Online Request Server Matching

Marco Riedel [1]

Heinz Nixdorf Institut, Universität Paderborn, D-33095 Paderborn, Germany

Abstract

In the following paper an alternative online variant of the matching problem in bipartite graphs is presented. It is triggered by a scheduling problem. There, a task is unknown up to its disclosure. However, when a task is revealed it is not necessary to take a decision on the service of that particular task. On the contrary, an online scheduler has to decide on how to use the current resources. Therefore, the problem is called *online request server matching* (ORSM). It differs substantially from the *online bipartite matching* problem of Karp et al. [7]. Hence, the analysis of an optimal, deterministic online algorithm for the ORSM problem results in a smaller competitive ratio of 1.5. An extension to a weighted bipartite matching problem is also introduced, and results of its investigation are presented.

Additional concepts for the ORSM model (e.g. lookahead, parallel resources, deadlines) are studied. All of these modifications are realized by restrictions on the input structure. Decreased competitive ratios are presented for some of these modified models.

Key words: competitive analysis, online bipartite matching, online scheduling

1 Introduction

Motivation. The problem which is investigated here is triggered by an online scheduling problem with deadlines. This problem can be found in server systems for continuous data streams (e.g. video transmissions). Hence, we have to deal with a real-time system. It works in an online fashion (future demands are unknown) and includes hard, individual deadlines for parts of the data. Fortunately, the situation is relaxed a little bit because it seems to be acceptable when a small amount of the data will not be delivered.

[1] Supported by DFG-Graduiertenkolleg "Parallele Rechnernetzwerke in der Produktionstechnik", GRK 124/2-96. This work was partly supported by the EU ESPRIT Long Term Research Project 20244 (ALCOM-IT).

In a first step of a number of simplifications we divide a continuous data stream into packages of roughly equal size. On this assumption we use a simple, discrete time model. When we abstract from additional restrictions, which would be necessary to model a restricted communication network between storage units and customers, the scheduling problem becomes a matching problem in a bipartite graph in a special online variant. The scheduler has to match requests for data packages with deadlines to appropriate server resources and time slots. Thereto, in each time step the scheduling algorithm decides which requests are served by the current resources. These decisions have some impact on the future because there are dependencies between the known, previous requests and the unknown, future ones. Thus, we deal with an online problem and the scheduler has the objective to serve as many requests as it can under the uncertainty of future, unknown demands. In this work we begin such studies with a very basic model for this problem. It is called *online request server matching* (ORSM). An extension to a weighted matching problem is motivated by additional priorities. It is also mentioned in the following.

The ORSM model is modified by the adding of concepts such as parallel resources, short deadlines, and lookahead. These variants represent steps to models for more realistic scheduling problems. The real world applications which triggered our studies, operate with such principles. A few more models of that kind, which also implement first network restrictions can be found in [8]. The study of our modified models has one major advantage. We can analyse the characteristics of each concept independently. Later, these concepts are combined step by step in order to investigate them.

All of these problems are *online*, i.e., the requests appear through time while irreversible decisions about the usage of resources have to be taken without knowledge of future requests. This kind of problems is studied by applying *competitive analysis*. Nowadays, this is a well established method. A comprehensive introduction to this topic is given in [3]. Roughly speaking, for the ORSM and similar benefit problems one determines the worst case ratio of the benefit achieved by an optimal solution to that of a solution calculated by an online algorithm. This ratio is called *competitive ratio*, and an online algorithm which guarantees a competitive ratio of c is called c-competitive. We apply competitive analysis in order to investigate all online problems here.

Previous and Related Work. In the last couple of years a vast amount of results on online scheduling have been published. [11] provides us with an encyclopedic survey. However, studies of scheduling models which allow deadlines are rare and cover very restricted cases only. None of them seems to be closely related to the ORSM model. The opposite is true for bipartite matching problems in online settings. The study of such problems has been initiated by Karp, Vazirani and Vazirani in [7]. Their model consists of a

'known' and an 'unknown' partition. Vertices of the unknown partition are revealed over time and an online algorithm can add at most one edge of a just revealed vertex to the online matching. The simple GREEDY algorithm is an optimal, 2-competitive, deterministic algorithm for this problem. The key contribution in [7] is the analysis of the optimal, randomized algorithm RANKING which is $e/(e-1) \approx 1.582$-competitive.

Various extensions and variants of this problem are studied later. A comprehensive survey and further reference can be found in [6].

An online matching problem in general graphs is the *roommates problem* [2]: By and by guests arrive at a hotel for a conference. The hotel consists of double rooms only. Every guest has a list of persons he would agree to share a room with (it is assumed that these lists are symmetric; they represent the adjacency lists of vertices in an undirected graph). The manager must immediately assign a room to every guest. At the same time he wants to minimize the number of occupied rooms. In [2] an optimal, deterministic, 1.5-competitive algorithm is shown. A lower bound of 3, and a 4-competitive algorithm are presented for a weighted extension of the problem.

