# On the Advice Complexity of Online Bipartite Matching and Online Stable Marriage

Shuichi Miyazaki[a]

[a] *Academic Center for Computing and Media Studies, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, JAPAN*

**Abstract**

In this paper, we study the advice complexity of the online bipartite matching problem and the online stable marriage problem. We show that for both problems, $\lceil \log_2(n!) \rceil$ bits of advice are necessary and sufficient for a deterministic online algorithm to be optimal, where $n$ denotes the number of vertices in one bipartition in the former problem, and the number of men in the latter.

*Key words:* online algorithms, advice complexity, bipartite matching, stable marriage

## 1. Introduction

### 1.1. Online Computation and Advice Complexity

Online computation has received considerable attention because of its relevance to real world problems. In an online problem, an online algorithm receives a sequence of requests one at a time, and must decide how to deal with the current request before receiving future ones. This decision cannot be revoked later.

The most popular and successful method of evaluating the efficiency of online algorithms is the *competitive analysis* [27], in which the performance of an online algorithm is compared with that of an optimal offline algorithm. However, it is sometimes pointed out that online and offline algorithms differ by nature and hence it is unfair to compare them directly. Accordingly, alternative ways of analyzing online algorithms have been proposed, such as the relative worst order ratio [8].

Another line of research is to give additional power to online algorithms for measuring the complexity of online problems. A typical example of this is resource augmentation [27, 20], where online algorithms are allowed to use more resources than optimal offline algorithms. Recently, a new framework, called the *advice complexity* [11, 15, 6], was proposed, in which an online algorithm performs its computation with the help of advice bits. There are several different models of online computation with advice. In the first model [11], an online algorithm is allowed to make a query at any time to an oracle that knows the whole input and has unlimited computation power, and use its answer (advice) for computation. In the second model [11, 15], an online algorithm receives a fixed number of advice bits attached with each request. In this paper, we adopt the most

*Email address:* `shuichi@media.kyoto-u.ac.jp` (Shuichi Miyazaki)

general and widely used model [6], which is defined using advice tapes. In this model, an online algorithm is given an advice tape at the beginning, whose contents is a binary sequence $\phi$. During the computation, an online algorithm may read the advice tape and use its contents for the decision. We say that an online algorithm $A$ is $c$-competitive with advice complexity $s(n)$ if $A$'s solution is at most $c$ times worse than optimal and $A$ accesses at most the first $s(n)$ bits of $\phi$ for any input of length $n$ (see e.g., [19] for formal definitions).

Up to the present, several online problems have been analyzed using the advice complexity, such as the paging problem [6], the $k$-server problem [5, 18], the knapsack problem [7], the buffer management problem [14], the graph coloring problem [2, 3], the string guessing problem [4], the graph exploration problem [13], and the independent set problem [12].

One of the fundamental questions on the advice complexity is whether or not an optimal strategy is to completely specify the behavior of an optimal offline algorithm. For example, in the knapsack problem, the best known strategy is to specify whether the current item should be packed or not using 1 bit per request, resulting in the advice complexity of $n$ where $n$ is the total number of items, although it is not known whether this is optimal; the current lower bound on the advice complexity is $n-1$, leaving a gap of one bit [7]. In contrast, if we do the same thing for paging, we need $\lceil \log_2 K \rceil$ bits of advice to specify which page to evict at a page fault, where $K$ is the size of the buffer. This results in the advice complexity of $n\lceil \log_2 K \rceil$ for instances with $n$ requests, whereas it is shown that $n + K$ bits of advice suffice [11, 6].

*1.2. Online Bipartite Matching and Online Stable Marriage*

The *online bipartite matching problem*, introduced by Karp et al. [22], is defined as follows: An instance is a bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$. Vertices in $U$ are initially present and vertices in $V$ arrive one by one. When a vertex $v \in V$ arrives, all the edges incident to $v$ are also revealed. An online algorithm chooses one of the unmatched neighbors (if any) of $v$ to be matched with $v$, or leaves $v$ unmatched. This decision must be made before the next vertex arrives and may not be revoked later. The goal of the problem is to construct a matching as large as possible. It is easy to see that a greedy algorithm always returns a maximal matching and is 1/2-competitive. Karp et al. [22] proposed a randomized algorithm Ranking, and proved that it is $(1 - 1/e)$-competitive. They also proved that no randomized online algorithm can do better than this. This problem has recently received considerable attention due to its applicability to advertisement space matching (see [26, 17, 1, 25, 21, 24, 9, 10] for example).

