How to write papers using LaTex

Chick*, Duck†
*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210024, China
†Department of Computer Science, City University of Hong Kong, Hong Kong

***Abstract*—This article describes how to use LaTex to write academic papers. It is written to help absolute beginners to gain a glimpse of how academic paper is organized using LaTex. Technical details and fancy tricks of LaTex will not be covered in this article(as I do not know any of them). Hopefully, this can serve as a template or maybe a reference when some of us set out to write a paper.**

I. Introduction

In Introduction, we introduce the background of our work, describe briefly the problems we discover and the contribution we make.

II. Related Work

In this section, we introduce related prior work regarding this paper's research topic. Usually, this section involves lots of citations. citation works like this: [1].

III. Problem and Statement

BGMDD(bipartite graph matching with dynamic duration) problem(Sec. III-A) and important concepts such as regret(Sec. III-B) are defined in this section. Table I lists the notation and basic definitions.

A. Problem Definition

***Definition** 1:* (Dynamic Bipartite Graph, DBG). *A dynamic bipartite graph is defined as $B = (L, R, E)$, where $L = \{i \in \mathbb{N}\}$ and $R = j \in \mathbb{N}$ are the sets of left and right nodes and $E \in L \times R$ is the set of edges between L and R. The node in L or R arrives independently from known probability distributions $P_L = \{p_l\}$ or $P_R = \{p_r\}$. Each node $i(j)$ arrives at time i(j), which is meaning to abuse the index to denote the nodes' arriving time. Each node has duration denoted by $i.d(j.d)$. If a node is not matched during the duration, the node would leave. Each edge $(i, j) \in E$ has a weight denoted by e_ij obeyed a distribution $P_E = \{p_e\}$.*

***Definition** 2:* (Matching Allocation). *A matching allocation over a dynamic bipartite graph B is denoted by $M = \{(i, j)|i \in L, j \in R\}$. It is a set of node pairs where each node appears at most once.The utility score of a matching allocation M over a dynamic bipartite graph B is measured by $U(B, M) = \{(i, j) \in Mw(i, j)\}$.*

In a real word, the nodes arriving distributions ($P_L = \{p_l\}$ $P_R = \{p_r\}$) and the edge weight distribution($P_E = \{p_e\}$) may change after some time. But it's easy to confirm the changes by collecting statistics. So in order to analysis convenient, we assume the distributions($P_L = \{p_l\}$, $P_R = \{p_r\}$ and $P_E = \{p_e\}$) are permanent. Alough by collecting statistics some changes could be measured, the change of duration of nodes can not be measured unless we doesn't match nodes and wait until the nodes leaving. It's impossible to do that to measure the changed of the duration.

***Definition** 3:* (Dynamic Duration Distribution.) *Each node's duration obey a distribution P_ld(or P_rd) independently. In this paper, we first analyze the situation that the durations of nodes in L and R obey the same distribution P_d, $P_d = P_ld = P_rd$. P_d can change after some time unknown. And we assume the type of the distribution should be pernament. The change of P_d is the decrease or increase of expectation of distribution. The length of the time in which P_d is stable should not be short.*

The change of duration distribution could be observed in many application scenarios. For example, in the food felivered scene, the customers' patient (which could be considered as duration) are always good in the early time,like 10:30 am (people are not very hungry), but very bad in1:30 am. The same situation would exist in online taxi-hailing service.