Results and Organization of the Material. In the next section the ORSM model is formally described. It includes all the modifications, which are studied later. In Sect. 3 an analysis of the 'standard' ORSM problem is presented. A lower bound and a matching upper bound are shown. It includes a 1.5-competitive algorithm. At the end of that section a few results of a weighted model variant are given. The competitive ratio of that model is in between the golden ratio ($(\sqrt{5}+1)/2 \approx 1.618$) and 2. In Sect. 4 a set of modifications of the ORSM model is examined. Table 1 on page 11 illustrates the results. A further model variant, which has caught our particular attention, is investigated in Sect. 5. It contains lower and upper bounds and a discussion of a specialized algorithm. All models in Sect. 4 and 5 are special cases of the ORSM problem. Thus, the 1.5 upper bound holds for all of these models. For some of them decreased competitive ratios are presented. The paper ends on some concluding notes in Sect. 6.

2 The Model

A formal definition of the *online request server matching* problem (ORSM) and modifications of it are now presented.

The underlying structure of the ORSM problem is a bipartite graph $G := (R \cup S, E)$. Both partitions R and S are totally ordered. We denote the vertices

by $r_1, r_2, r_3, \ldots$ and by $s_1, s_2, s_3, \ldots$ with $r_i \in R$, $s_j \in S$, and the indices indicate the position in the order. This order is interpreted as a discrete time model. Vertices of partition S represent a single resource called *server*. It is available for one unit per time step. Partition R is interpreted as a set of tasks. Such a task has a demand of one server unit to be completed. They are called *requests* and in each time step one of them may occur. An edge $\{r_i, s_j\}$ between a request vertex r_i and a server vertex s_j means that the request r_i can be served in time step j. The set of edges $E \subset R \times S$ is constructed according to the following restriction:

$$\{r_i, s_j\} \in E \;\Rightarrow\; i \leq j \;\; . \tag{1}$$

Consequently, a request occurring at time step i must not specify a possible service time in the past. Without this restriction the modelled scheduling problem does not make sense and no competitive online algorithm would be possible.

Remark 1 *The above definition of the set of edges implies that a request does not need to specify an interval of consecutive time steps for the service. Establishing this definition we had in mind that our very much simplified model is a preliminary stage to more complex models. Such models include several server resources and the requests can be handled by a subset of resources only. See also [1,8] for descriptions of such models. On the other hand, the scheduling problem becomes trivial when requests specify a time interval for the service only. The simple 'earliest deadline first' heuristics is 1-competitive for such a special single server problem.*

Now we have to specify how this model works online: When the system starts, partition R is completely unknown.[2] At the beginning of time step i request r_i is revealed as input, i.e., vertex r_i and all of its incident edges are inserted into the previously known part of G. If no request appears, vertex r_i will be isolated. After this input an algorithm has to decide on how to use the server in the current time step i. Indeed, it can add an edge incident to s_i to the online matching M. It is worth noting that due to restriction (1) *all* edges incident to s_i are known when this decision has to be taken. The online algorithm has the objective to maximize the cardinality of the matching M, i.e. to serve as many requests as it can.

We want to emphasize the difference between our model and the one in [7]. Karp et al. have studied an online version of the bipartite matching problem,

[2] A close examination reveals that this is not the whole truth: It is known that in each time step i a new vertex r_i is inserted but its set of incident edges is unknown before i. For reasons of convenience we interpret the input process in the above introduced way.

with the input at a time being a vertex out of the 'unknown' partition and all of its incident edges. Thereafter, an online algorithm has to select at most one of these *just revealed* edges for the online matching. On the contrary, the decision on a matching edge in the ORSM problem is taken by selecting one of the edges which are incident to a vertex of the *opposite* partition. Nevertheless, the ORSM problem can be interpreted as a special case of the online bipartite matching. Thereto, the server partition is the 'unknown' partition and whenever a decision has to be taken, all edges incident to that vertex are known. Furthermore, all additional information given by the ORSM model is ignored. Shifting focus, we can also recognize that the ORSM problem is a special case of the *roommates problem*. The precise link is shown in [9].

The adding of a weight function $w : E \to \mathbb{R}_+$ to graph G of the ORSM model results in an extension. In this case the objective is the optimization of the total weight of a weighted matching. This version is called *online request server weighted matching* problem, or in short wORSM problem. A detailed discussion of this variant of the problem can also be found in [9].

Other variants of the standard model can be constructed by restricting the input structure. Indeed, it is possible to express additional concepts such as:

- *Lookahead* ℓ $(\ell \in \mathbb{N})$ – when taking the decision on the usage of server vertex s_i, the next ℓ inputs are revealed. This model is equivalent to the ORSM model with the restriction $\{r_i, s_j\} \in E \Rightarrow i + l \leq j$.
- *Block input* b $(b \in \mathbb{N},\ b \geq 2)$ – in every time step i the next b request vertices $(r_{ib+1}, \ldots, r_{ib+b})$ are revealed and the usage of the next b server vertices $(s_{ib+1}, \ldots, s_{ib+b})$ has to be determined. Indeed, the total order on the vertices can be relaxed and b parallel resources can be modelled. When we divide the total order of the ORSM model into blocks of size b and when the edges only connect request vertices of odd block numbers to server vertices of even blocks, we get an equivalent description for 'block input'.
- *Request degree* g $(g \in \mathbb{N}, g \geq 2)$ – request vertices have a constant number g of edges, or they are isolated (no request appears in that time step). It is not fruitful to only bound that number g because adversaries always take advantage of the smallest degree.
- *Limited edge length* d $(d \in \mathbb{N})$ – an edge can span at most d time steps, i.e., for every edge $\{r_i, s_j\} \in E$ it must hold $j \leq i + d$. This restriction models short, individual deadlines.

