Detailed Project Plan

0 Project Information

Project title: Theoretical Research on Online Matching (Gruop 1)

Project supervisor: Zhiyi Huang [1]

Group members: KinHei Kwok [2]

1 Introduction

Matching algorithms, or more specifically referred as **Graph Matching algorithms** under the context of Computer Science, has been a fundamental part of Graph theory due to its wide applications (typically related to resource allocation, such as ride sharing platforms with non-contracted drivers as in Uber (Huang et al., 2018a)). For its popularity in the scene, readers are assumed to have a basic understanding on the problem definition of online graph matching, and those who are unfamiliar with the literature are suggested to go through the Appendix A for a gentle introduction to the topic. While the topic has been around the scene ever since the founding of the Graph theory field, and solutions of the original problem (Dénes (1931), Philip (1935)) have been widely circulated, there are still many open problems to be answered for the *online* variant of the problem, and the field remains to be under active development.

1.1 Current Frontier of Research

The following topics of particular significance are shortlisted for readers to better understand the current focus of the HKU TCS Group regarding graph matching.

Online Bipartite Matching: This subclass of online matching problem restricts the input graph to be an instance of *bipartite graph*, denoted as $\mathrm{G(L, R, E)}$, where $\mathrm{L} \cup \mathrm{R}$ is the set of vertices, with L and R representing the offline and online vertices respectively. Under this setting, while all vertices $\mathrm{L} \cup \mathrm{R}$ are revealed to the algorithm on initialization, all edges $(u \in \mathrm{L}, v \in \mathrm{R})$ are only known to the algorithm on arrival of v, i.e. an online update indicating v's availability is received. Moreover, the algorithm is required to either irreversibly match it to some unmatched neighbor u (i.e. u will not be available for future arrivals of $v' \in R$), or to leave v unmatched forever. Since the algorithm is forced to make decisions without full knowledge of the graph, it is doomed to make sub-optimal decisions compared to its offline counterparts, and its performance is evaluated by comparing the output against

[1]Department of Computer Science, The University of Hong Kong, Email: Zhiyi@cs.hku.hk
[2]Department of Computer Science, The University of Hong Kong, Email: haleyk@hku.hk

the maximum matching under some benchmark, with the most trivial examples being **Maximum cardinality bipartite matching** and **Maximum weighted bipartite matching**. The choice of benchmark has gave rise to various open problems, so as the arguments/interpretation used by the model, such as the **Ranking algorithm** (Karp et al., 1990), and more notably the

Randomized Primal-Dual Framework (Devanur et al., 2013): While the technical details of the framework is beyond the scope of this project plan, on high level the framework also introduces a dual Linear Program formulation in contrast to the primal LP, which significantly reduces the complexity of analysis thanks to the rich literature and tools of dual LPs. This framework has been the foundation stone of many improvements and innovations in the field for the past few years, such as the

Fully Online Matching Model: Recently Huang et al. (2018a) has proposed a new model where both L and R are made online. To accommodate this change, each update could signify *either* the arrival or the departure (i.e. deadline) of the node v, and the algorithm need not to finalize its decision w.r.t. v immediately on its arrival, but any time before (or on) receiving its departure update. Note that the original model is a special case of this *de novo* model, where all online vertices' having arrival updates followed by departure updates immediately, and all offline vertices have their departure updates postponed indefinitely.

While the tentative title of this project "Theoretical Research on Online Matching" enjoys an extremely broad scope, it is expected that the work will be mostly related to the fully online (bipartite) matching model, due to the lack of room for improvement in the original model. On the other hand, the fully online matching model provides a lot of exciting opportunities, as its performance under various benchmarks, using different online algorithms are yet to be analyzed. I believe that this topic is a perfect candidate for my Final Year Project, as

- It promises room for work;
- I have studied all the prerequisites of this topic, namely dual LP and graph matching;
- It provides a hands-on experience on a challenging topic to prepare for my post graduate studies in the future;
- It is one of the most well-recognized fields in the TCS community.