The *stable marriage problem* is a classical combinatorial problem proposed by Gale and Shapley in their celebrated seminal paper [16]. Its instances consist of the same number $n$ of men and women. Each man submits a *preference list* which ranks all the women according to his preference. Similarly, each woman submits a preference list of men. A matching is a one-to-one correspondence between men and women. For a matching $M$, let $M(p)$ denote the partner of a person $p$. If man $m$ prefers woman $w$ to $M(m)$ and $w$ prefers $m$ to $M(w)$, then we say that $(m, w)$ is a *blocking pair* for $M$. A matching with no blocking pair is *stable*. Gale and Shapley proved that any instance admits at least one stable matching and proposed a simple algorithm to find one [16].

An online variant of the stable marriage problem was considered by Khuller et al. [23]. In their model, the women's preference lists are open to an online algorithm in advance, while the men arrive one by one, revealing his preference list. Upon arrival of a man, the algorithm must match him to one of the currently unassigned women. The purpose of the algorithm is to construct a matching so that the number of blocking pairs is minimized. Since, as mentioned above, there always exists a stable matching, the optimal cost is always zero. Therefore, the competitive analysis does not fit this problem and they evaluated online algorithms merely by the number of blocking pairs. They showed that any online algorithm produces a matching with $\Omega(n^2)$ blocking pairs in the worst case, but a greedy algorithm called First-Come-First-Served obtains a matching with $O(n \log n)$ blocking pairs on average.

### *1.3. Our Contribution*

In this paper, we study the advice complexity of the online bipartite matching problem and the online stable marriage problem. We show that, for both problems, $\lceil \log_2(n!) \rceil$ bits of advice are necessary and sufficient for a deterministic online algorithm to be optimal. This shows that in these problems, the best strategy is to guide the online algorithm by completely specifying the optimal offline algorithm's behavior. To the best of our knowledge, the exact advice complexity is known only for the ski rental problem [11], Winner-takes-all [5], and the string guessing problem [4], for which one bit of advice, $n$ bits of advice, and $\lceil n \log_2 q \rceil$ bits of advice, respectively, are necessary and sufficient, where $q$ denotes the size of the alphabet for the string.

## 2. Main Results

### *2.1. Online Bipartite Matching*

**Theorem 2.1.** *There is a deterministic optimal online algorithm for the online bipartite matching problem that uses $\lceil \log_2(n!) \rceil$ bits of advice.*

*Proof.* Let $G = (U, V, E)$ be the bipartite graph revealed at the end of the input, and $M^*$ be a maximum matching of $G$. Let $\tilde{M}^*$ be a perfect matching obtained by extending $M^*$, that is, if $M^*$ is already a perfect matching, then $\tilde{M}^* = M^*$, otherwise, $\tilde{M}^*$ is a result of arbitrarily adding edges between unmatched vertices $u (\in U)$ and $v (\in V)$. Note that edges in $\tilde{M}^* \setminus M^*$ do not exist in $E$ since $M^*$ is maximum.

The oracle presents $\tilde{M}^*$ as advice. Since $|U| = |V| = n$, there are $n!$ perfect matchings and hence $\lceil \log_2(n!) \rceil$ bits suffices for this purpose. An online algorithm acts as follows: Suppose that a vertex $v \in V$ arrives and $v$ is matched with $u$ in $\tilde{M}^*$. If $(u, v) \in E$, then the algorithm matches $v$ to $u$. Otherwise, it leaves $v$ unmatched. It is not hard to see that the matching obtained by this algorithm is exactly $M^*$. □

**Theorem 2.2.** *No deterministic online algorithm for the online bipartite matching problem that uses less than $\lceil \log_2(n!) \rceil$ bits of advice can be optimal.*

