Colored Packets with Deadlines and Metric Space Transition Cost

This thesis was submitted in partial fulfillment of the requirements for the master degree (M.Sc.) at the Tel-Aviv University School of Computer Science

Submitted by
Adi Vardi

This work was supervised by
Prof. Yossi Azar

April 2013
Acknowledgements

I would like to thank my advisor, Prof. Yossi Azar for his guidance, the continuous support, patience and kindness. Despite his busy schedule, he always found the time to meet with me. I could not have imagined a better advisor.

My parents, Shlomo and Galia have been a constant source of support. Without their love and devotion this work could not have been accomplished.
Abstract

We consider scheduling of colored packets with transition costs which form a general metric space. We design $1 - O\left(\sqrt{\frac{\text{MST}(G)}{L}}\right)$ competitive algorithm. Our main result is an hardness result of $1 - \Omega\left(\sqrt{\frac{\text{MST}(G)}{L}}\right)$ which matches the competitive ratio of the algorithm for each metric space separately. In particular we improve the hardness result of Azar at el. for uniform metric space. We also extend our result for a weighted directed graph with triangular inequality and show $1 - O\left(\sqrt{\frac{\text{TSP}(G)}{L}}\right)$ competitive algorithm and nearly matching hardness result. In proving our hardness results we use some interesting non-standard embedding.
# Contents

1 Introduction .................................................. 2
   1.1 Our results ............................................. 2
   1.2 Related work ............................................ 4
   1.3 Structure of the paper ................................... 5

2 The Model .................................................... 6

3 Hardness Results for General Metric Space and Weighted Directed Graph 7
   3.1 Hardness Result for Star Metric ....................... 7
   3.2 Hardness Result for a general Metric Space .......... 14
   3.3 Hardness Result for Directed Graphs .................. 18

4 Online Scheduling Algorithm for General Metric Space and Directed Weighted Graph 23
   4.1 The algorithm .......................................... 23
1 Introduction

One of the most fundamental problem in competitive analysis is the packets scheduling problem. In this problem we are given a sequence of incoming packets. Each packet is unit size and has a deadline. The goal is to find a schedule that maximize number of packets that were transmitted before deadline expiration. The earliest deadline first (EDF) strategy is known to achieve optimal throughput for this problem. This paper explores a more general problem - maximizing colored packets throughput. In our model a switch has m incoming port and one output port. Each incoming port is associated with unbounded buffer. At each time unit new packets arrive to the queues, each packet has a deadline. The switch maintains current incoming port on which pending packets can be transmitted. There is a reconfiguration overhead when the switch changes the current incoming port. The goal is to maximize the number of packets that were transmitted before deadline expiration. We view the reconfiguration overhead as a metric space (and sometimes even as a weighted directed graph). In particular this problem generalizes the problem presented in [6], where the reconfiguration overhead is uniform. In this paper we characterize when it is possible to achieve $1 - o(1)$ competitive ratio and when it is not. This is done as a function of the specific metric space (or weighted directed graph) and the laxity (the minimum difference between the expiration and arrival time of packets). The ideal online competitive ratio of $1 - o(1)$ is quite rare. This is one example that this can be achieved.

1.1 Our results

Let $C$ denote the number of different packet colors. Let $L = \min_{i \in \sigma} \{d_i - r_i\}$ denote the minimum laxity of the packets. Also, let $\text{MST}(G)$ be the weight of the minimum spanning tree (MST) of a graph $G$. Our results are as follows.

- For general metric space we design an algorithm with competitive ratio of $1 - O\left(\sqrt{\frac{\text{MST}(G)}{L}}\right)$.

- We show a tight hardness result of $1 - \Omega\left(\sqrt{\frac{\text{MST}(G)}{L}}\right)$ for each metric space separately. Note that for the uniform metric space our result improves the hardness result of [6]. Specifically we improve their $1 - \Omega\left(\frac{C}{T}\right)$ hardness result to $1 - \Omega\left(\sqrt{\frac{C}{T}}\right)$ and matches their algorithmic result for uniform metric space.
We also consider the more general case where the transition costs form an weighted directed graph with triangular inequality. Let $TSP(G)$ denote the weight of the minimal Traveling Salesperson Problem (TSP) in graph $G$ (Calculating the accurate size is known to be NP-Complete, but since all the graphs in our model satisfy the triangular inequality we can approximate $TSP(G)$). Here we show nearly tight bounds for each graph. Specifically:

- For any weighted directed graph $G$ we design an algorithm with competitive ratio of 
  \[ 1 - O\left(\sqrt{\frac{TSP(G)}{L}}\right) \].

- For any weighted directed graph $G$ we prove an hardness result of 
  \[ 1 - \Omega\left(\sqrt{\frac{TSP(G)}{L \log C}}\right) \].

Note that for a metric space $MST(G) \leq TSP(G) < 2MST(G)$ and hence the algorithmic result of metric space and directed graph are equivalent. Note that for directed graph $TSP(G)$ and directed $MST(G)$ are not within a constant factor. The directed MST depends on the chosen root of the tree, but even the ratio between TSP to the maximum over all roots of directed MST may be as large as $\Theta(n)$ [2]. Fortunately the TSP in directed graph can be approximate up to a logarithmic factor ($O(\log n / \log \log n)$) [27, 12, 23, 5]

Usually it is enough to show hardness result on a specific metric space in order to prove that an algorithmic result is tight. In this paper we show results for each metric space (weighted directed graph) separately, and the hardness result and algorithmic result are different for each metric space (weighted directed graph).

In order to prove the hardness result we first prove it to a star metric and then for general metric space. One of the technique used in proving the hardness results for general metric space is showing the existence of an interesting embedding from any metric space $G$ on nodes $V$ to a metric star $S$ whose leaves correspond to $V$. The embedding satisfies the following requirements, given some fixed node $v_0$:

- The weight of $S$ is equal to the weight of MST in $G$.

- The weight of every Steiner tree in $S$ that contains $v_0$ is not larger then the Steiner tree on the same nodes in $G$.

This embedding is different than the usual embedding since we do not refer specifically to distances between vertices. Typically embedding is used to prove algorithmic result by simplifying the metric
space. In contrast, our embedding is used to prove an hardness result.

1.2 Related work

In the bounded delay model [29] we are given a sequence of incoming packets. Each packet is characterized by a value and a deadline. The goal is to find a schedule that maximize number of packets that were transmitted before deadline expiration. The earliest deadline first (EDF) strategy is known to achieve optimal throughput when all the packets have the same value. For arbitrary values the following results are known. Deterministic algorithmic result of about 0.547 [20, 35] and hardness result of $1/\phi \approx 0.618$ [26, 15, 4]. Randomized algorithmic result of $1 - 1/e \approx 0.632$ [10, 14] and hardness result of 0.8 [15]. The closet model to our model is the colored packets with deadline problem [6]. In this model we are given a sequence of incoming packets. Each packet is characterized by a color and a deadline (all packets have the same value). The goal is to find a schedule that maximize number of packets that were transmitted before deadline expiration, such that there is a transition time-slot between the transmission of packets from different colors. An algorithmic result of $1 - O\left(\sqrt{\frac{C}{L}}\right)$ and hardness result of $1 - \Omega\left(\frac{C}{L}\right)$ are shown in [6], when C is the number of colors and L is the minimum difference between expiration and arrival time of packets. Additional papers related to the bounded delay model are [34, 16, 11].