These concepts can be combined with each other. Then, lookahead ℓ and limited edge length d are defined in terms of full blocks when block input is an implemented concept. I.e., an edge can reach the next d blocks, and the lookahead is ℓ blocks of size b. Model variants with such concepts are worth studying because real world scheduling problems work with such restrictions.

In the discussions and proof of this work we use M_{ALG} to denote an online matching which is constructed by algorithm ALG. OPT denotes an optimal (offline) algorithm. The size of a matching M is denoted by $|M|$, i.e. the number of edges in M.

3 Analysis of the ORSM Problem

This section starts with a general lower bound for the competitive ratio of the ORSM problem for deterministic online algorithms. Hereafter, the optimal 1.5-competitive algorithm LMM is presented and analysed. Finally, results of the weighted variant of the ORSM model are presented.

The Lower Bound

By applying the standard argument of an adversary strategy, we show the following general lower bound:

Theorem 2 *Every deterministic online algorithm ALG for the ORSM problem has a competitive ratio of at least* 1.5.

PROOF. The adversary strategy starts with the following input structure: $\{\{r_1, s_2\}, \{r_1, s_3\}, \{r_2, s_2\}, \{r_2, s_4\}\}$.

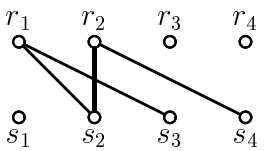

Fig. 1. Situation at time $i = 2$.

ALG can react to this input at time $i = 2$ in three different ways:

Case 1 (Fig. 2): ALG adds edge $\{r_1, s_2\}$ to the online matching M_{ALG}. In the next step the adversary presents edge $\{r_3, s_4\}$. ALG cannot use the server vertex s_3. Therefore, $|M_{\mathsf{ALG}}| \leq 2$. However, the optimal solution results in $|M_{\mathsf{OPT}}| = 3$.

Case 2 (Fig. 3): ALG adds edge $\{r_2, s_2\}$ to the online matching M_{ALG}. In the next step the adversary presents edge $\{r_3, s_3\}$. ALG cannot use s_4. Again $|M_{\mathsf{ALG}}| \leq 2$ although the optimal matching results in $|M_{\mathsf{OPT}}| = 3$.

Case 3 (Fig. 4): ALG decides not to match s_2. The adversary presents the input of Case 1 and it holds $|M_{\mathsf{ALG}}| \leq 2$, $|M_{\mathsf{OPT}}| = 3$. Alternatively, it is fairly

obvious that ALG cannot take advantage of such a decision.

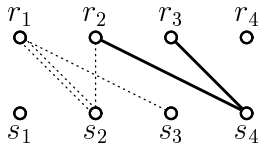

Fig. 2. $\{r_1, s_2\} \in M_{\mathsf{ALG}}$

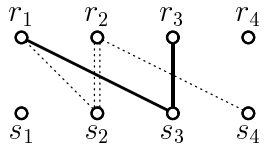

Fig. 3. $\{r_2, s_2\} \in M_{\mathsf{ALG}}$

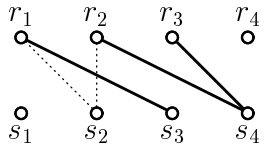

Fig. 4. $M_{\mathsf{ALG}} = \varnothing$

The figures show the states at time $i = 3$, after the different possible decisions of ALG and the reactions of the adversary; a double line is a matching edge; dotted edges have been removed before $i = 3$.

The described strategy can be repeated infinitely every four time steps and thus the following ratio is implied:

$$\frac{|M_{\mathsf{OPT}}|}{|M_{\mathsf{ALG}}|} \geq \frac{3}{2} .$$

□

The Algorithm LMM

At time step i graph G representing the input of an online algorithm is known up to request vertex r_i. To be more precise, we know the subgraph of G induced by set $\{r_k |\ r_k \in R, 1 \leq k \leq i\} \cup S$. Due to the irreversible decisions of the former time steps all previous server vertices $s_1, \ldots, s_{i-1}$ and all hitherto matched request vertices cannot be rearranged anymore. Thus, we have a 'local' bipartite subgraph B_i of G which is induced by vertex set $V_i = \{r_k |\ r_k \in R, 1 \leq k \leq i, r_k \text{ not yet matched}\} \cup \{s_k |\ s_k \in S, i \leq k\}$.