*Proof.* We construct $n!$ instances in such a way that (1) each instance admits a unique perfect matching and (2) no two instances share the same perfect matching. For a better exposition, we denote the vertices in $U$ as $1, 2, \ldots, n-1, n$ and do the same for $V$. The vertices in $V$ arrive in the order $n, n-1, \ldots, 2, 1$. To determine the edge set, we associate a permutation of $1, 2, \ldots, n$ to each instance. There are $n!$ permutations and each one of them is associated with one of $n!$ instances. In an instance $G_\Pi$ associated with permutation $\Pi$, each vertex $i \in V$ is adjacent to $\Pi(1), \Pi(2), \ldots, \Pi(i)$ ($\in U$). It is easy to show by induction that $G_\Pi$ has the unique perfect matching in which the vertex $i \in V$ is matched to the vertex $\Pi(i) \in U$ for each $i$, and hence the above conditions (1) and (2) are satisfied.

Let $A$ be an online algorithm that uses less than $\lceil \log_2(n!) \rceil$ bits of advice. Then $A$ must use the same advice string for two different instances constructed as above, say $G_{\Pi_1}$ and $G_{\Pi_2}$. Let $i^*$ be the largest integer $i$ such that $\Pi_1(i) \neq \Pi_2(i)$. Then $\Pi_1(i) = \Pi_2(i)$ for $i^* + 1 \leq i \leq n$ and hence by construction, each vertex $i^*$ through $n$ (of $V$) has the same neighbors in $G_{\Pi_1}$ and $G_{\Pi_2}$. Therefore, when $A$ decides the vertex of $U$ to be matched with $i^*$, the information it has received so far is the edges incident to $i^*$ through $n$ (of $V$) and the contents of the advice string, that are the same for $G_{\Pi_1}$ and $G_{\Pi_2}$. Therefore, $A$ matches $i^*$ to the same vertex in $G_{\Pi_1}$ and $G_{\Pi_2}$, but $i^*$ must be matched to different vertices in the unique perfect matchings of $G_{\Pi_1}$ and $G_{\Pi_2}$ because $\Pi_1(i^*) \neq \Pi_2(i^*)$. This completes the proof. □

### *2.2. Online Stable Marriage*

**Theorem 2.3.** *There is a deterministic optimal online algorithm for the online stable marriage problem that uses* $\lceil \log_2(n!) \rceil$ *bits of advice.*

*Proof.* Since there are $n$ men and $n$ women, there are $n!$ perfect matchings, and as mentioned before, at least one of them is stable. The oracle specifies one of the stable matchings using $\lceil \log_2(n!) \rceil$ bits and the online algorithm follows this advice. □

**Theorem 2.4.** *No deterministic online algorithm for the online stable marriage problem that uses less than* $\lceil \log_2(n!) \rceil$ *bits of advice can be optimal.*

*Proof.* The proof goes like that of Theorem 2.2. We will construct $n!$ instances in such a way that (1) each instance admits a unique stable matching and (2) no two instances share the same stable matching. We denote the men as $1, 2, \ldots, n-1, n$ and do the same for the women. The women's preference lists are identical; each woman prefers men in an increasing order of the indices, i.e., the man 1 is the most preferred and the man $n$ is the least preferred. This guarantees the condition (1) above; it is not hard to see that the unique stable matching can be obtained by assigning the man 1 to his most preferred woman, assigning the man 2 to his most preferred woman among those who are currently unassigned, and so on.

The men arrive in the reverse order of the indices, that is, the man $n$ arrives first, the man $n-1$ arrives next, and so on. To determine the men's preference lists, we associate a permutation of $1, 2, \ldots, n$ to each instance. There are $n!$ permutations and each one of them is associated with one of $n!$ instances. We construct an instance $I_\Pi$ associated with permutation $\Pi$ in such a way that in the unique stable matching for $I_\Pi$, each man $i$ is matched with the woman $\Pi(i)$. Then, clearly the condition (2) is satisfied.