1.2 Open Problems

Before the fully online model was introduced, the community was mainly working on the following benchmarks under the original model. These topics, sorted by my preferences non-decreasingly, are yet to be solved in the context of the fully online model (and also in the original model). The following settings are not mutually exclusive with each other and could be considered simultaneously.

Online Maximum Weighted Matching: As a generalized version of the original setting, there exists a gap between its approximation ratio and the unweighted result, under the assumption of random arrival order (See the discussion below). The performance gap between best known algorithms persists under the fully online matching model, and it would be interesting to investigate the impact of fully online matching model on the gap.

Random Arrival Order: In the context of **Algorithm complexity analysis**, worst-case analysis is usually used as default. Under the context of graph matching, most of the community's work benchmarks the algorithm's performances against the worst-case approximation guaranteed by the algorithm against any problem instance, which is referred as the "arbitrary arrival" analysis. This field has already saturated in the sense that the optimal result has been proven by Karp et al. (1990) decades ago, with research opportunities only open for new interpretations of the optimality proof (such as the randomized primal-dual framework). Meanwhile, in many practical use cases the underlying algorithm is transparent to the online party (e.g. ride sharing), and it may not be well-justified that the online party's arrival order is adapting aggressively against the algorithm's interest (e.g. University admission on a rolling basis). This has inspired the line of "random arrival" analysis which could be treated as an analogy of average-case analysis. Under this setting, the online party's arrival order is assumed to be shuffled uniformly at random, and the optimal result is yet to be found, but the current state-of-the-art result has been encouraging as it goes beyond the $1-1/e$ approximation ratio which is a natural bound imposed on many important Computer Science approximation problems (Huang et al., 2018b).

General Graph Matching: The general setting has been receiving a lot less attention compared with the bipartite setting, could be partly attributed to the difference between the complexity of their analysis. Yet, the paper which innovated the fully on(Huang et al., 2018a) proposed an approximate ratio that beats the state-of-the-art ratio in the original setting, which hints a possible stream of improvement with a mindset that reasons with the new model.

1.3 Tentative Schedule

It is forecasted that the project comes with a high risk of running into bottle-necks due to its challenging nature, and it is impossible to predict our research progress in the upcoming months, a tentative schedule of the project is proposed as in Table 1. Readers should be reminded that the following schedule is highly susceptible to changes.

Table 1: Project timetable

Date	Milestones	Course requirement?
29/09/2019	Deliverables of inception • Detailed project plan • Project web page	Yes
Late Oct / Early Nov	Complete background research • Scan through most of the works in the field • Scrutinize at least 3 of them of particular interest • Narrow down the project scope to a specific topic	No
Semester break	Conduct research based on the specific topic	No
13-17/01/2020	First presentation	Yes
Mid Jan	Mid-progress review • Report on preliminary progress • Decide if the targeted result is achievable • If not, justify the cause of impossibility in interim report and select another topic	No
02/02/2020	Deliverables of Elaboration • Preliminary implementation • Detailed interim report	Yes
Early April	Finalize the research results • Decide if the research result is successful enough for publishment	No
19/04/2020	Deliverables of Construction • Finalized tested implementation • Final report	Yes
20-24/04/2020	Final presentation	Yes
05/05/2020	Project exhibition	Yes
03/06/2020	Project competition	Top only

1.4 Deliverables

While the course expects the students to implement their projects to submit a tangible deliverable (highlighted in red in Table 1), our success on the topic is not guaranteed owing to the project's nature. The mid-progress review on mid January is expected to serve as a sanity check as we will have a better understanding in the problem's hardness, while having sufficient time ahead to turn around if needed. Either way, the project is expected to produce a thesis or a survey within the Online Matching field, which is either ready for publishment if the project is successful, or as a tangible proof of involvement in the project.

1.4.1 Project website

Other than this report, the inception stage also requires the student to submit a project website, which is hosted here on Haley Kwok's FYP account.

1.4.2 Software/Technologies used

Academic papers produced by this project should be formatted in LaTeXprofessionally. There are no other strict constraints posed on the other deliverables.

The project's website is developed with the Github Pages utility, using some Jekyll templates.