In the FIFO-queue model [29] we are given an a sequence of incoming packets. The packets are placed in a FIFO-queue with bounded buffer size. The challenges for an online algorithm are to decide weather to accept or discard arriving packet, and to choose a queue for transmission in each time unit. The problem was studied in [3, 7, 1, 30, 39, 20, 22]. Another related problem is the sorting buffer problem [38]. In this problem we are given a server with unbounded capacity and an incoming sequence of requests. Each request is correspond to a point in a metric space. The goal is to serve all request while minimizing total distance travel by the server. This problem can be interpreted as a multi-port device problem. This model was studied in [19, 33, 18, 32, 31, 8, 24, 17].

In proving the hardness result we use non-standard embedding. Embedding have been studied extensively over the years. the typical goal is to approximate complicated metric space by a "simpler” metric space such that the distances in the target metric space are no smaller than those in the original metric space and the maximum stretch (also called distortion) is bounded. The importance of this problem lies on the fact distances are a key issue in many approximation and
online problems. Bourgain [13] proved that any n-point metric space can be embedded in \( \ell_2 \) with distortion \( O(\log n) \). Rao [37] investigate the special case of embedding planner-graphs. He proved that any n-point planner-graph metric can be embedded in \( \ell_2 \) with distortion \( O(\sqrt{\log n}) \). Much effort was invested in embedding metric graph into tree metric. In order to use such a simple target metric we have to use randomization. Specifically there are simple examples of graphs for which the distortion using deterministic embedding is at least \( \Omega(n) \) [36, 25, 9]. Karp [28] was the first to suggest the probabilistic metric. Bartal [9] formally defined probabilistic embedding and proved that any probabilistic embedding of an expander graph into a tree has a distortion of at least \( \Omega(\log n) \). He also proved polylog\( (n) \) distortion for general metric space. Finally Fakcharoenphol et al. [21] showed that any n-point metric space can be embedded into a distribution over dominating tree metrics with distortion \( O(\log n) \).

1.3 Structure of the paper

In section 2 we describe the model. In section 3.1 we prove a tight hardness result for a star metric. In section 3.2 we prove a tight hardness result for a general metric space. In addition we show the existence of an non-standard embedding from any metric space to a star metric. In section 3.3 we prove a near tight hardness result for any weighted directed graph. In section 4 we describe the online \((1 - o(1))\)-competitive algorithm for our problem and its analysis.
2 The Model

We formally model the problem as follows. When the switch changes the current incoming port from \( j \) to \( k \), the reconfiguration overhead is \( w(j,k) \) time-slots for \( j \neq k \). Clearly \( w(j,k) = 0 \) for \( j = k \). We view port also as colors. We are given online sequence of packets \( \sigma \). Each packet is characterized by a triplet \( (r_i, d_i, c_i) \), where \( r_i \in \mathbb{N}_+ \) and \( d_i \in \mathbb{N}_+ \) are the respective arrive time and deadline time of the packet, and \( c_i \) is the color. The goal is to find a schedule that maximizes the number of transmitted packets, and satisfy the following conditions:

- In each time-slot either we transmit a packet, or we are in color transition phase, or this is an idle time-slot.
- Every scheduled packet \( i \) is transmitted between time unit \( r_i \) and time unit \( d_i \). Otherwise the packet is dropped.
- Between the transmission of packet with color \( j \) and a successive packet with color \( k \) there are a \( w(j,k) \) time-slots dedicated to color transition for \( j \neq k \).

We view \( w(j,k) \) as a directed graph. If we wish to change color from \( i \) to \( j \), we may first change the color from \( i \) to \( k \) and then from \( k \) to \( j \). Consequently each graph satisfy the triangular inequality.

- Typically \( w(j,k) = w(k,j) \) and hence the graph becomes undirected and can be viewed as a metric space.
- An interesting special case is when \( w(j,k) = 1 \) for \( j \neq k \), i.e. the uniform metric space. This case has been considered in [6].
- A special case of the general metric space case is a star metric (which is a generalization of uniform metric). In a star metric, transition from color \( i \) to color \( j \) require \( w_i + w_j \) time-slots (when \( w_i \) denote the weight of the edge into node \( i \)). This is equivalent to the case when the transition time to color \( j' \) is \( w_{j'} \).

Let \( \text{ALG}(\sigma) \) (\( \text{OPT}(\sigma) \)) denote the throughput of the online (optimal) schedule with respect to a sequence \( \sigma \).


3 Hardness Results for General Metric Space and Weighted Directed Graph

3.1 Hardness Result for Star Metric

In this section we consider the case where the transition time between colors is represented by a star metric. This is also equivalent to the case where the transition time to color $i$ is $w_i$. It generalizes the model presented in [6], where all transition time are the same (in particular 1). We prove that it is not possible to achieve a competitive ratio of $1 - o(1)$ when the weight of the star metric (i.e the sum of the weights of the edges of the star) is asymptotically larger than the minimum laxity of the packets. Applying the result to the uniform model improves the hardness result presented in [6] and proves that the BG algorithm from [6] is asymptotically optimal. The general idea is that the adversary creates packets with large deadline at each time unit, and blocks of packets with close deadline. The number of packets that arrive during a block is significantly smaller than the number of time-slots in the block. Any online algorithm must choose between 3 options:

- Transmit both packets with large deadline and packets with close deadline in many blocks. In this case there will be many packets with large deadline in the sequence, and the optimal schedule can avoid colors switching.

- Transmit mostly packets with close deadline. In this case there will be many packets with large deadline in the sequence, and the online algorithm loses idle time.

- Transmit mostly packets with large deadline. In this case there will be many packets with close deadline in the sequence. The online algorithm does not transmit all of them due to deadline expiration. The optimal schedule transmits all the packets with close deadline before transmitting the packets with large deadline and avoid deadline expiration.

Let $w(S)$ denote weight of a star metric $S$ (i.e the sum of the weights of the edges of $S$). It is clear that for a star metric $S$, $2w(S) = TSP(S)$. We define $F = \sqrt{w(S)L}$ for a star metric $S$.

Theorem 3.1. No deterministic or randomized online algorithm can achieve a competitive ratio better than $1 - \Omega\left(\frac{w(S)}{L}\right)$ in any given star metric $S$ when $L = \Omega(w(S))$. Otherwise, if $L = O(w(S))$ the bound becomes $\alpha < 1$ for some constant.
Proof. Let S be a given star metric with C nodes. We assume without loss of generality that 
$L = \Omega(w(S))$ since otherwise, one may use packets with laxity of $\Omega(w(S))$, and obtain an hardness 
result of $\alpha < 1$ for some constant. Let type A color denote color 0 and type B color denote colors 
$1...C - 1$. Let type A packet and type B packet denote packets with type A color and type B color 
respectively. Let $w_i$ denote the weight into color $i$.

We begin by describing the sequence $\sigma(S, ALG)$.