Now we show how to construct the men's preference lists of $I_\Pi$. Let $X$ be the set of women and for $1 \le i \le n$, let $X_\Pi(i) = \{\Pi(1), \Pi(2), \ldots, \Pi(i)\}$. Also, for a set $J$ of integers, let $[J]$ denote the ordered list of the elements of $J$ in an increasing order. Now the man $i$'s preference list is

$$i : [X_\Pi(i)] \; [X \setminus X_\Pi(i)].$$

That is, in the man $i$'s preference list, the women in $X_\Pi(i)$ first appear in an increasing order of indices, and then any remaining women appear in an increasing order of indices. For a better exposition, we give an example of $n = 6$ for a permutation $\Pi$ that maps 1, 2, 3, 4, 5, and 6 to 3, 1, 6, 4, 2, and 5, respectively. The men's preference lists are as follows, where a vertical bar is used to show the boundary between $[X_\Pi(i)]$ and $[X \setminus X_\Pi(i)]$.

1 : 3 | 1 2 4 5 6
2 : 1 3 | 2 4 5 6
3 : 1 3 6 | 2 4 5
4 : 1 3 4 6 | 2 5
5 : 1 2 3 4 6 | 5
6 : 1 2 3 4 5 6 |

We show by induction that $I_\Pi$ satisfies the aforementioned condition, that is, each man $i$ is matched with the woman $\Pi(i)$ in the unique stable matching for $I_\Pi$. As mentioned previously, the man 1 is matched with the top woman in his list, and hence 1 is matched with $\Pi(1)$ as required. Assuming that the statement is true for men 1 through $i-1$, we consider the man $i$'s partner. The first $i$ women in man $i$'s preference list are exactly $X_\Pi(i)$. Among them, the women in $X_\Pi(i-1)$ are already matched with men 1 through $i-1$ by the induction hypothesis. Since the man $i$ is matched with the most preferred unmatched woman, he must be matched with the woman $\Pi(i)$, who is the unique woman in $X_\Pi(i) \setminus X_\Pi(i-1)$.

Let $A$ be an online algorithm that uses less than $\lceil \log_2(n!) \rceil$ bits of advice. Then $A$ must use the same advice string for two different instances constructed as above, say $I_{\Pi_1}$ and $I_{\Pi_2}$. Let $i^*$ be the largest integer $i$ such that $\Pi_1(i) \ne \Pi_2(i)$. Then $X_{\Pi_1}(i) = X_{\Pi_2}(i)$ for $i^* + 1 \le i \le n$ and hence by construction, the preference lists of men $i^*$ through $n$ are the same in $I_{\Pi_1}$ and $I_{\Pi_2}$. Therefore, when $A$ decides the partner of the man $i^*$, the information it has received so far is the preference lists of men $i^*$ through $n$ and the contents of the advice string, that are the same for $I_{\Pi_1}$ and $I_{\Pi_2}$. Therefore, $A$ assigns the same woman to the man $i^*$ in $I_{\Pi_1}$ and $I_{\Pi_2}$, but $i^*$ must be assigned different women in their respective stable matchings because $\Pi_1(i^*) \ne \Pi_2(i^*)$. This completes the proof. □

## 3. Conclusion

In this paper, we studied the advice complexity of the online bipartite matching problem and the online stable marriage problem. We considered the number of advice bits needed for deterministic online algorithms to be optimal, and gave the exact bounds for both problems.

There are several possible extensions of this research, which include considering the relationship between the number of available advice bits and the competitive ratio, and parameterizing the advice complexity by, e.g., the number of edges in an input graph.

**Acknowledgments.** The author would like to thank the reviewers for their helpful comments. This work was supported by JSPS KAKENHI Grant Number 24500013.

## References

[1] G. Aggarwal, G. Goel, C. Karande, and A. Mehta, "Online vertex-weighted bipartite matching and single-bid budgeted allocations, Proc. SODA 2011, pp. 1253-1264.

[2] M. P. Bianchi, H.-J. Böckenhauer, J. Hromkovič, and L. Keller, "Online coloring of bipartite graphs with and without advice," Proc. COCOON 2012, pp. 519–530, 2012.

[3] M. P. Bianchi, H-J. Böckenhauer, J. Hromkovič, S. Krug, and B. Steffen, "On the advice complexity of the online L(2,1)-coloring problem on paths and cycles," Proc. COCOON 2013, pp. 53–64, 2013.

[4] H.-J. Böckenhauer, J. Hromkovič, D. Komm, S. Krug, J. Smula, A. Sprock, "The string guessing problem as a method to prove lower bounds on the advice complexity," Proc. COCOON 2013, pp. 493–505, 2013.