Our online algorithm is called 'Local Maximum Matching' (LMM) because it constructs a maximum matching on every local subgraph B_i (denoted by $\mathcal{M}(B_i)$ / see e.g. [4] for some further explanations of how to do so). The exact function of LMM follows:

1: **loop** {for all time steps i}
2: read input of time i and build up B_i
3: construct a maximum matching $\mathcal{M}(B_i)$ on B_i :
start with all matching edges of $\mathcal{M}(B_{i-1})$ which are edges in B_i;
4: look for an augmenting path which starts at vertex r_i and do the augmentation when found
{Due to the maximum cardinality of $\mathcal{M}(B_{i-1})$,
every augmenting path must have r_i on one end.}
5: **if** s_i is matched in $\mathcal{M}(B_i)$ **then**
6: add the matching edge of s_i to the online matching M_{LMM}
7: **else if** s_i is not isolated in B_i **then**

{all neighbours of s_i are matched in $\mathcal{M}(B_i)$}
8: add an arbitrary edge $\{s_i, r\}$ of B_i to the matching M_{LMM} and delete the matching edge of r in $\mathcal{M}(B_i)$
9: **end if**
10: **end loop**

Line 8 of this algorithm is essential and prefers the current vertex s_i.

The Upper Bound

In order to analyse the performance of LMM we need two observations:

(1) After request vertex r_i was matched in B_i (in line 4 of LMM), r_i is in all local maximum matchings of the following time steps[3] up to that time step where the current matching edge of r_i is added to M_{LMM} (in line 6 or 8 of LMM).
(2) If s_i is not isolated in B_i, then s_i is in the final matching M_{LMM}. (See lines 5 to 8 of LMM.)

Theorem 3 *The deterministic online algorithm LMM for the ORSM problem is 1.5-competitive.*

PROOF. We show that an online matching M_{LMM} cannot be increased by augmenting paths of length one or three. (A set of augmenting paths is described by the exor-operation between M_{LMM} and a fixed M_{OPT}. Consult [4] for details.) Therefore, the shortest augmenting paths with respect to M_{LMM} must have a length of five. This fact immediately implies the theorem because by performing the augmentation, two matching edges out of M_{LMM} become three edges in M_{OPT}. Longer augmenting paths have a lower ratio and matching edges outside such augmenting paths decrease the overall ratio too.

By applying a complete distinction of cases, we prove the non-existence of augmenting paths of length one and three in the online matching M_{LMM} by contradiction.

Fig. 5. Structure of augmenting paths of length one and three.

Case 1 Augmenting path of length one $\{s_i, r_a\} \notin M_{\mathsf{LMM}}$: Vertex r_a has never been matched because r_a is not in M_{LMM} (reverse application of observation 1). So $\{s_i, r_a\}$ is in B_i and this fact contradicts observation 2.

[3] See the copy process of line 3 and remember that augmentations change matching edges but they do not remove vertices out of the matching.

Case 2 Augmenting path of length three and $i < j$: r_a was matched at time j only, which implies that edge $\{s_i, r_a\}$ was in B_i. This is a contradiction to observation 2.

Case 3 Augmenting path of length three and $i > j$: At time j, request vertices r_a and r_b are not in M_{LMM} because r_b has never been matched (again reverse application of observation 1), and $\{r_a, s_j\}$ is added to M_{LMM} just at time j. Thus, the whole path $P := \{\{s_i, r_a\}, \{r_a, s_j\}, \{s_j, r_b\}\}$ is in B_j. The decision $\{s_j, r_a\} \in M_{\mathsf{LMM}}$ could not have been made by line 8 of LMM. This special treatment requires that s_j is not matched in B_j. However, in such a case $\{s_j, r_b\}$ must be inserted into $\mathcal{M}(B_j)$ to ensure its optimality. Indeed, path P is an augmenting one (and therefore a contradiction to the optimality of $\mathcal{M}(B_j)$) unless s_i is matched in B_j, i.e., there exists a request vertex r_c with $\{r_c, s_i\} \in \mathcal{M}(B_j)$. Only the special treatment of line 8 of LMM can remove a matching edge completely from a previously matched server vertex. The fact that s_i is not matched in the final solution M_{LMM} implies that at a time k ($j < k < i$) the matching edge $\{r_c, s_i\}$ is removed by line 8 of LMM, and edge $\{s_k, r_c\}$ is inserted into M_{LMM}. This is only possible when the server vertex s_k is not matched in B_k.

However, the definition of the ORSM problem implies, that both edges $\{r_c, s_i\}$ and $\{r_c, s_k\}$ are known at time j. Therefore, the depicted path from s_k to r_b is an augmenting path in $\mathcal{M}(B_j)$. This is again a contradiction to the optimality of $\mathcal{M}(B_j)$, or s_k is matched in $\mathcal{M}(B_j)$. The above line of argument on s_i is now applied to s_k. Indeed, B_j is a finite graph, and therefore, the repeated application of the given arguments must end in a contradiction to the existence of an augmenting path of length three in M_{LMM}.

The case that an augmentation exchanges the matching edge of s_i within time interval (i, k) has not been considered as yet. However, the matching edge of s_i is removed at time k, and therefore s_k, is not matched in B_k. When such a situation occurs after the extension of the matching by an augmenting path and via s_i we can conclude the existence of a similar structure P' before, taking into account that the graph is bipartite and all edges incident to a request vertex are revealed simultaneously. P' is a subpath of that augmenting path and starts with s_i and a matching edge, its edges alternates between non-matching and matching edges, and it ends with a server vertex $s_{k'}$ which is not matched. Therefore, the above arguments can be applied to P' and this leads us to the required contradiction.

