RESEARCH STATEMENT

Yinghui Wu (Y.Wu-18@sms.ed.ac.uk)
University of Edinburgh

My research interests span the areas of (dynamic) graph models and algorithms, web service models and algorithms, and data quality. The main topics of my research is in (1) developing novel graph (pattern) models as well as the related theoretical analysis; (2) developing novel techniques for static and dynamic graph (pattern) matching; and (3) developing novel techniques in web service models, as well as *e.g.,* Web service composition and aggregation, among other related areas. In general, my research belongs to the category of graph databases, as well as the network models and algorithms design.

Background and Current Work

Among the vital issues in the emerging application involving networks and other real life graphs is the graph (pattern) matching problem. Generally speaking, (1) given two graphs, the *graph matching problem* determines if the two graphs are similar based on some similarity metric; (2) given a small pattern (query) graph and a large data graph, the *graph pattern matching problem* computes all the valid matches of the pattern graph in the data graph. The above problems have a wide application in the fields covering social matching, network querying, information extraction, object identification, bioinformatics and chemistry industry, among many other areas.

There exists a rich body of previous work in the graph matching problem, most of which model the graph (pattern) matching based on *graph homomorphism, subgraph isomorphism* and *graph simulation.* Specifically, the graph homomorphism (resp. subgraph isomorphism) is a *function* (resp. bijection) over the nodes of one graph to the nodes of the other (resp. a subgraph of the other) that indicates an edge-to-edge mapping. Similarly, the graph simulation is a *binary relation* over the nodes of the two graphs that indicates an edge-to-edge matching.

A closer observation of these models reveals that (1) graph homomorphism and subgraph isomorphism are (a) based on label equality of nodes and edge-to-edge mapping, and (b) hard to compute (NP-hard, *i.e.,* there is no exact polynomial time algorithms); (2) graph simulation is based on label equality and edge to edge matchings; (3) the incremental graph (pattern) matching over dynamic graphs based on these models have not been studied much, which is, however, extremely important in *e.g.,* social networks. The graph (pattern) matching approaches based on these are thus too restrictive to capture both structural and semantic similarity in real life graphs, and are often lack of performance guarantees over matching quality.

Homomorphism Revised for Graph Matching

The graph matching in *e.g.,* object identification [4], web mirror detection [1] among other areas are mostly defined in terms of either graph homomorphism or subgraph isomorphism, reflecting the *exact* matching that requires edge to edge mapping. In real life applications this may be too restrictive. For example, an edge in one web graph may correspond to a path of arbitrary length in another.

This motivate us to extend the traditional graph homomorphism into a more general case. In our work, (1) I explored the real life application scenarios, and formulated (with a colleague) the concepts of *p-homomorphism* (p-hom) and 1-1 p-hom, which extend the graph homomorphism and subgraph isomorphism, respectively, by mapping the *edges* from one graph to the *paths* in the other. (2) We developed the metrics based on these extensions and introduce novel graph similarity measurements. These reformulate the graph matching into two optimization problems having the quantitative measurements on the matching quality. I showed that these two problems are hard to approximate (APX-hard [10]), while they can be solved with performance guarantees in polynomial time. (3) I came up with the approximation algorithms in (2), and further developed optimization techniques for the algorithms with my colleagues. (4) I utilized both real life (*e.g.,* web graphs [6]) and synthetic datasets to verify the effectiveness and efficiency of our model and algorithms.

The model along with the techniques we proposed presently incorporates and extends the traditional matching methods defined over homomorphism and isomorphism, and is capable for capturing more meaningful matches efficiently in real life applications, *e.g.,* web mirror detection, verified by our experimental study.

Simulation Revised for Graph Pattern Matching

The *graph simulation* [5] has been used to model graph pattern matching problem in many applications *e.g.,* social matching [2, 9], where the matches for a pattern graph consist of all the simulators of the pattern nodes. However, the edge-to-edge matching simulation relies on is too restrictive for finding the matches. For example, in *e.g.,* social networks and traffic networks the valid matches for a pattern node often exists not only within its neighbors but within the nodes several hops away from it. On the other hand, the *p*-homomorphisms we proposed are still too restrictive for the matches as relations rather than functions, and remain to be NP-hard.