**sequence structure:** Type B Packets arrive in blocks of $3F$ time-slots, up to $N = \frac{1}{3} \sqrt{\frac{L}{w(S)}}$ blocks.
Let $t_i = 1 + 3(i - 1)F$ denote the beginning time of block $i$. For each block $i$, where $N \geq i \geq 1$, $F$ 
packets arrive at time $t_i + 2F$ of various colors. Specifically $\frac{w_c}{w(S) - w_0}F$ packets 
$(t_i + 2F, L + t_i + 2F, c)$, 
for each $1 \leq c \leq C - 1$ are released. Type A packet $(t, 3L, 0)$ is released at each time unit $t$ in each 
block. When block $i$ ends there are several possible cases:

1. If with probability at least $1/4$ there are at least $F/2$ untransmitted type B packets (denoted 
by condition 1). $L$ packets $(t_{i+1}, L + t_{i+1}, 1)$ are released and the sequence is terminated. 
Clearly $t_{i+1}$ is the final release time. See figure 2.

2. Else, if with probability at least $1/4$, at most $2F$ packets were transmitted during the block 
(denoted by condition 2). $3L$ packets $(t_{i+1}, 3L, 0)$ are released and the sequence is terminated 
Clearly $t_{i+1}$ is the final release time. See figure 3.

3. Else, if $i = N$ (there are $N$ blocks, none of them satisfied condition 1 or 2) $3L$ packets 
$(L + 1, 3L, 0)$ are released, and the sequence is terminated. Clearly $L + 1$ is the final release 
time. See figure 4.

4. Else ($i < N$) then block $i+1$ starts.

![Block's structure](image)

Figure 1: Block's structure. The pair $(r,d)$ represent release time $r$ and deadline $d$.

**observations:**

1. No type B packets arrive in the first $2F$ time units of a block nor in the last $F - 1$ time units 
of a block.
Figure 2: Sequence structure for case 1. See figure 1 for block’s structure.

Figure 3: Sequence structure for case 2.

Figure 4: Sequence structure for case 3. Recall that \( N = \frac{1}{3} \sqrt{\frac{L}{w(S)}} \).
2. Each block consists of $3F$ time-slots. Hence if ALG transmitted at most $2F$ packets during a block, there must have been at least $F$ idle time-slots,

3. There are up to $\frac{1}{3}\sqrt{\frac{L}{w(S)}}$ blocks and each block consists of $3\sqrt{w(S)L}$ time-slots. It is clear that the final release time is at most $L + 1$.

4. Exactly one type A packet arrive at each time slot until the final release time. Therefore up to $L$ type A packets arrive before (not including) the final release time.

5. During each block exactly $F$ type B packets arrive. It is clear that at most $L/3$ type B packets arrive before (not including) the final release time.

Now we can analyze the competitive ratio of $\sigma(S, \text{ALG})$. Consider the following possible sequences (according to the termination type):

1. Case 1: Let $Y$ denote the number of packets in the sequence. According to the observations the sequence consists of at most $L$ type A packets, and at most $\frac{4}{3}L$ type B packets ($L/3$ until the final release time and $L$ at the final release time). Hence $Y \leq L + \frac{4}{3}L \leq 3L$. At time $t_{i+1}$ there is a probability of at least $1/4$ that ALG has $L + F/2$ untransmitted type B packets. Since type B packets have laxity of $L$, ALG can transmits at most $L + 1$ of them, and must drop at least $F/2 - 1$. The expected number of transmitted packets is:

$$E(\text{ALG}(\sigma)) \leq Y - \frac{1}{4}(F/2 - 1) = Y - F/8 + 1/4.$$ 

We describe an offline algorithm OPT' as follows. OPT' transmits type B packets first, and then transmits type A packets.

- Recall that all type B packets in a block arrive at once after $2F$ time units. Let $j_k$, $1 \leq k \leq C$ be the order of colors in the minimal TSP. In each block OPT' transmits the type B packets according to that order - first all packet with color $j_1$, then all the packet with color $j_2$ and so on. It is clear that OPT' needs at most $F + 2w(S)$ time-slots to transmits the packets ($F$ for packet transmission and $2w(S)$ for color transition). OPT' transmits the packets from their arrival time and until the $2w(S)$ time-slots of the next block. Recall that $L = \Omega(w(S))$ and that $F = \sqrt{w(S)L}$. We conclude that
2F > 2w(S). Hence there is no overlap between the transmission of type B packets from different blocks. In the last block OPT' missed 2w(S) packets (since there are no additional blocks).

- The L packets (t_{i+1}, L + t_{i+1}, 1) arrived during the final release time are transmitted by OPT' consequently from time t_{i+1}. OPT’ can transmit L packets except for one transition phase, and hence may lose at most 2w(S) packets. According to the observations the final release time t_{i+1} is at most L + 1. We conclude that OPT’ transmits all type B packets until time unit 2L.

- OPT’ transmits the L type A packets consequently from time unit 2L + 1. Since the deadlines are 3L, OPT’ transmits all type A packets.

We conclude that: OPT(σ) ≥ OPT’(σ) ≥ Y − 4w(S), and the competitive ratio is:

\[
E(\text{ALG}(\sigma)) \leq \frac{Y - F/8 + 1/4}{Y - 4w(S)} \leq \frac{3L - \frac{1}{8} \left(\sqrt{w(S)L} + 1/4\right)}{3L - 4w(S)} = 1 - \Omega\left(\sqrt{\frac{w(S)}{L}}\right).
\]

2. Case 2: The sequence consists of more than 3L type A packets, all deadlines are at most 3L.

The probability that ALG was idle during F time-slots is at least 1/4. Hence the expected number of transmitted packets is:

\[
E(\text{ALG}(\sigma)) \leq 3L - F/4.
\]

At each time unit until the final release time, OPT’ transmits the type A packet that arrived at the same time unit. Consequently from the final release time and until time unit 3L, OPT’ transmits the type A packets that arrived at the final release time. Therefore OPT’ transmits 3L type A packets. Hence the competitive ratio is:

\[
E(\text{ALG}(\sigma)) \leq \frac{3L - F/4}{3L} = 1 - \Omega\left(\sqrt{\frac{w(S)}{L}}\right).
\]

3. Case 3: the sequence consists of 4L type A packets, all deadlines are at most 3L. Let U_i be the event that the number of untransmitted type B packets at the end of block i is less than F/2. If U_i occurs then let j_k, 1 ≤ k ≤ r be the type B colors transmitted by ALG in block i.
At least \( F/2 \) packets that arrived in this block has to be transmitted. Therefore:

\[
\frac{w_{j_1}}{w(S) - w_0} F + \frac{w_{j_2}}{w(S) - w_0} F + \cdots + \frac{w_{j_r}}{w(S) - w_0} F \geq F/2
\]

or

\[
w_{j_1} + w_{j_2} + \cdots + w_{j_r} \geq \frac{w(S) - w_0}{2}.
\]

Let \( E_i \) be the event that more than \( 2F \) packets are transmitted during block \( i \). If event \( U_{i-1} \) and \( E_i \) occurs then during the first \( 2F \) time-slots of block \( i \) there are at most \( F/2 \) untransmitted type B packet (recall that all type B packets in a block arrive at once after \( 2F \) time units), and more than \( 2F \) packets were transmitted during the block. Since each block consist of \( 3F \) time-slots, at least \( F \) packets were transmitted during the first \( 2F \) time slot. Therefore at least one type A packet was transmitted during the first \( 2F \) time-slots.