Together with Theorem 2 this is a tight analysis of LMM and it is 1.5-competitive. □

The Weighted Model

A study of the wORSM problem can be found in [9]. Here, we only present the general lower bound. Divergent form previous usage, $|M|$ denotes the total weight of a weighted matching M in this section, i.e. $|M| = \sum_{e \in M} w(e)$.

Theorem 4 *Every deterministic online algorithm ALG for the wORSM problem has a competitive ratio of at least* $\phi = (\sqrt{5}+1)/2 \approx 1.618$.

PROOF. The adversary strategy starts with input edges $\{r_1, s_1\}$ and $\{r_1, s_2\}$ and their weights are $w(\{r_1, s_1\}) = 1$ and $w(\{r_1, s_2\}) = \phi$ as you can see in Fig. 6. ALG can react to this input at time $i = 1$ in two different ways:

Case 1 (Fig. 7): ALG adds edge $\{r_1, s_1\}$ to the weighted online matching M_{ALG}. Thereafter, the adversary does not present any new edge incident to s_2. So $|M_{\mathsf{ALG}}| = 1$ and $|M_{\mathsf{OPT}}| = \phi$ holds.

Case 2 (Fig. 8): ALG does not change the online matching M_{ALG}. The adversary presents edge $\{r_2, s_2\}$ with weight $w(\{r_2, s_2\}) = \phi$. Now ALG can construct a matching with weight $|M_{\mathsf{ALG}}| \leq \phi$ only, whereas it holds $|M_{\mathsf{OPT}}| = 1 + \phi$. The ratio of these two values is also ϕ.

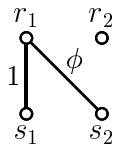

Fig. 6. Situation at time $i = 1$.

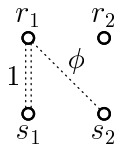

Fig. 7. $\{r_1, s_2\} \in M_{\mathsf{ALG}}$

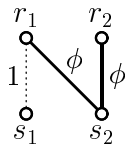

Fig. 8. $M_{\mathsf{ALG}} = \varnothing$

Fig. 7 and 8 show the situations at time $i = 2$ after the two possible decisions of ALG and the reactions of the adversary; a double line is a matching edge; dotted lines are edges which have been removed before $i = 2$.

The adversary can repeat this strategy every two time steps up to infinity and this fact shows the lower bound of the competitive ratio to be $\phi = (\sqrt{5}+1)/2$. □

Algorithm LMM can be slightly modified in order to solve the wORSM problem. Therefore, the online algorithm calculates a maximum weighted matching on the local bipartite graph B_i. This algorithm, called wLMM, is also described in [10]. In [9] a tight analysis can be found and it shows a competitive ratio of 2 for wLMM. This fact points out a gap to the general lower bound presented in Theorem 4.

4 The ORSM Problem with Restricted Inputs

Modified models of the ORSM problem have been introduced and motivated at the end of Sect. 2. All of them can be derived from the standard model by adding restrictions to the input structure. Therefore, the 1.5 upper bound of Theorem 3 is valid for all of these models. Table 1 displays the results of this section.

Table 1
The ORSM problem with different combinations of additional concepts and the resulting competitive ratios.

Lookahead ℓ	Block Input b	Request Degree g	Edge Length d	Competitive Ratio	Comment
×	×	×		1.5	
			×	1.5	$d \geq 2$
	×		×	1.5	
×			×	$\begin{cases} \frac{2\lfloor \ell/d \rfloor + 3}{2\lfloor \ell/d \rfloor + 2} & \text{if } (\ell+1) \bmod d \neq 0 \\ \frac{2\lceil \ell/d \rceil + 2}{2\lceil \ell/d \rceil + 1} & \text{if } (\ell+1) \bmod d = 0 \end{cases}$	
×	×		×	$\frac{2\lfloor \ell/d \rfloor + 3}{2\lfloor \ell/d \rfloor + 2}$	

The results in Table 1 are tight and the upper bounds are shown by LMM. The proof of the bounds consists of simple modifications of the proof of Theorem 2 and 3. Next, we formalize these results and give the proof.

The concepts of lookahead, block input, and g-regular request vertices do not change the competitive ratio of the standard ORSM model. The same holds for all combinations of the above mentioned concepts. The reason for this behaviour is the fact that edges in the bipartite graph can be of arbitrary 'length'. I.e., an edge can span a time interval of arbitrary length. Hence, an adversary can construct inputs, which do not reveal more information to the online algorithm then inputs in the standard model do.

Theorem 5 *The ORSM problem with lookahead ℓ, block input b, and g-regular request vertices has a competitive ratio of* 1.5.

PROOF. The upper bound is proven by Theorem 3. The proof of the lower bound uses the adversary strategy of Theorem 2 with a slightly modified input structure:

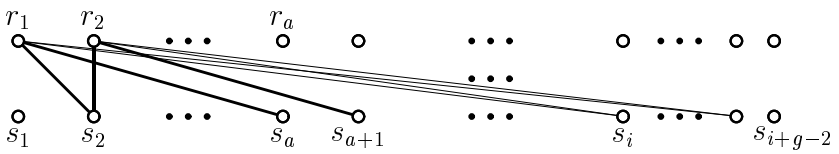

Fig. 9. The input and situation at time $i = 2$.