***Example** 3.1:* xxxxxxxx

***Definition** 4:* (BGMDD problem). *Give a dynamic bipartite graph with duration changed dynamic, the BGMDD problem is to find a matching allcation M to maximize the utility score in the online scenario.*

The BGMDD problem inherits and develops the two-sided online maximum bipartite matching problem in which the duration is given upon nodes' arrival. And it's also different from xxxx

B. MDP and Bandit Modeling for BGMDD problem

When the duration distribution is permanent, [X] states that the batch splitting way to solve DBG problem (dynamic bipartite graph) is a *Markov decision process (MDP)*. The current sets of left and right nodes and edges can be considered as state space, $\mathcal{S}$. The way to match the nodes, include matching or not matching and how to match, could be considered as action space, $\mathcal{A}$. Obviously, when P_R, P_L, P_e and P_d are pernament, the transition distribution, $T_r : (\mathcal{S} \times \mathcal{A} \times \mathcal{S})$, is deterministic. The utility score of matching is the reward, $R_w : (\mathcal{S} \times \mathcal{A} \to \mathbb{R})$. The target is to maxmine the cumulative score.

But when the duration distribution is dynamic, the analyzation should be changed. Because when P_d is dynamic, the the transition distibution is dynamic too. The process should not be considered as standard *MDP*. Instead, the total process can be considered as the combination of bandit problem and *standard MDP*. We can find that process during the stable time of P_d is *standard MDP* . So the problem translates into *multi-MDPs*. If we can measure the change of P_d, we just need to switch solving strategy to the corresponding *MDP*.

But as mention above, the changed of P_d can not be measured directly. So the problem becomes a bandit problem. We can use a "strategy" to decide how to match, and after a round(include hundreds of time interval) we need to calculate the total "reward"(cumulative score) and decide which strategy should be used for next round according to the history. So the strategys corresponding to different *standard MDPs* are considered as "arms" in bandit literature. The "bandit" property of BGMDD problem would be proved in sectionxx.

In conclusion, the process of BGMDD problem is *MDP* from micro aspect and a "bandit problem" in macro angle.

C. Evaluation Metric

1) Competitive Ratio: xxx

2) Regret: xxx

D. Greedy Algorithm

If we don't care about the total score or competitive ratio, almost all dynamic bipartite graph problem could be solved by Greedy algorithm. Greedy algorithm matches the nodes with the maximum edge weight, and don't match when no nodes can be matched.

Greedy algorithm don't use the duration actively, for it always matches nodes as soon as possible. So compared to other algorithms which use duration to wait for the better match, Greedy algorithm can't get the better result. But as compensation, Greedy algorithm is so stablely that the number of "losing node" which mean the nodes isn't matched during the their duration and leave may be less.

It seems that the score of Greedy algorithm is unrelated to duration. However their relevance are very strong. We would prove that there is a positive correlation between the score of Greedy algorithm and the duration(Section xxx).

***Property** 1:* Kittens are cute.

***Lemma** 1:* People love cute things.

Proof: Trivial. ∎

IV. Solution based on bandit with static-batch arms

In this section, we would describe the basic idea why we need the bandit algorithm to solve BGMDD problem and the detail about the algorithm based on bandit with static-batch arms. At first, a approach to turn P_d into a const would be introduced to make the following analysis easier. Then, the batch way to slove DBG probelm would be introduced and the bound of the score expectation using a specific static batch would be calculated. Next, we would state that it's a bandit problem when we use batch way to solve BGMDD problem and the bandit algorithm we use in this paper, discounted UCB, would be introduced. Finally, the theoretical performance of above algorithms in BGMDD problem would be anaylze.

A. Expectation Approximate

As the statement in section xx, the durations of nodes always obey some distribution in real application scenarios. But it's difficult to anaylze the performance of alforithms when the duration is a distribution. So we abuse the expectation of the distribution, E_d, to replace P_d in the following anaylsis. When the duration is a constant,i.e. E_d, we use T_e to represent the total score. When the duration is a distribution, i.e. P_d, we use T_p to represent the total score. Then we can get the difference when we use E_d to represent P_d in the anaylsis in **Theorem 1**.