Combining the results, if \( U_i, U_{i-1} \) and \( E_i \) occurs then:

- During block \( i \) at least \( \frac{w(S) - w_0}{2} \) time-slots used for type B color transition.
- Type A packet was transmitted during the first \( 2F \) time-slots of the block.
- Type B packet was transmitted during the last \( F \) time-slots of block \( i - 1 \) (if exist).

From the last two bullets we conclude that for \( i \neq 1 \) there was a transition to type A color between the last \( F \) time-slots of block \( i - 1 \) and the first \( 2F \) time-slots of block \( i \). Since there is no overlap between those periods in different blocks, we do not count the same transition twice. We conclude that if \( U_i, U_{i-1} \) and \( E_i \) occurs and \( i \neq 1 \), \( \frac{w(S) - w_0}{2} + w_0 \geq w(S)/2 \) time-slots used for color transitions. Since none of the blocks satisfy condition 1 or 2, for all \( i \) s.t. \( \frac{L}{3w(S)} \geq i \geq 2 \) we have:

\[
Pr[U_i] \geq 3/4, Pr[U_{i-1}] \geq 3/4, Pr[E_i] \geq 3/4.
\]
Therefore:

\[
Pr[U_i \cap U_{i-1} \cap E_i] = 1 - Pr[\neg(U_i \cap U_{i-1} \cap E_i)] \\
= 1 - Pr[\neg U_i \cup \neg U_{i-1} \cup \neg E_i] \\
\geq 1 - 1/4 - 1/4 - 1/4 = 1/4 .
\]

The sequence consists of \( \frac{1}{3}\sqrt{\frac{L}{w(S)}} \) blocks, \( \frac{1}{3}\sqrt{\frac{L}{w(S)}} - 1 \) if we exclude the first block. In each block there is a probability of at least 1/4 that \( w(S)/2 \) time-slots used for color transition. Hence the expected number of lost packets in each block is at least \( \frac{1}{4} \frac{w(S)}{2} \). We conclude that the expected number of transmitted packets is:

\[
E(\text{ALG}(\sigma)) \leq 3L - \left( \frac{1}{3}\sqrt{\frac{L}{w(S)}} - 1 \right) \frac{w(S)}{2} \frac{1}{4} = 3L - \frac{1}{24} \left( \sqrt{w(S)L} \right) + \frac{w(S)}{8}.
\]

At each time unit until the final release time, OPT’ transmits the type A packet that arrived at the same time unit. Consequently from the final release time and until time unit \( 3L \), OPT’ transmits the type A packets that arrived at the final release time. Therefore OPT’ transmits \( 3L \) type A packets. Hence the competitive ratio is:

\[
\frac{E(\text{ALG}(\sigma))}{\text{OPT}(\sigma)} \leq 3L - \frac{1}{24} \left( \sqrt{w(S)L} \right) + \frac{w(S)}{8} = 1 - \Omega \left( \sqrt{\frac{w(S)}{L}} \right).
\]

**Corollary 3.2.** No deterministic or randomized online algorithm can achieve a competitive ratio better than \( 1 - \Omega \left( \sqrt{\frac{C}{L}} \right) \) when all color transitions takes one unit of time and \( L = \Omega(C) \). Otherwise, if \( L = O(C) \) the bound becomes \( \alpha < 1 \) for some constant.

**Proof.** Let S be a star metric with \( C \) nodes such that the weight of each edge is equal to 1/2. Clearly each color transition requires one time unit and \( w(S) = C/2 \). Applying theorem 3.1 we obtain hardness result of \( 1 - \Omega \left( \sqrt{\frac{C}{L}} \right) \).

Remark: It is clear that when all color transition takes \( D \) units of time the hardness result is \( 1 - \Omega \left( \sqrt{\frac{DC}{L}} \right) \).
3.2 Hardness Result for a general Metric Space

In this section we consider the case where the transition time is represented by a metric space G. A natural approach is to reduce G to a less complex graph (e.g. a star metric) and use a similar sequence to \( \sigma(S,\text{ALG}) \) described in section 3.1. Note that hardness result on sub graph is not an hardness result on a graph. For example an hardness result for MST of a metric space G is not an hardness result for G since the algorithm may use the additional edges to reduce the transition time. We begin by describing the requirements for embedding from G into a star metric S, such that using \( \sigma(S,\text{ALG}) \) for ALG on metric space G yields hardness result of \( 1 - \Omega\left(\sqrt{\frac{\text{MST}(G)}{L}}\right) \).

Then we prove that such embedding exists for any metric space G. By combining the results we conclude that the hardness result of \( 1 - \Omega\left(\sqrt{\frac{\text{MST}(G)}{L}}\right) \) holds for any metric space.

We begin by introducing some new definitions:

- We define \( w(T) = \sum_{e \in V} w(e) \) for a tree \( T = (V,E) \), and let \( P_T(v) \) denote the parent of node \( v \) in a rooted tree \( T \).

- We define \( w_S(V) = \sum_{v \in V} w(c,v) = \sum_{v_i \in V} w_i \) for a star metric S with a center c. It is clear that for a star S with leaves V \( w_S(V) = w(S) \).

- Let \( T_G(V) \) be the minimum weight connected component that contains the set V (i.e. the minimum Steiner tree on this points) in metric space G.

First we prove that the embedding exists for any metric space G.

**Theorem 3.3.** For any given metric space G on nodes V there exists an embedding \( f : G \rightarrow S \) from G to a star metric S with leaves V and a special leave \( v_0 \in V \) (f depends on \( v_0 \)) such that:

1. \( w(S) = T_G(V) (= \text{MST}(G)) \).

2. For every \( V' \subseteq V \) such that \( v_0 \in V' \), \( w(T_G(V')) \geq w_S(V') \).

**Proof.** Let G be a given metric space on nodes V. Let T be the MST for G created with Prim algorithm with a root \( v_0 \in V \). Let S be a star metric with leaves V such that for each \( u \in V, \ w_u = w(u, P_T(u)) \). Clearly \( w_{v_0} = 0 \). We prove that S and \( v_0 \) satisfy the lemma’s condition:

1. Clearly \( w(S) = w(T) \). Since T is a MST for G, \( w(S) = w(T) = \text{MST}(G) = T_G(V) \).
2. Consider $V' \subseteq V$ and $V' = \{v_0, v_1, \ldots, v_{r-1}\}$. Recall that we defined $w_u = w(u, P_T(u))$. Clearly $w_S(V') = \sum_{j=1}^{r-1} w(v_j, P_T(v_j))$. Hence it is sufficient to prove that $w(T_G(V')) \geq \sum_{j=1}^{r-1} w(v_j, P_T(v_j))$. Let $u_0 = v_0, u_1, \ldots, u_{r-1}$ be the order that Prim added $V'$ nodes to $T$. Let $E_P = \{e_1, \ldots, e_{r-1}\}$ be the corresponded edges ($u_0$ does not have a corresponding edge).

We run Prim from $v_0$ and perform the following modification:

- $T' \leftarrow T_G(V')$.
- Let $u_i$ be the next node added by Prim to the connected component and $e_i$ the corresponded edge.
  
  (a) If $u_i \notin V'$ then add $e_i$ and $u_i$ to $T'$.
  