The adversary sets $a = (\ell + 1) \cdot b + 1$. An online algorithm ALG has to take a decision on s_2 and it is in the same situation as it is in Theorem 2. The reaction of the adversary is also the same: it constructs $\{r_a, s_a\}$ or $\{r_a, s_{a+1}\}$ as input. All of the needless $g-1$ or $g-2$ edges of requests used in the input structure (and in all of its repetitions) are 'collected' at unique $g-1$ server vertices far away ($s_i, \ldots, s_{i+g-2}$ are situated after all used request vertices). Hence, for an arbitrary number n of repetitions of this strategy it holds that $|M_{\mathsf{OPT}}| = 3n$ and $|M_{\mathsf{ALG}}| = 2n + g - 1$, with g being constant and independent of the input length. This leads to a lower bound of the competitive ratio of 1.5. □

The next question that arises is what will happen when the length of the edges is limited by d? Trivial models are implied by $d \in \{0, 1\}$. Therefore, only $d \geq 2$ is of interest. The competitive ratio remains 1.5 because the proof of Theorem 2 works. The adversary of this proof uses only edges with a length of at most 2. We get:

Theorem 6 *The ORSM problem with limited edge length d ($d \geq 2$) has a competitive ratio of* 1.5.

In the combination of limited edge length d and block input, d is interpreted as the number of blocks an edge can span. A similar theorem follows:

Theorem 7 *The ORSM problem with block input b and limited edge length d ($d \geq 1$) has a competitive ratio of* 1.5.

PROOF. The upper bound is proven by Theorem 3 and the matching lower bound works like Theorem 2 with a modified input structure:

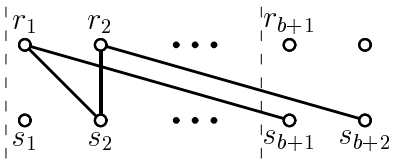

Fig. 10. Situation at time $i = 2$. The dotted lines represent borders between input blocks.

The online algorithm cannot receive more information on the graph structure as long as $d \geq 1$, i.e., edges can reach the next block. Thus, the adversary uses

the input depicted above, and includes edge $\{r_{b+1}, s_{b+1}\}$ or $\{r_{b+1}, s_{b+2}\}$ in time step $b+1$. The further line of argumentation follows the one in Theorem 2. □

When the concepts of lookahead ℓ and limited edge length d are combined, the competitive ratio decreases whenever ℓ is a multiple of d. Thereby, it is guaranteed that an online algorithm has some knowledge of the graph structure in advance whenever a decision has to be taken.

Theorem 8 *The competitive ratio $\mathcal{R}$ of the ORSM problem with lookahead ℓ and limited edge length d ($d \geq 2$) is:*

$$\mathcal{R} = \begin{cases} \dfrac{2\lfloor \ell/d \rfloor + 3}{2\lfloor \ell/d \rfloor + 2} & ,if \quad (\ell+1) \bmod d \neq 0 \\ \dfrac{2\lceil \ell/d \rceil + 2}{2\lceil \ell/d \rceil + 1} & ,if \quad (\ell+1) \bmod d = 0 \end{cases} .$$

PROOF. The adversary uses almost always the maximal edge length d for the proof of the lower bound. Besides, a strategy similar to the one in Theorem 2 is used. It starts with the following input structure for $(\ell+1) \bmod d \neq 0$. This is an 'elongation' of the known one.

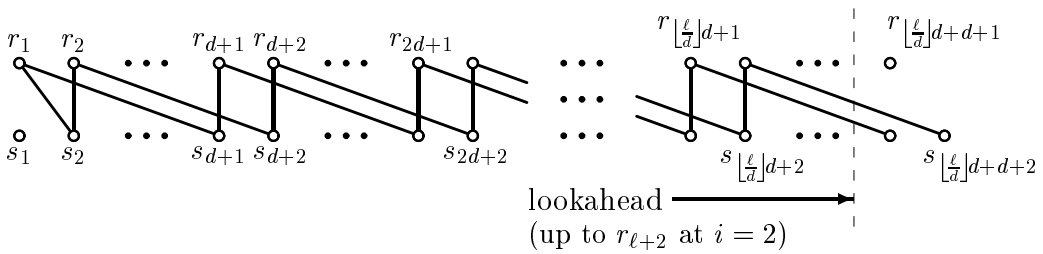

Fig. 11. Situation at time $i = 2$ with a lookahead of ℓ next inputs.

The adversary reacts to the decision on s_2 with edge $\{r_{\lfloor l/d \rfloor d+d+1}, s_{\lfloor l/d \rfloor d+d+1}\}$, or it inserts $\{r_{\lfloor l/d \rfloor d+d+1}, s_{\lfloor l/d \rfloor d+d+2}\}$, respectively. For this input structure it holds $|M_{\mathsf{ALG}}| \leq 2\lfloor \ell/d \rfloor + 2$ and $|M_{\mathsf{OPT}}| = 2\lfloor \ell/d \rfloor + 3$. Again, the argumentation is the same as in Theorem 2.

Whenever $(\ell+1) \bmod d = 0$ holds, the situation becomes asymmetric. Thus, the adversary starts with the input structure depicted in Fig. 12.