***Theorem** 1:* When the distribution is limited(like 99% value in distribution is limited), we can find the max value and min value in P_d, denoted by D_{max} and D_{min}. When duration is D_{max} or D_{min}, the total score is represented by $T_{D_{max}}$ or $T_{D_{min}}$. The difference between T_d and T_e can be no higher then the difference between $T_{D_{max}}$ and $T_{D_{min}}$. Which is meaning that for any giving distribution P_d if we can find D_max and D_min we have:

$$|T_d - T_e| \leq |T_{D_{max}} - T_{D_{min}}|$$

Proof: It's obviously that we need make P_R, P_L, P_E and the way to solve BGMDD problem keep the same when the duration is changed. In this situation, when the duration is bigger, the total score must be bigger. And Considered that E_d is the expectation of P_d, So we have the following three inequations:

$$T_{D_{min}} \leq T_d \leq T_{D_{max}}$$

$$D_{min} \leq E_d \leq D_{max}$$

$$T_{D_{min}} \leq T_e \leq T_{D_{max}}$$

Make the first inequation subtract the third inequation, we have:

$$T_{D_{min}} - T_{D_{max}} \leq T_d - T_e \leq T_{D_{max}} - T_{D_{min}}$$

Rearranging, we obtain that

$$|T_d - T_e| \leq |T_{D_{max}} - T_{D_{min}}|$$

∎

As we don't consider the property of specific distribution, the bound is actually very loose. For some special dirtribution, like normal distribution, D_{max} could be replaced by the value bigger then the 95% values in distribution. And so as the D_{min}.

According to **Theorem 1**, the difference between T_d and T_e depends on the difference between $T_{D_{max}}$ and $T_{D_{min}}$. We would prove that when the values of D_{max} and D_{min} is much bigger then the difference between D_{max} and D_{min} the difference between $T_{D_{max}}$ and $T_{D_{min}}$ would be small in section xxx. Under this situation, we can replace the distribution P_d by its expection E_d to analyze the process.

B. Batch-based method

Batch-based method is to divide DBG probelm into a batch partition problem. The main idea of batch method is to wait for a specific duration(the length of batch) and then match the existing nodes in the graph by Hungarian algorithm[x]. It's a very simple and valid method to solve DBG problem, because when the node can wait for the better, batch method can accumulate enough nodes to make a better match.

It's obvious that the score of a match at the end of a batch duration depends on how much the nodes exsit in current graph. When the number of nodes is bigger, the score is bigger.

***Theorem** 2:* Assume $e_1, e_2, ..., e_n$ are n independent identically distributed variables obeying P_E. Then we consider the expection of max value among $e_1, e_2, ..., e_n$ as $X(n)$ which means $X(n) = E(max(e_1, e_2, ..., e_n))$. When P_E and n are specified, $X(n)$ is determinated. And when n is increasing, $X(n)$ is increasing which means the $X(n)$ is monotone increasing with n. When the number of left nodes is L and the number of right number is R at the matching moment, we have the bound of score expection:

$$\sum_{i=1}^{n} X(m-i) \leq E(T(L,R)) \leq nX(m)$$

$$n = min(L,R), m = max(L,R)$$

where $T(L,R)$ means the total score of this matching when the number of left nodes is L and the number of right number is R.

Proof: For convenience, we assume that L is smaller than R. Then, we focus on left nodes, because there would be no left nodes left after matching when L is smaller than R. In this situation, each left node has R choices to match, but not all left node can choose the best choice because some best choices maybe correspond to the same right node. So the best situation is every left node's best choice is different from others, which means the expection of every left node's matching score is $X(R)$. So the total score is $L \times X(R)$ in the best situation.

Now, consider the worst situstion. We first choose the largest weight edge and match corresponding nodes. Next, find another left node and we find that its best choice is matched, so its choice just $R-1$ left. So the node need to choose from the $R-1$ choice. Then the next left node come across again, it's equivalent that the node just have $R-2$ choice to choose at first. The situation happens again and again. So the expection of total score in this situation is $\sum_{i=1}^{L} X(R-i)$.