In light of this, we developed an extended model for graph simulation, namely, the *bounded simulation*, in which an edge of a pattern graph denotes the connectivity in a data graph within a predefined number of hops. (1) I (along with a colleague) formulated an enriched model for the pattern and data graphs, based on which we proposed the concept of the bounded simulation, and formulated the graph pattern matching based on the bounded simulation relation. (2) I showed that with this revision, graph pattern matching can be performed in *cubic-time* by providing such an algorithm, rather than intractable as its counterpart defined with functions. In addition, I have shown a strong result that there exists a match result that is *maximum and unique.* (3) I looked into the difference between the result representations of various graph pattern matching approaches, and proposed a set of *result graphs* to intuitively represent the matching result as a graph, rather than a relation. (4) I experimentally verify the effectiveness and efficiency of our model and algorithms, using three real life datasets (*e.g.,* Youtube, PBlog, and Matter) and synthetic datasets.

The above model extends the graph simulation. The graph pattern matching problem defined over the model can be solved in polynomial time with an exact, maximum and unique solution, in contrast to its counterpart defined with homomorphism and isomorphism, which are NP-hard and may have exponential number of solutions. In one word, the model provides a promising graph pattern matching method that is both effective and efficient over complex networks such as social networks, as verified by our experimental study.

Reachability Queries and Graph Pattern Queries

It is increasingly common to find real life graphs in which edges bear different types, indicating a variety of relationships (*e.g.,* [3]). For such graphs we proposed a class of reachability queries (*RQs*) and a class of graph patterns (*PQs*), in which an edge is specified with a regular expression of a certain form, expressing the connectivity in a data graph via edges of various types. In our work, (1) I explored different real life application scenarios, based on which we formulated the concepts of *RQs* and *PQs*, and further formulated the graph query evaluation problems; (2) I (along with a colleague) investigated the containment and minimization problems for *RQs* and *PQs*. We showed that these fundamental problems are in quadratic time for *RQs* and are in cubic time for *PQs*, respectively. (3) I (along with a colleague) developed algorithms for answering *RQs*, in quadratic time as for their traditional counterpart. (3) I provided two cubic-time algorithms for evaluating *PQs*, as opposed to the NP-completeness of graph pattern matching via subgraph isomorphism. In addition, I developed the optimization techniques for the algorithms to reach the low computational complexity. (4) I showed the effectiveness, efficiency and scalability of these algorithms via an experimental study using both real-life data (*e.g.,* Youtube, Global Terrorist Network) and synthetic data.

These graph queries further generalize the bounded simulation patterns, and are capable for capturing the multiple edge types for the graph pattern matching problem in the emerging applications *e.g.,* social networks. Better still, their increased expressive power does not come with extra complexity, not only for fundamental problems such as containment and minimization, but also for the query evaluation.

Incremental Graph Pattern Matching

In practice a data graph is typically large, and is frequently updated with small changes [7]. It is often prohibitively expensive to recompute matches from scratch via *batch* algorithms when the graph is updated. With this comes the need for *incremental* algorithms that compute *changes* to the matches in response to updates, to minimize unnecessary recomputation. I have noticed that there is little work on the incremental graph pattern matching. In the following work I investigated the incremental algorithms for graph pattern matching defined in terms of graph simulation, bounded simulation and subgraph isomorphism, respectively.

Incremental simulation. For graph simulation, I have developed incremental algorithms for unit updates and certain graph patterns. These algorithms are *optimal*: in linear time in the size of *the changes* in the input and output, which characterizes the cost that is inherent to the problem itself [8]. For general patterns, I showed that the incremental matching problem is *unbounded*, *i.e.,* its cost is not determined by the size of the changes alone. In addition, I developed an incremental algorithm for a set of updates consisting of both edge insertions and deletions with performance guarantee that does not depend on the size of the data graph.

Incremental bounded simulation. I investigated the incremental bounded simulation problem and showed that it is unbounded even for unit updates and path patterns. In addition, I developed two incremental algorithms using the distance matrix and the landmark vectors, respectively, for single updates. Similarly, I developed two incremental algorithms for the bounded simulation dealing with multiple updates.

Incremental subgraph isomorphism. I (along with a colleague) investigated the incremental subgraph isomorphism. We show that the problem is intractable and unbounded for unit updates and path patterns. Nevertheless, we developed a heuristic algorithm to tackle the problem.

I experimentally verified the efficiency of these incremental algorithms, and showed that these incremental algorithms significantly outperform their batch counterparts in response to small changes (up to 25 %), using both real-life data (*e.g.,* Youtube, Citation network) and synthetic data.

This work is a first step towards the incremental graph pattern matching computation. As far as I know, little has been done for the incremental graph pattern matching based on these metrics; on the other hand, the above work provides an incremental computation scheme for the general graph pattern matching problem.