  (b) Else, if $e_i \in E_P$ then do nothing
  
  (c) Else, if $e_i \notin E_P$ then
    
    i. Add $e_i$ to $T'$.
    
    ii. Let $C'$ be the cycle created by adding $e_i$ to $T'$. Let $e'$ be the edge with maximal weight on $C'$ such that $e' \cap \{u_0, \ldots, u_{i-1}\} \neq \emptyset$ and $e' \notin \{e_1, \ldots, e_i\}$ (i.e. $e'$ is the maximal among the edges in $C'$ that belong to $T_G(V')$, and one of their nodes was already added to Prim’s connected component. We later prove that such an edge always exists). Remove $e'$ from $T'$.

First we prove the correctness of the algorithm:

- After each step of the algorithm $T'$ is tree which contains $V'$:
  
  - Since $T_G(V')$ is a tree, $T'$ is a tree in the beginning of the algorithm.
  
  - Since Prim start from $v_0 \in V'$ and create a tree, adding Prim’s edge $e_i$ in case a keeps $T'$ connected and without cycles.
  
  - When node $u_i$ is added by Prim to the connected component in case c, it creates a cycle (adding edge to a connected component create a cycle). Therefore deleting any edge from this cycle keeps $T'$ connected and without cycles.

- When node $u_i$ is added by Prim in case c, there exists at least one edge $e'$ such that $e' \cap \{u_0, \ldots, u_{i-1}\} \neq \emptyset$ and $e' \notin \{e_1, \ldots, e_i\}$, because Prim does not create cycles.
Now we claim that the algorithm satisfy the following invariant: 

The sum of edges in $T'$ that contain nodes from $V'$ do not increase. To prove the claim we note that the sum changes only in case c (i.e when the algorithm add edge $e_i$ and delete edge $e'$ from $T'$). Since Prim added edge $e_i$ instead of $e'$, $w(e_i) \leq w(e')$ (Prim always choose the edge with the minimal weight).

Now we can prove the second property of the embedding. By definition of Prim $e_i = (u_i, P_T(u_i))$. Hence $\sum_{j=1}^{r-1} w(e_j) = \sum_{j=1}^{r-1} w(v_{ij}, P_T(v_{ij}))$. Let $e'_i$ be the edge deleted from $T'$ when edge $e_i$ was added (case c) or $e_i$ in case no edge was deleted (case b). We have:

$$w_S(V') = \sum_{j=1}^{r-1} w(v_{ij}, P_T(v_{ij})) = \sum_{j=1}^{r-1} w(e_j) \leq \sum_{j=1}^{r-1} w(e'_j) = w(T_G(V')).$$

when the first equality follows from definition, the first inequality result from the invariant and the last equality follows by definition.

Now we use the embedding to prove hardness result of $1 - \Omega\left(\sqrt{\frac{\text{MST}(G)}{L}}\right)$.

**Theorem 3.4.** No deterministic or randomized online algorithm can achieve a competitive ratio better than $1 - \Omega\left(\sqrt{\frac{\text{MST}(G)}{L}}\right)$ in any given metric space $G$ when $L = \Omega(\text{MST}(G))$. Otherwise, if $L = O(\text{MST}(G))$ the bound becomes $\alpha < 1$ for some constant.

**Proof.** Let $G$ be a given metric space on nodes $V$. We use the embedding from theorem 3.4. Let $S, v_0$ be the output of the embedding. Let $\sigma(S, \text{ALG})$ be the sequence described in theorem 3.1, when $v_0$ is type A color and the other colors are type B colors. Recall that by definition $F = \sqrt{w(S)L}$. We use $\sigma$ for ALG on $G$. We assume without loss of generality that $L = \Omega(\text{MST}(G))$ since otherwise, one may use packets with laxity of $\Omega(\text{MST}(G))$, and obtain an hardness result of $\alpha < 1$ for some constant. Consider the following possible cases, similar to the proof of theorem 3.1.

1. In the first case there exists a block $i$ such that with probability at least 1/4, at the end of the block there are at least $F/2$ untransmitted type B packets. In theorem 3.1 we proved that:

   - The sequence consists of up to $3L$ packets.
• The expected number of packets ALG missed is at least $F/8 - 1/4$.

• OPT missed up to $2TSP(G) \leq 4MST(G)$ packets.

Therefore the competitive ratio depends only on $F$, $MST(G)$ and $L$:

$$\frac{E(\text{ALG}(\sigma))}{\text{OPT}(\sigma)} \leq \frac{3L - F/8 + 1/4}{3L - 4MST(G)}$$

$$= \frac{3L - \frac{1}{8} \left( \sqrt{w(S)L} \right) + 1/4}{3L - 4MST(G)}$$

$$= \frac{3L - \frac{1}{8} \left( \sqrt{MST(G)L} \right) + 1/4}{3L - 4MST(G)} = 1 - \Omega \left( \sqrt{w(S)L} \right) \text{.}$$

When the second equality result from the fact that $w(S) = MST(G)$.

2. In the second case there exists a block $i$ such that with probability at least $1/4$ at most $2F$ packets were transmitted during the block. In theorem 3.1 we proved that:

• At most $3L$ packets can be transmitted.

• The expected number of packets ALG missed is at least $F/4$.

• OPT' transmitted $3L$ type A packets.

Therefore the competitive ratio depends only on $F$ and $L$:

$$\frac{E(\text{ALG}(\sigma))}{\text{OPT}(\sigma)} \leq \frac{3L - F/4}{3L}$$

$$= \frac{3L - \frac{1}{4} \left( \sqrt{w(S)L} \right)}{3L}$$

$$= 1 - \Omega \left( \sqrt{\frac{w(S)}{L}} \right) = 1 - \Omega \left( \sqrt{\frac{MST(G)}{L}} \right) \text{.}$$

3. In the third case ALG transmitted type A packet and at least $F/2$ type B packets at each block. In theorem 3.1 we proved that:

• At most $3L$ packets can be transmitted.

• The expected number of packets ALG missed in each block (excluding the first block) due to color transitions is at least $\frac{1}{4} w(S)/2$.
• OPT transmitted 3L type A packets.

By the first condition required by this lemma, each sequence of color transitions in G requires more transition time than in S. Therefore the expected number of packets ALG missed per block (excluding the first block) is at least \( \frac{1}{4} \frac{w(S)}{2} \). Since the number of blocks is \( \frac{L}{w(S)} \) we conclude that the competitive ratio is:

\[
\frac{E(\text{ALG}(\sigma))}{\text{OPT}(\sigma)} \leq 3L - \left( \frac{1}{3} \frac{L}{w(S)} - 1 \right) \frac{w(S)}{8} = 1 - \Omega \left( \frac{w(S)}{L} \right) = 1 - \Omega \left( \frac{\text{MST}(G)}{L} \right).
\]

### 3.3 Hardness Result for Directed Graphs

In this section we prove near tight hardness result of \( 1 - \Omega \left( \sqrt{\frac{\text{TSP}(G)}{L \log C}} \right) \) for directed graphs with triangular inequality. As in section 3.1 the sequence consist of blocks. Each block consist of phases. Each phase “forces” the online algorithm to transmits packets from at least 1/2 of the remaining type B colors. After \( \log C \) phases the online algorithm has to spend at least \( \text{TSP}(G) \) time-slots for color transition. This technique guarantee that the online algorithm lose enough packets which implies an hardness result of \( 1 - \Omega \left( \sqrt{\frac{\text{TSP}(G)}{L \log C}} \right) \).

