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: [**?**].

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.

***Theorem** 1:* People love kittens.

Proof: Trivial. ■

IV. Solution based on bandit with static-batch arms

A. Expectation Approximate

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

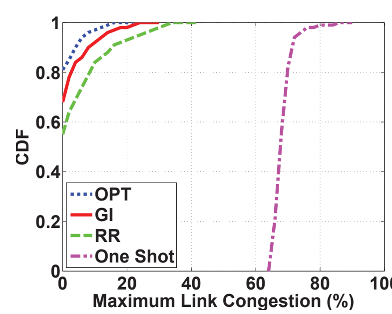
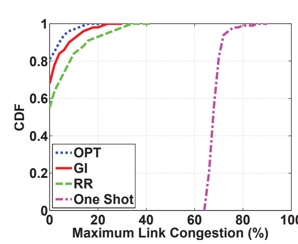
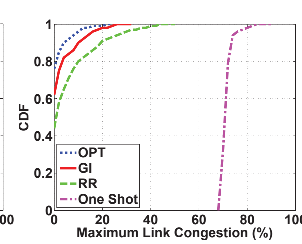


Fig. 1. insert one figure

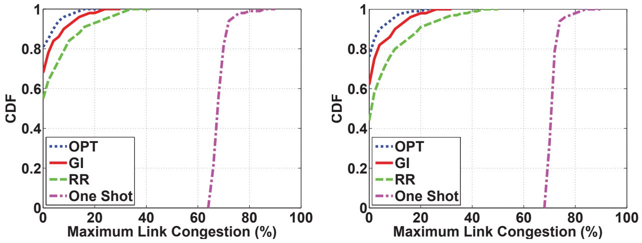


(a) fig1



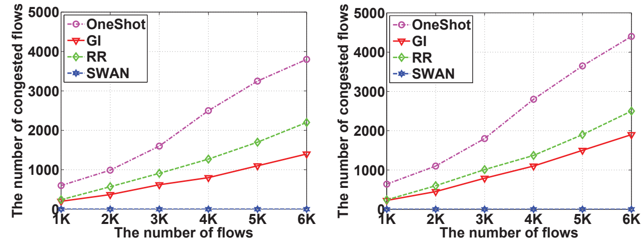
(b) fig2

Fig. 2. put two figures together horizontally



(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.

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

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