[5] H.-J. Böckenhauer, D. Komm, R. Královič, and R. Královič, "On the advice complexity of the $k$-server problem," Proc. ICALP 2011, LNCS 6755, pp. 207–218, 2011.

[6] H.-J. Böckenhauer, D. Komm, R. Královič, R. Královič, and T. Mömke, "On the advice complexity of online problems," Proc. ISAAC 2009, LNCS 5878, pp. 331–340, 2009.

[7] H.-J. Böckenhauer, D. Komm, R. Královič, and P. Rossmanith, "On the advice complexity of the knapsack problem," Proc. LATIN 2012, LNCS 7256, pp. 61–72, 2012.

[8] J. Boyar and L. M. Favrholdt, "The relative worst order ratio for online algorithms," ACM Transactions on Algorithms 3(2), 2007.

[9] N. R. Devanur and K. Jain, "Online matching with concave returns," Proc. STOC 2012, pp. 137-144, 2012.

[10] N. R. Devanur, K. Jain, and R. D. Kleinberg, "Randomized primal-dual analysis of RANKING for online bipartite matching," Proc. SODA 2013, pp. 101-107, 2013.

[11] S. Dobrev, R. Královič, and D. Pardubská, "Measuring the problem-relevant information in input," *RAIRO - Theoretical Informatics and Applications* 43(3), pp. 585–613, 2009.

[12] S. Dobrev, R. Královič, and R. Královič, "Independent set with advice: the impact of graph knowledge," Proc. WAOA 2012, LNCS 7846, pp. 2–15, 2013.

[13] S. Dobrev, R. Královic, and E. Markou, "Online graph exploration with advice," Proc. SIROCCO 2012, LNCS 7355, pp. 267–278, 2012.

[14] R. Dorrigiv, M. He, and N. Zeh, "On the advice complexity of buffer management," Proc. ISAAC 2012, LNCS 7676, pp. 136–145, 2012.

[15] Y. Emek, P. Fraigniaud, A. Korman, and A. Rosén, "Online computation with advice," Proc. ICALP 2009, LNCS 5555, pp. 427–438, 2009.

[16] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, Vol. 69, pp. 9–15, 1962.

[17] G. Goel and A. Mehta, "Online budgeted matching in random input models with applications to Adwords," Proc. SODA 2008, pp. 982-991, 2008.

[18] S. Gupta, S. Kamali, and A. López-Ortiz, "On advice complexity of the $k$-server problem under sparse metrics," Proc. SIROCCO 2013, LNCS 8179, pp. 55–67, 2013.

[19] J. Hromkovič, R. Královič, and R. Královič, "Information complexity of online problems," Proc. MFCS 2010, LNCS 6281, pp. 24–36, 2010.

[20] B. Kalyanasundaram and K. Pruhs, "Speed is more powerful than clairvoyance," Proc. FOCS'95, pp. 214–221, 1995.

[21] C. Karande, A. Mehta, and P. Tripathi, "Online bipartite matching with unknown distributions," Proc. STOC 2011, pp. 587-596, 2011.

[22] R. M. Karp, U. V. Vazirani, and V. V. Vazirani, "An optimal algorithm for on-line bipartite matching," Proc. STOC'90, pp. 352-358, 1990.

[23] S. Khuller, S.G. Mitchell, and V. V. Vazirani, "On-line algorithms for weighted bipartite matching and stable marriages," Theoretical Computer Science, 127(2), pp. 255–267, 1994.

[24] M. Mahdian and Q. Yan, "Online bipartite matching with random arrivals: an approach based on strongly factor-revealing LPs," Proc. STOC 2011, pp. 597-606, 2011.

[25] V. H. Manshadi, S. O. Gharan, and A. Saberi, "Online stochastic matching: online actions based on offline statistics," Proc. SODA 2011, pp. 1285-1294, 2011.

[26] A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani, "AdWords and generalized online matching," J. ACM 54(5), Article No. 22, 2007.

[27] D.D. Sleator and R.E. Tarjan, "Amortized efficiency of list update and paging rules," Communications of the ACM 28(2), pp. 202–208, 1985.