**Theorem 3.5.** No deterministic online algorithm can achieve a competitive ratio better than \( 1 - \Omega \left( \sqrt{\frac{\text{TSP}(G)}{L \log C}} \right) \) in any given directed graph G with triangular inequality when \( L = \Omega(\text{TSP}(G) \log C) \). Otherwise, if \( L = O(\text{TSP}(G) \log C) \) the bound becomes \( 1 - \Omega(\frac{1}{\log C}) \).

**Proof.** We begin by describing the packet sequence \( \sigma(G, \text{ALG}) \).

**sequence structure:** We assume without loss of generality that \( L = \Omega(\text{TSP}(G) \log C) \) since otherwise, one may use packets with laxity of \( \Omega(\text{TSP}(G) \log C) \), and obtain an hardness result of \( 1 - \Omega(\frac{1}{\log C}) \). We define \( H = \sqrt{\frac{\text{TSP}(G)L}{\log C}} \), \( N = \frac{1}{5} \sqrt{\frac{L}{\text{TSP}(G) \log C}} \). Type B Packets arrive in blocks, up to \( N \) blocks. Type A packets \((t, 3L, 0)\) are released at each time unit \( t \) in each block. Let \( t_i \) denote the beginning time of block \( i \). The first block starts at time-slot 1 and blocks are consecutive. A block consist of phases: start phase, regular phases and possibly end phase. The start phase consist
of $2H \log C$ time-slots. Every block that does not satisfy condition ($I_1$) (defined later in the paper) has an end phase with $2H \log C$ time-slots. The number of regular phases is at least 1 and at most $\log C$ and depends on ALG’s behavior. Each regular phase consists of $H$ time-slots. We denote by $t_{i,j}$ the beginning time of regular phase $j$ in block $i$. First regular phase starts after the start phase, and regular phases are consecutive. At the beginning of the first regular phase $H$ packets arrive of various colors. Specifically $\frac{H}{C-1}$ packets $(t_{i,1}, L + t_{i,1}, c)$ for each $1 \leq c \leq C - 1$ are released. When regular phase $j$ in block $i$ ends there are several possible cases:

1. There are at least $H/2$ untransmitted type B packets. In this case (denoted by condition ($I_1$)) $L$ packets $(t_{i,j+1}, L + t_{i,j+1}, 1)$ are released and the sequence is terminated. Clearly $t_{i,j+1}$ is the final release time.

2. Else, during the regular phases packets from all type B colors were transmitted. We complete the block by an end phase, and consider the following cases:

   (a) If no type A packet was transmitted during the start phase or during the end phase (denoted by condition ($I_2$)). Then $3L$ packets $(t_{i+1,1}, 3L, 0)$ are released, and the sequence is terminated. Clearly $t_{i+1,1}$ is the final release time.

   (b) Else, if $i = N$ (there are $N$ blocks and none of them satisfied conditions ($I_1$) or ($I_2$)). Let $t$ be the time-slot consecutive to the end of the last block. Then $3L$ packets $(t, 3L, 0)$ are released, and the sequence is terminated. Clearly $t$ is the final release time.

   (c) Else ($i < N$) then block $i+1$ starts.

3. Else, let $c_k, 1 \leq k \leq r$ be the colors that were not transmitted during the regular phases. At the beginning of regular phase $j+1$, $H$ packets arrive of various colors. Specifically $\frac{H}{r}$ packets $(t_{i,j+1}, L + t_{i,j+1}, c_k)$ for $1 \leq k \leq r$ are released, and the condition is checked again.

**Lemma 3.6.** There are at most $\log C$ regular phases in each block.

**Proof.** During each regular phase at least half of the remaining type B colors (i.e. type B colors that were not transmitted during previous regular phases) are transmitted. It is clear that after at most $\log C$ regular phases all type B packets are transmitted. ■
First we make the following observations:

1. No type B packets arrive during the start phase nor in the end phase of a block.

2. Since there are at most \( \frac{1}{5} \sqrt{\frac{L}{TSP(G) \log C}} \) blocks, and each block consists of at most \( 5H \log C \) time-slots (\( 2H \log C \) during the start phase, \( 2H \log C \) during the end phase, up to \( H \log C \) during the phases stage), then the final release time is at most \( L + 1 \).

3. Exactly one type A packet arrives at each time-slot until the final release time. Therefore up to \( L \) type A packets arrive before (not including) the final release time.

4. In each block the number of type B packets released is at most \( \frac{1}{5} \) of its size (type B packets are released only during the regular phases). Hence there are at most \( \frac{L}{5} \) type B packets before (not including) the final release time.

Now we can analyze the competitive ratio of \( \sigma(G, ALG) \). Consider the following possible sequences (according to the termination type):

1. Case 1: Let \( Y \) denote the number of packets in the sequence. According to the observations the sequence consists of at most \( L \) type A packets, and at most \( \frac{6}{5} L \) type B packets (\( L/5 \) until the final release time and \( L \) at the final release time). Hence \( Y \leq L + \frac{6}{5} L \leq 3L \). At time \( t_{i,j+1} \) ALG has \( L + H/2 \) untransmitted type B packets. Since type B packets have laxity of \( L \), ALG can transmits at most \( L + 1 \) of them and drop at least \( H/2 - 1 \). The number of transmitted packets is:

\[
ALG(\sigma) \leq Y - H/2 + 1.
\]

We describe an offline algorithm OPT' as follows. OPT' transmits all type B packets first, and then transmits type A packets.

- Recall that no type B packets arrive during the end phase. Let graph \( G_{i,j} \) be the spanning subgraph of \( G \) that contains only the colors transmitted in regular phase \( j \) of block \( i \). Let \( c_{i,j,k}, 1 \leq k \leq r_{i,j} \) be the order of colors in the minimal TSP for graph \( G_{i,j} \). In each regular phase (e.g. regular phase \( i \) in block \( j \)) OPT' transmits the type B packets according to that order - first all packet with color \( c_{i,j,1} \), then all the packet with color \( c_{i,j,2} \) and so on. It is clear that OPT' needs at most \( H + TSP(G) \) time-slots to transmits...
the packets ($H$ for packet transmission and $TSP(G)$ for color transition). OPT’ transmits the packets ordered by arrival time. Each regular phase consist of $H$ time-slots, but OPT’ need up to $H + TSP(G)$ time-slots to transmit the packets. Therefore at the end of the last regular phase in each block there are at most $TSP(G) \log C$ untransmitted packets. These packets are transmitted in the end phase. There are enough time-slots since $H \log C > TSP(G) \log C$ (recall that $L = \Omega(TSP(G) \log C)$ and $H = \sqrt{\frac{TSP(G)\log C}{C}}$).

In the last block OPT’ missed up to $TSP(G) \log C$ packets (since there is no end phase).

- The $L$ packets $(t_{i,j+1}, L + t_{i,j+1}, 1)$ arrived during the final release time are transmitted by OPT’ consequently from time $t_{i,j+1}$. OPT’ can transmits $L$ packets except for one transition period, and hence may lose at most $TSP(G)$ packets.