The reaction of the adversary to the online decision on s_2 is the insertion of edge $\{r_{\lceil l/d \rceil d+2}, s_{\lceil l/d \rceil d+2}\}$, or it inserts $\{r_{\lceil l/d \rceil d+2}, s_{\lceil l/d \rceil d+3}\}$, respectively. It holds $|M_{\mathsf{ALG}}| \leq 2\lceil l/d \rceil + 1$ and $|M_{\mathsf{OPT}}| = 2\lceil l/d \rceil + 2$, and the infinite repetitions of the described strategies provide the lower bounds.

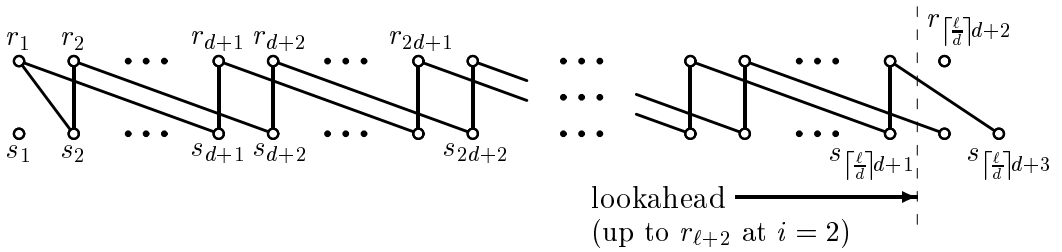

Fig. 12. Situation of the asymmetric case at time $i = 2$ with a lookahead of ℓ next inputs.

The online algorithm LMM is used to show the matching upper bounds. Therefore, the line of arguments of Theorem 3 is applied. When taking a decision on s_i, LMM knows the $(2\lfloor l/d \rfloor+2)$-neighbourhood of s_i in G (if $(\ell+1) \bmod d \neq 0$). This fact follows from the limitation of the edge length and the value of the lookahead. Thus, the non-existence of augmenting paths of length $4\lfloor l/d \rfloor + 3$ in M_{LMM} can be shown, and the upper bound follows. With one more case distinction the non-existence of augmenting paths of length $4\lceil l/d \rceil + 1$ in M_{LMM} can be shown for the asymmetric case when $(\ell + 1) \bmod d = 0$. □

In the model with block input, lookahead, and limited edge length the lookahead and the maximal edge length are defined with respect to full blocks. Therefore, both ends of a path constructed by an adversary like in Fig. 11, are situated in the same block. An exception for the competitive ratio as seen in Fig. 12 when $(\ell + 1) \bmod d = 0$ does not exist and the following theorem holds:

Theorem 9 *The ORSM problem with block input b, lookahead ℓ, and limited edge length d ($b \geq 2$, $d \geq 1$) has a competitive ratio of*

$$\frac{2\lfloor \ell/d \rfloor + 3}{2\lfloor \ell/d \rfloor + 2} \ .$$

Studies of the ORSM model with fixed request degree g and limited edge length d are reported in the following section. This model is a next step towards real world scheduling problems.

5 The ORSM Problem with Fixed Request Degree and Limited Edge Length

The ORSM model with these two restrictions entail a finite number of different inputs for every time step. Hence, the local bipartite graph B_i consists of at most the vertices $\{r_{i-d}, \ldots, r_i\} \cup \{s_i, \ldots, s_{i+d}\}$. When the model parameters g

and d are fixed, the last observation implies the existence of a *finite* number of different local bipartite graphs. These graphs represent the configurations of our online algorithms. Therefore, it is possible to determine the competitive ratio by 'exhaustive search' done by a computer programme. In [5, p. 534], some proof of the existence of potential functions for analysing online algorithms is given. Based on that idea our programme calculates and checks such potential functions. For very small values of g and d the memory consumption and running time did not exceed our computer resources. Consequently, a few competitive ratios could be determined. They are depicted as bold numbers in Table 2. The reason for this investigation is the following question: Do competitive ratios less than 1.5 exists in this restricted ORSM model?

We have also constructed new adversary strategies to show lower bounds for the competitive ratio in this model. These adversaries work for limited ranges of the parameters only and they are based on the same idea as Theorem 2. Their input structures are tedious extensions of the given one. In order to get an impression of the numerical values of these lower bounds, a few of them are illustrated in Table 2. Please note that the values in the columns are arranged by the *difference* between parameter d and g instead of the plain value of d.

Table 2

$d-g=$	0	1	2	3	4	5	6	7	8
$g=2$	**1.250**	**1.333**	**1.429**	**1.500**					
3	**1.167**	**1.286**	1.333	1.375	1.429	1.500			
4	**1.125**	**1.222**	1.300	1.333	1.364	1.400	1.417	1.429	1.500
5	**1.100**	1.182	1.250	1.308	1.333	1.357	1.385	1.400	1.400
6	**1.083**	1.154	1.214	1.267	1.312	1.333	1.353	1.375	1.389

It is obvious that the values in a row are monotonically increasing. Every lower bound for value d holds for all larger values of d. Hence, the redundant numbers of 1.5 in the upper right corner are omitted.

