification of Tertiary Structure Resemblance in Proteins Using a Maximal Common Subgraph Isomorphism Algorithm", *J. Mol. Biology* (1993) 229, pp. 707-721.

[15] C. Bron and J. Kerbosch, "Algorithm 457 – finding all cliques of an undirected graph," *Comm. ACM*, v16, pp. 575-577, 1973.

[16] Egon Balas and Chang Sung Yu, "Finding a Maximum Clique in an Arbitrary Graph," *SIAM Computing*, vol. 15, no. 4, pp. 1054-1068, Nov. 1986.

[17] P.M. Pardalos, "A Branch and Bound Algorithm for the Maximum Clique Problem," *Comp. Opertations Res.*, vol. 19, no. 5, pp. 363-375, 1992.

[18] Keng, N. P. and D. Y. Y. Yun, "A Planning/Scheduling Methodology for the Constrained Resource Problem," *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 20-25, 1989.

[19] A. Jagota, "Approximating Maximum Clique with a Hopfield Network," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 724-735, May 1995.

[20] Chao-wen Kevin Chen and D. Y. Y. Yun, "Automation of 3D Object Modeling from Multiple Views," *1998 Symposium on Image, Speech, Signal Processing and Robotics (ISSPR'98)*, Hong Kong, Sept. 1998.