The worst situation could hardly happen if we use Hungarian algorithm to match the graph.xxxx So it's just the lower bound.xxxx ■

C. Optimal static batch duration at specific E_d

The simplest batch based method to solve DBG problem is to match all exsiting nodes in the graph using Hungarian algorithm periodically. Intuitively, under a specific E_d, the length of matching period would affect the total score.

According to *Theorem 2*, the total score in a matching depends on L and R.

Algorithm 1 Put your caption here

```
procedure PROC(a, b)                          ▷ This is an example
    System Initialization
    Read the value
    if condition = True then
        Do this
        if Condition ≥ 1 then
            Do that
        else if Condition ≠ 5 then
            Do another
            Do that as well
        else
            Do otherwise
    while something ≠ 0 do ▷ put some comments here
        var1 ← var2                           ▷ another comment
        var3 ← var4
```

V. Evaluation

Since evaluation section is where figures and tables appear the most, I put examples of inserting figures and tables here, but they can be used elsewhere.

A. figures

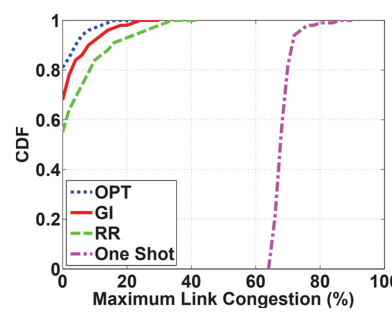
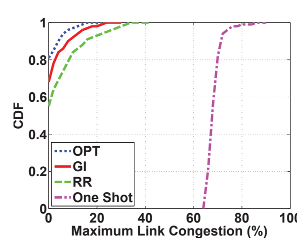
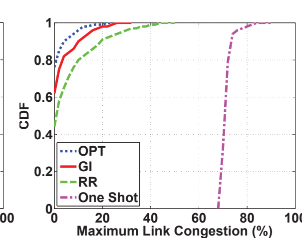


Fig. 1. insert one figure



(a) fig1



(b) fig2

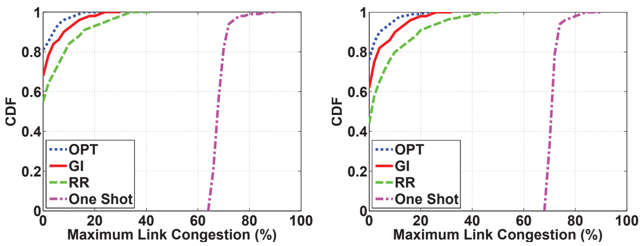
Fig. 2. put two figures together horizontally

B. tables

TABLE I
Running time for finding congestion-free update plans

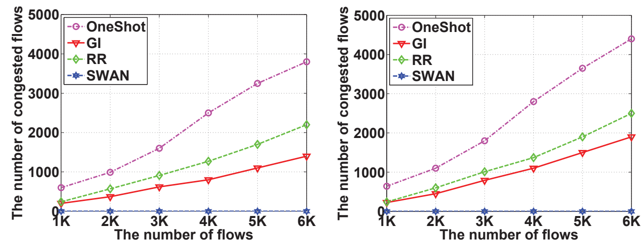
	1K	2K	3K	4K	5K
DCN	0.73 min	1.40 min	2.10 min	2.96 min	4.12 min
WAN	0.60 min	1.01 min	1.57 min	2.43 min	3.12 min

write something to explain your table in here



(a) DCN scenario (b) WAN scenario

Fig. 3. Maximum link congestion comparison.



(a) DCN scenario (b) WAN scenario

Fig. 4. The number of congested flows.

VI. Conclusion

In this paper, we list the basic component of an academic paper and show how they are organized using LaTex.

VII. Acknowledgement

We thank the anonymous reviewers for their helpful comments on draft of this paper. The work is partly supported by XX project(maybe not).

References

[1] D. Adams and R. T. Davies. *The hitchhikers guide to galaxy*. Pan Books, 2009.

Appendix A
Appendix section