It is an interesting fact that the competitive ratio is bounded below by 1.5 for large values of d ($d \geq 4g-4$, $g \geq 3$). Thus, the matching upper bound is shown by LMM and the additional knowledge of the graph structure does not lead to improved competitive ratios. The opposite is true for small values of d. The output of our computer programme shows decreased values for the competitive ratio. However, the matching upper bounds depicted in bold numbers in Table 2, require a refined online algorithm. Such an algorithm selects the local maximum matching $\mathcal{M}(B_i)$ in respect to additional objectives. Strictly speaking, such an algorithm takes into account the bipartite graph $\tilde{B}_i$ which remains after having removed the matching edge of s_i and its vertices from B_i. Firstly, it only considers solutions which maximize the number of non-isolated server vertices in $\tilde{B}_i$. Secondly, the edges of $\tilde{B}_i$ should be incident to

server vertices of small indices. I.e., the algorithm prefers a solution with early server vertices. Both objectives are leading to a decision, which increases the degree of freedom for future decisions, and the success of an adversary can be derogated.

From the above studies of the ORSM problem with fixed request degree g and limited edge length d we can conclude the following facts:

- For large values of d ($d \geq 4g - 4$, $g \geq 3$) the online algorithm LMM is 1.5-competitive and this is optimal in the sense of competitive analysis.
- Decreased competitive ratios are possible for small values of d.
- Such decreased competitive ratios require more sophisticated online algorithms.

6 Concluding Remarks

The analysis of the ORSM problem with fixed request degree and limited edge length is not complete. It is necessary to develop techniques in order to prove upper bounds of the competitive ratio analytically. The calculation of potential functions by a computer programme may be a step in the right direction.

The ORSM problem should be modified later in order to model scheduling problems with parallel resources and unit tasks with dedicated resources and individual time windows for their service. The aim is the analysis of a complete server and network environment for continuous media streams.

We have the feeling that the upper bound of the wORSM problem can be improved and the existing gap can be closed. Thus, an online algorithm must prefer the current server vertex, e.g. by increasing the weights of its edges.

Fairly often does the following question arise: Can a randomized algorithm perform 'better'? Randomized online algorithms are studied in a slightly different model of competitive analysis. However, that was not the subject of our investigation. Nevertheless, a simple extension of the lower bound construction of Theorem 2 immediately reveals a $6/5 = 1.2$ lower bound for the competitive ratio of randomized online algorithms against the oblivious adversary (the weakest adversary model). Therefore, the adversary chooses randomly, independently, and uniformly one of the input structures which is used in the proof of Theorem 2 in order to build up the input of the next four time steps. It is also not difficult to transform LMM into a randomized online algorithm. Thus, it randomly selects one of the proper maximum cardinality matchings on B_i. However, we do not have an improved analysis yet. Note that a straightforward application of the RANKING algorithm [7] cannot achieve an improved

bound for the ORSM problem.

Acknowledgements

The author would like to thank Yossi Azar for a discussion and the remark that the ORSM problem can be seen as a special case of the online bipartite matching problem. Thanks also to two anonymous reviewers for their helpful comments.

References

[1] Petra Berenbrink, Marco Riedel, and Christian Scheideler. Simple competitive request scheduling strategies. In *Proceedings of the Eleventh ACM Symposium on Parallel Algorithms and Architectures, (Saint-Malo, France, June 27-30, 1999)*, New York, 1999. ACM Press.

[2] Ethan Bernstein and Sridhar Rajagopalan. The roommates problem: Online matching on general graphs. Technical Report CSD-93-757, University of California, Berkeley, CS Department, 1993.

[3] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[4] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal on Mathematics*, 7:449–467, 1965.

[5] Sandy Irani and Anna R. Karlin. Online computation. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 13, pages 521–564. PWS Publishing Company, Boston, Mass., 1996.

[6] Bala Kalyanasundaram and Kirk Pruhs. On-line network optimization problems. In Amos Fiat and Gerhard J. Woeginger, editors, *Online Algorithms: The State of the Art*, LNCS 1442, pages 268–280. Springer-Verlag, Berlin-Heidelberg-New York, 1998.

[7] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing —STOC '90 (Baltimore, Maryland, May 14–16, 1990)*, pages 352–358, New York, 1990. ACM SIGACT, ACM Press.

[8] Burkhard Monien, Petra Berenbrink, Reinhard Lüling, and Marco Riedel. Online scheduling of continuous media streams. In C. Freksa, M. Jantzen, and R. Valk, editors, *Foundations of Computer Science: Potential-Theory-Cognition*, LNCS 1337, pages 313–320, Berlin-Heidelberg-New York, 1997. Springer-Verlag.

[9] Marco Riedel. Online request server matching. Technical Report tr-ri-98-195, University of Paderborn, Germany, April 1998.

[10] Marco Riedel. Online matching for scheduling problems. In Chrisoph Meinel and Sophie Tison, editors, *Proceedings of the 16th Annual Symposium on Theoretical Aspects in Computer Science —STACS '99, (Trier, Germany, March 4–6, 1999)*, LNCS 1563, pages 571–580, Berlin-Heidelberg-New York, 1999. Springer-Verlag.

[11] Jiří Sgall. On-line scheduling. In Amos Fiat and Gerhard J. Woeginger, editors, *Online Algorithms: The State of the Art*, LNCS 1442, pages 196–231. Springer-Verlag, Berlin-Heidelberg-New York, 1998.