References

Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 101–107, Philadelphia, PA, USA, 2013. Society for Industrial and Applied Mathematics. ISBN 978-1-611972-51-1. URL http://dl.acm.org/citation.cfm?id=2627817.2627824.

Kőnig Dénes. Gráfok és mátrixok. *Matematikai és Fizikai Lapok*, 38, 1931.

Zhiyi Huang, Ning Kang, Zhihao Gavin Tang, Xiaowei Wu, Yuhao Zhang, and Xue Zhu. How to match when all vertices arrive online. *CoRR*, abs/1802.03905, 2018a. URL http://arxiv.org/abs/1802.03905.

Zhiyi Huang, Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. Online vertex-weighted bipartite matching: Beating 1-1/e with random arrivals. *CoRR*, abs/1804.07458, 2018b. URL http://arxiv.org/abs/1804.07458.

Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, 1990.

Hall Philip. On representatives of subsets. *J. London Math. Soc.*, 10:26–30, 1935. URL doi:10.1112/jlms/s1-10.37.26.

A Introduction to Graph Matching for non-specialists

This subsection is devoted to introducing the basics of graph matching algorithms.

Studies about graph matching usually revolves around the *maximum matching* for any arbitrary undirected graph supplied as input. The input Graph, denoted as G, composes of a set of Vertices V and (possibly weighted) Edges E, or more commonly written in a more concise way $\mathrm{G}(\mathrm{V},\mathrm{E})$. A Matching MATCHING w.r.t. G corresponds to a subset of of V where no two edges in E shares a vertex in MATCHING. Another way to interpret a matching is that all vertices $v \in M$ are **matched** to exactly one neighbour u, with u and v both being members of M. Figure. 1 gives some example matchings, with edges between the matched vertices highlighted in red.

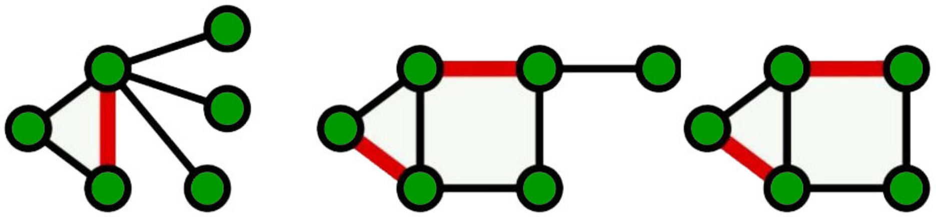

Figure 1: Original credits to Wikipedia , adapted by GeeksforGeeks

Recently, our research group has been focusing on studying the use cases where the input graph $\mathrm{G} = (\mathrm{V},\mathrm{E})$ is an instance of *bipartite graph*. The entire set of vertices of a bipartite graph could be partitioned into two independent sets denoted as L and R, such that $\mathrm{L} \cup \mathrm{R} = \mathrm{V}$, and no two vertices from the same set is connected by an edge (which could be also interpreted as every edge of E connects some member of L to a member of R). Unless otherwise specified, the bipartite graph is denoted as $\mathrm{G} = (\mathrm{L},\mathrm{R},\mathrm{E})$ to exemplify its membership. While our works are forced to give up some generality due to such specialization, this model is still capable in capturing many interesting use cases which follows the client-server pattern. On the other hand, this specialization is expected to give a large room for improvements as shown by previous works, e.g. their difference in the lowerbound under the weighted online setting Huang et al. (2018a), and famous results which are only applicable to bipartite graphs such as the Hall's marriage theorem (Philip, 1935) and the Koing's theorem (Dénes, 1931) for the offline variant.

B Outlined official guidelines

The official guideline for detailed project plan supplemented by the course's page is outlined as follows, for the grader's convenience.

- **Project background**
- **Project objective**
  Both of them has been sufficiently covered in Section 1.1, 1.2 and Appendix A
- **Project methodology**
  The final deliverable uses the fully online model with a high chance, as justified in Section 1.1
- **Project Schedule and Milestones**
  A detailed list of verifiable milestones is listed in Section 1.3.