- According to the observations the final release time $t_{i,j+1}$ is at most $L + 1$. We conclude that OPT’ transmits type B packets until time unit $2L$, and type A packets between $2L + 1$ and $3L$.

We conclude that:

$$\text{OPT}(\sigma) \geq \text{OPT'}(\sigma) \geq Y - (\log C + 1)TSP(G)$$

and the competitive ratio is:

$$\frac{\text{ALG}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{Y - H/2 + 1}{Y - (\log C + 1)TSP(G)} \leq \frac{3L - \frac{1}{2} \left(\frac{TSP(G)\log C}{C}\right) + 1}{3L - (\log C + 1)TSP(G)} = 1 - \Omega \left(\sqrt{\frac{TSP(G)}{L \log C}}\right).$$

2. Case 2.A: The sequence consists of more than $3L$ type A packets, all deadlines are at most $3L$. ALG did not transmit type A packets during the start phase, or during the end phase. Condition ($I_1$) guaranteed that at the end of each regular phase there are up to $H/2$ type B packets. Recall that no type B packets arrive during the start phase nor in the end phase of a block. If ALG did not transmit type A packet during the $2H \log C$ time-slots of the start phase, there are at least $H \log C$ idle time-slots during that period (he had only $H/2$ untransmitted type B packets). From a symmetric argument If ALG did not transmit type A packet during the $2H \log C$ time-slots of the end phase, there are at least $H \log C$ idle
time-slots during that period. We conclude that the number of transmitted packets is:

\[ \text{ALG}(\sigma) \leq 3L - H \log C. \]

At each time unit until the final release time, OPT’ transmits the type A packet that arrived at the same time unit. From the final release time and until time unit \(3L\), OPT’ transmits the type A packets that arrived at the final release time. Therefore OPT’ transmits \(3L\) type A packets. Hence the competitive ratio is:

\[ \frac{\text{ALG}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{3L - H \log C}{3L} = \frac{3L - \sqrt{\log C}}{3L} = 1 - \Omega \left( \sqrt{\frac{\text{TSP}(G) \log C}{L}} \right). \]

3. Case 2.B: The sequence consists of \(4L\) type A packets, all deadlines are at most \(3L\). Every block satisfy condition \((I_1)\) and \((I_2)\). Therefore type A packet is transmitted during the start phase of each block. Then packets from all the type B colors are transmitted during the regular phases and then type A packet is transmitted during the end phase. It is clear that during each block ALG spent at least \(\text{TSP}(G)\) time-slots for color transitions. Since there are \(\frac{1}{5} \sqrt{\frac{L}{\text{TSP}(G) \log C}}\) blocks, at least \(\frac{1}{5} \sqrt{\frac{\text{TSP}(G) L}{\log C}}\) time-slots used for color transition. Hence:

\[ \text{ALG}(\sigma) \leq 3L - \frac{1}{5} \sqrt{\frac{\text{TSP}(G) L}{\log C}}. \]

At each time unit until the final release time, OPT’ transmits the type A packet that arrived at the same time unit. From the final release time and until time unit \(3L\), OPT’ transmits the type A packets that arrived at the final release time. Therefore OPT’ transmits \(3L\) type A packets. Hence the competitive ratio is:

\[ \frac{\text{ALG}(\sigma)}{\text{OPT}(\sigma)} \leq \frac{3L - \frac{1}{5} \sqrt{\frac{\text{TSP}(G) L}{\log C}}}{3L} = 1 - \Omega \left( \sqrt{\frac{\text{TSP}(G) L}{L \log C}} \right). \]

This complete the proof.
4 Online Scheduling Algorithm for General Metric Space and Directed Weighted Graph

In this section we design a deterministic online algorithm, for a weighted directed or undirected graph. Without loss of generality we may assume the triangular inequality. Hence we can view the graph as a complete graph. In particular, general metric space corresponds to an undirected graphs. The algorithm achieves a competitive ratio of $1 - o(1)$, when the minimum weight of the TSP is asymptotically small with respect to the minimum laxity of the packets. As shown in the previous sections, this requirement is essential in designing $1 - o(1)$ competitive algorithm.

4.1 The algorithm

The algorithm is a natural extension of the BG algorithm from [6]. Algorithm BG works in phases of $\sqrt{CL}$ time-slots. At each phases it collects the packets for the next phase. It transmits them according to the colors, from color 0 to color $C - 1$. Our algorithm that we call TSP-EDF formally described in figure 5 works in phase of $K = \sqrt{TSP(G)L}$ time-slots. In each phase the algorithm transmits packets by colors. The order of the colors is determined by the minimum TSP. The algorithm achieves a competitive ratio of $1 - 3\sqrt{\frac{TSP(G)}{L}}$. Clearly finding the $TSP(G)$ is not known to be on polynomial time (NP hard problem). To make our algorithm polynomial we use an approximation of the TSP (e.g. 2-approximation for undirected and $\log C$-approximation for directed). This would replace $TSP(G)$ with $MST(G)$ for undirected graph and with $TSP(G) \log C$ for directed graph.

Analysis. The analysis is similar to the analysis in [6]. First we need to demonstrate that the output schedule $\rho$ is feasible. Specifically, one needs to prove that every scheduled packet $i$ is transmitted during the time frame $[r_i, d_i]$, and that there is a color transition of length $w(i, j)$ between the transmission of any two successive packets with different colors $i$ and $j$.

Lemma 4.1. Algorithm TSP-EDF generates a valid schedule.

Proof. It is implicit by its description. The exact proof is similar to the proof of lemma 3.1 in [6].

$\blacksquare$
In each phase $\ell = 1, 2, \ldots$ do

- Reduce the deadline of each untransmitted packet $(r, d, c)$ from $d$ to $K[d/K]$.
- Let $S^\ell$ be the collection of untransmitted packets such that their reduce deadline was not exceed. Let $S^\ell,K$ be the K-length prefix of EDF schedule (according to the modified deadline) of $S^\ell$. Let $S^\ell,K_j \subseteq S^\ell,K$ denote the subset of packets having color $c_j$. Let $i_1, i_2, \ldots, i_C$ denote the order of the colors in the minimal TSP (or approximation).
- $\rho_\ell$ initially consists of all packets of $S^\ell,K_{i_1}$ scheduled consecutively, then all packets of $S^\ell,K_{i_2}$ scheduled consecutively, and so on.
- $\rho_\ell$ is modified such that a $w(v_i, v_j)$ color transition time-slots are added between each two successive color groups $S^\ell,K_i, S^\ell,K_j$.
- $\rho_\ell$ is modified such that its length will be exactly $K$. If the length of $\rho_\ell$ is more than $K$ then the last packets are dropped. If its length is less than $K$ then it is affixed with idle time-slots.
- The packets are transmitted according to $\rho_\ell$.

Figure 5: Algorithm TSP-EDF.