Web Service Aggregation and Graph Pattern Matching

The graph matching techniques has been widely used in a variety of application areas. However, not much has been done on the application of the graph pattern matching in the area of Web service composition and aggregation problem. Given a *synthesized Web service mediator* consists of template services (describing the tasks of a Web service) with an *aggregation function*, as well as a library of available *atomic services*, the *aggregation* problem finds for the mediator a best set of atomic services that realizes the mediator and maximizes (or minimizes) the aggregation function.

I (with three other colleagues) studied the aggregation problem for synthesized mediators of Web services, Specifically, (1) I have shown how to apply the graph pattern matching techniques as a preprocessing for an efficient initial selection phase of the valid atomic services for the mediator; and (2) we showed that the complexity of the general aggregation problem depends on the underlying graph structure of the mediator. We have shown several results of this kind, with matching lower bounds (NP and PSPACE), and analyzed restrictions that lead to polynomial-time solutions.

Future Works — A Research Agenda

My future research topics on the graph (pattern) matching and related problems, generally speaking, are two-fold as follows: (1) generalizing different graph (pattern) matching models into a unified model, and (2) developing practical graph querying and matching methods that work for special application areas. In addition, I intend to explore the efficient heuristic methods for web service aggregation among other related problems.

Developing general graph pattern matching models. In the near future, I am interested in the general graph matching models, for instance, the formulation of a group of graph models and queries that further extend *RQs* and *PQs* by supporting general regular expressions. One can easily verify that the simulation, the bounded simulation, *RQs* and *PQs* all are the special cases of this model. Nevertheless, with this comes increased complexity. Indeed, the containment and minimization problems become PSPACE-complete even for *RQs*. In particular, I have a keen interest in developing efficient (incremental and heuristic) algorithms for the general graph matching model. This topic is worth exploring not only for theoretical interests, but also for practical application, *e.g.*,, the designing of a uniform graph (pattern) matching scheme.

Identifying effective application domains. The general graph matching and querying model often come with a high complexity as remarked earlier. To this end I intend to identify more application domains in which simulation-based patterns and queries are most effective. For example, within the application over the social networks, some special properties may help in developing efficient and effective (incremental) graph pattern matching algorithms. In addition, I believe that novel incremental algorithms can be developed for the graph queries in particular applications, *e.g.,* traffic networks. Finally, as the performance and scalability of our graph (pattern) matching methods will remain key, I intend to compare our algorithms with more methods that claim to be efficient over large networks, *e.g.,* feature-based approaches relying on indexes for frequently occurred small subgraphs, namely, the features.

Web service aggregation. We have established several complexity bounds for the aggregation problem of the web service aggregation problem. Nevertheless, (1) the practical PTIME cases for the problem in certain specific application deserve a study; (2) it is interesting to revisit the composition problem when the aggregation synthesis is brought into the play, and (3) I intend to develop efficient heuristic algorithms to realize mediators with aggregation synthesis for specific applications.

My research will involve a combination of the theoretical study on the graph (pattern) matching and the practical application of the methods over various specified areas. For the first part, I intend to collaborate with my colleagues from the theoretical communities, to discover the all-matching complexity bounds for the (new) models; for the latter part, I intend to collaborate with Industry and research laboratories in understanding and developing the practical solutions for the problems. I believe that I will achieve this with my past experience of research in this field, the joint work done with my colleagues, and the communication and participation with the Industry. In one word, I am exciting at the research and development work involving these topics, and am keen to solve both theoretical and practical problems issued from the upcoming new challenging application areas.

References

[1] K. Bharat and A. Broder. Mirror, mirror on the Web: a study of host pairs with replicated content. *Comput. Netw.*, 31(11-16), 1999.

[2] J. Brynielsson, J. Högberg, L. Kaati, C. Martenson, and P. Svenson. Detecting social positions using simulation. In *ASONAM*, 2010.

[3] M. J. Brzozowski, T. Hogg, and G. Szabó. Friends and foes: ideological social networking. In *CHI*, 2008.

[4] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *TKDE*, 19(1), 2007.

[5] M. R. Henzinger, T. A. Henzinger, and P. W. Kopke. Computing simulations on finite and infinite graphs. In *FOCS*, 1995.

[6] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The Web as a graph: Measurements, models, and methods. In *COCOON*, 1999.

[7] A. Ntoulas, J. Cho, and C. Olston. What's new on the Web? The evolution of the Web from a search engine perspective. In *WWW*, 2004.

[8] G. Ramalingam and T. Reps. On the computational complexity of dynamic graph problems. *TCS*, 158(1-2), 1996.

[9] L. Terveen and D. W. McDonald. Social matching: A framework and research agenda. *ACM Trans. Comput.-Hum. Interact.*, 12(3), 2005.

[10] V. V. Vazirani. *Approximation Algorithms.* Springer, 2003.