Now we analyze the performance guarantee of the algorithm. We first define two input sequence $\sigma'$ and $\tilde{\sigma}$, which are modifications of $\sigma$. The input sequence $\sigma'$ consists of all packets in $\sigma$, but modifies the color of packets to a fix color $c'$. Specifically, each packet $(r, d, c) \in \sigma$ defines a packet $(r, d, c') \in \sigma'$. The input sequence $\tilde{\sigma}$ consists of all packets in $\sigma$ such that a packet $(r, d, c) \in \sigma$ gives rise to a packet $\left(K[r/K], K[d/K], c'\right) \in \tilde{\sigma}$, where $c'$ is a fix color. Hence, all packets in $\tilde{\sigma}$ have the same color, and the release and deadline times of each packet in $\tilde{\sigma}$ are aligned with start/end time of the corresponding phase such that the span of each packet is fully contained in the span of that packet according to $\sigma$. Note that the span of a packet $(r, d, c)$ is the time frame $[r, d]$.

**Lemma 4.2.** $\text{OPT}(\tilde{\sigma}) = \text{ALG}(\tilde{\sigma})$.

**Proof.** Note that algorithm TSP-EDF has three modification from regular EDF:

- Packet’s deadline times are modified to $K[d/K]$.
- Packet’s release times are modified to $K[r/K]$ (because in each phase only packets released during previous phases are transmitted).
- Color transition time-slots are added between the transmission of packets from different colors.
The release and deadline times of the packets in $\tilde{\sigma}$ are aligned and all the packets has the same color. Hence ALG’s schedule is identical to EDF’s schedule. Since EDF is optimal for sequences that consist of packets with one color, $OPT(\tilde{\sigma}) = ALG(\tilde{\sigma})$.

**Lemma 4.3.** $OPT(\tilde{\sigma}) \geq \left(1 - 2\sqrt{\frac{TSP(G)}{L}}\right) OPT(\sigma')$.

**Proof.** $\lambda$-perturbation defined in [6] as follows: An input sequence $\hat{\delta}$ is a $\lambda$-perturbation of $\delta$ if $\hat{\delta}$ consists of all packets of $\delta$, and each packet $(\hat{r}, \hat{d}) \in \hat{\delta}$ corresponding to packet $(r, d) \in \delta$ satisfies $\hat{r} - r \leq \lambda$ and $d - \hat{d} \leq \lambda$.

By definition $\tilde{\sigma}$ is $K$-perturbation of $\sigma'$, and the colors of all packets are identical. Hence by using Theorem 2.2 from [6] we obtain the following inequality.

**Lemma 4.4.** $ALG(\sigma) \geq \left(1 - \sqrt{\frac{TSP(G)}{L}}\right) ALG(\sigma')$.

**Proof.** The difference between the schedule TSP-EDF generates for $\sigma$ and the schedule it generates for $\sigma'$ is that packets might be dropped at the end of each phase in $\sigma'$ due to color transition. The worse case for $\sigma$ is when there are no idle time-slots in any of the phases of $\sigma'$. Otherwise the idle time-slots might use for color transition and avoid dropping packets. Therefore there are at most $\left\lceil ALG(\sigma')/K \right\rceil - 1$ phases in which algorithm TSP-EDF transmits packets (the $-1$ term is by the fact that the algorithm does not transmit any packet during the first phase). Since there are no more than $TSP(G)$ color transitions in each phase, we obtain the following inequality:

\[
ALG(\sigma) \geq ALG(\sigma') - \left(\left\lceil ALG(\sigma')/K \right\rceil - 1\right) TSP(G) \\
= \quad ALG(\sigma') - \left(\frac{ALG(\sigma')}{\sqrt{TSP(G)L}} - 1\right) TSP(G) \\
\geq\left(1 - \sqrt{\frac{TSP(G)}{L}}\right) ALG(\sigma') .
\]

We are now ready to prove the main theorem of this section.

**Theorem 4.5.** Algorithm TSP-EDF attains a competitive ratio of $1 - 3\sqrt{\frac{TSP(G)}{L}}$. 

**Proof.** Using the previously stated results, we obtain that

\[
\text{ALG}(\sigma) \geq \left(1 - \sqrt{\frac{TSP(G)}{L}}\right) \text{ALG}(\sigma')
= \left(1 - \sqrt{\frac{TSP(G)}{L}}\right) \text{ALG}(\tilde{\sigma})
= \left(1 - \sqrt{\frac{TSP(G)}{L}}\right) \text{OPT}(\tilde{\sigma})
\geq \left(1 - \sqrt{\frac{TSP(G)}{L}}\right) \left(1 - 2\sqrt{\frac{TSP(G)}{L}}\right) \text{OPT}(\sigma')
\geq \left(1 - 3\sqrt{\frac{TSP(G)}{L}}\right) \text{OPT}(\sigma).
\]

The first inequality results from Lemma 4.4. The first equality follows by the definition of the algorithm. The second equality holds by lemma 4.2. The second inequality result from Lemma 4.3. Finally, the last inequality follows as \(\sigma'\) is alike \(\sigma\), but all packets have the same color. This implies that any schedule feasible for \(\sigma\) is also feasible for \(\sigma'\), and thus, \(\text{OPT}(\sigma') \geq \text{OPT}(\sigma)\).
References


תקציר

 אנו מציגים את בעיית החבילות עם צבעים (buffer) וזמן סיום ותשלום על מעבר צבעים בבעיה זו לנתב יש m פורטים נכנסים ופורט יציאה בודדים. עבור כל וקטור נוכחות וירטואלי זריזה-hook, כאשר לכל הזריזה יש חבטה. הנתב משדר בכל וקטור נוכחות וירטואלי זריזה-hook מחרשה, כאשר לכל חבטה חבטה תלויה בזירה. המטרדה היאแกיון התור המבוקש והבוכים מהריווח המשדרות. לכל וקטור הזריזשה.

рабатываנו של זירית פניז תלויה בזירה. המטרדה היא לה максם את מספר החבילות המשדרות. עבור מטריקה אחידה אנו שיפורים את החסם התחתון שהוצג בAzar at el. 2009 לחסם עבורי המטריקה אוטו משולש (או גרף מקורי) וננוון החbatis במטריקה התור בזירה (או גרף מקורי).

בע馗ודו אנו חוקרים את ההפסדים של הזריזות ומרגון והזירות במטריקה (או גרף מקורי) המרגרן את המטריקה עם זירית הזריזות. בודק לכל ממשיכת הזריזות ובחרת זירית המ☁ 그렇게 הזריזות עבורי מטריקת (או גרף מקורי).

 עבור מצרי הזריזות המטריקה את המטריקות עם זירית הזריזות ומרגון והזירות במטריקה (או גרף מקורי) וננוון החbatis במטריקה התור בזירה (או גרף מקורי).

 עבור מצרי הזריזות המטריקה את המטריקות עם זירית הזריזות ומרגון והזירות במטריקה (או גרף מקורי) וננוון החbatis במטריקה התור בזירה (או גרף מקורי).
חברת אל-אביב
הפקולטה למדעי המדויקים
ע"ש רימונד וברלי סאקלר
בית הספר למדעי המחשב ע"ש בלנסטיק

היבלוות ע"ש גבי, ז"ה סהיא
וחדלו לmpzבר צבצ

הרבר ג. ח. מהנדס מתכוננים לפקולת החכאי
"מוסמך אוניברסטי" (M.Sc.) באוניברסיטת תל-אביב

בית הספר למדעי המחשב

על יידי עידי רודה

העבורה והכנה בהדרכה של פרופ' יוסי עזר

אריר חצ"ש