Advanced Algorithms

Lecture 12: Beating Greedy for Online Bipartite Matching

Lecturers: Steve Choi and Nathan Klein *Scribe: Steve Choi*

1 Recap of Online Bipartite Matching

Remember we are studying the question: how do we maintain a good solution to a problem when the *input is not given to us ahead of time?* And in particular, we're studying the problem of online bipartite matching.

Definition 1.1 (Online Bipartite Matching)**.** *An instance $\mathcal{I}$ of online bipartite matching consists of a bipartite graph $G = (\textsc{Off} \uplus \textsc{On}, E)$ and an ordering $v_1, \ldots, v_{|\textsc{On}|}$ of the online nodes so that node v_i arrives at timestep i. At each timestep i, the algorithm must decide what node in $\textsc{Off}$ to match $v_i \in \textsc{On}$ to, if any, using only the knowledge of the set of nodes $\textsc{Off}$ and the edges $\bigcup_{j=1}^{i} \delta(v_i)$.*

And we want to find an algorithm $\mathcal{A}$ with a competitive ratio α at close to 1 as possible. Remember that if $\mathcal{A}(\mathcal{I})$ is the set of edges in the matching returned by $\mathcal{A}$ on instance $\mathcal{I}$ and $M(\mathcal{I})$ is the edges in the maximum matching in $\mathcal{I}$, then the competitive ratio of $\mathcal{A}$ is:

$$\alpha = \inf_{\mathcal{I}} \frac{\mathbb{E}\left[|\mathcal{A}(\mathcal{I})|\right]}{|M(\mathcal{I})|}$$

where the expectation is over the randomness in the algorithm $\mathcal{A}$. We showed that the greedy algorithm is $\frac{1}{2}$ competitive. A natural question is whether this is optimal.

2 The Red-Blue Algorithm

In this lecture, we will show that if we color every offline node red or blue independently and run the greedy algorithm in which online nodes *prefer to match to red nodes*, then the competitive ratio improves to $\frac{5}{9}$. This algorithm is a simplification of the so-called ranking algorithm by Karp, Vazirani, and Vazirani [KVV90] and was analyzed by Dürr, Konrad, and Renault [DKR16]. We thank Zhuan Khye Koh for helping with the style of proof presented here.

As in the primal-dual analysis of greedy, if we get a match, we will set the dual values y_u and y_v using a potential function $g : \{\text{red}, \text{blue}\} \to [0, 1]$ so that we increase the dual by 1 whenever we add an edge to the matching. We will decide the function later, but it will simply indicate how we decide to "split" the match value of 1 on the dual variables.

Theorem 2.1. *The Red-Blue Algorithm achieves an expected competitive ratio of at least $\frac{5}{9}$.*

Proof. First, we establish how we prove an α-competitive ratio using the dual LP formulation.

Lemma 2.2. *For some $\alpha \in [0, 1]$, if $\mathbb{E}[y_u + y_v] \geq \alpha$, then $\mathbb{E}[|M|] \geq \alpha \cdot OPT$.*

Proof. Consider any matching $|M|$. We have

$$|M| = \sum_{e \in M} 1 = \sum_{\{u,v\} \in M} y_u + y_v \qquad \text{(since } y_u + y_v = (1 - g(\cdot)) + g(\cdot) = 1\text{)}$$
$$= \sum_{u \in \textsc{On}} y_u + \sum_{v \in \textsc{Off}} y_v. \qquad \text{(since } y_u, y_v = 0 \text{ if not matched)}$$

Algorithm 1 The Red-Blue Algorithm

Input: $G = (\textsc{Off} \uplus \textsc{On}, E)$.
for each $v \in \textsc{Off}$ **do**
  Color v red with probability $\frac{1}{2}$ and blue otherwise. Let $C_v \in \{\text{red}, \text{blue}\}$ be the color of v.
end for
Initialize all dual variables to 0.
for each arrival $u \in \textsc{On}$ **do**
  Let $UN(u)$ be the set of unmatched neighbors of u. If $UN(u) = \varnothing$, skip to the next vertex.
  Match u to any red node if available. If not, match to any blue node. Let v be the node matched to u.
  Set $y_u = 1 - g(C_v)$ and $y_v = g(C_v)$.
end for
return the matching M

Which means, if we take the expectation of $|M|$, we have

$$\begin{aligned}\mathbb{E}[|M|] &= \mathbb{E}\left[\sum_{u \in \textsc{On}} y_u + \sum_{v \in \textsc{Off}} y_v\right] \\ &= \sum_{u \in \textsc{On}} \mathbb{E}[y_u] + \sum_{v \in \textsc{Off}} \mathbb{E}[y_v] \qquad \text{(by linearity of expectation)}\end{aligned}$$

From the lemma statement, we assumed

$$\mathbb{E}[y_u + y_v] \geq \alpha \iff \frac{1}{\alpha}\mathbb{E}[y_u + y_v] \geq 1$$

for every edge. This implies the vector $\frac{1}{\alpha}\mathbb{E}[\vec{y}]$ satisfies all the dual LP constraints and forms a feasible dual solution, and we have

$$\frac{1}{\alpha}\left(\sum_{u \in \textsc{On}} \mathbb{E}[y_u] + \sum_{v \in \textsc{Off}} \mathbb{E}[y_v]\right) \geq OPT_D \geq OPT_P = OPT,$$

since a feasible dual solution is at least as large as the optimal dual solution OPT_D, and the optimal dual value gives an upper bound on the primal solution OPT_P, which is then equal to OPT as we proved in homework 2. Then by rearranging the inequality, we prove the lemma. □

So now, we just have to prove that for all edges $\{u, v\} \in E$, the expected value of the dual values $y_u + y_v$ is at least $\frac{5}{9}$.

Consider an edge $\{u, v\} \in E$. The idea of the proof is to construct an alternate scenario, where node v is excluded, then add back v and randomize over C_v. Before we jump into the lemma and the proof, we will fix some notation as follows:

- C^{-v}: the set of colors of the nodes in $\textsc{Off} - \{v\}$.
- G^{-v}: the graph G without node v.
- $\mathrm{RB}(G^{-v}, C^{-v})$: the alternate setting where v is excluded and C^{-v} is fixed.

Lemma 2.3. *For any $v \in \textsc{Off}$, we have*

$$\mathop{\mathbb{E}}_{C_v \sim \{red, blue\}}[y_u + y_v \mid C^{-v}] = \frac{5}{9}$$

Proof. Consider fixing some realization of the colors in C^{-v}. Then, one of the three cases are possible:

1. u is not matched.
2. u is matched to a red node.
3. u is matched to a blue node.

We will analyze the dual values of each case separately, then find a global minimum.

Case 1: u is not matched.

If u is not matched in $\mathrm{RB}(G^{-v}, C^{-v})$, then adding v back to G^{-v} will always cause v to be matched in the original setting. This could be with u, or a neighbor of v that arrives before. Thus, we have

$$\mathop{\mathbb{E}}_{C_v \sim \{\text{red,blue}\}}[y_v \mid C^{-v}] = \frac{1}{2}g(\text{red}) + \frac{1}{2}g(\text{blue}). \tag{1}$$

Note that in this case, we can only give a trivial lower bound of 0 for y_u, since we cannot guarantee that u will be matched.

Case 2: u is matched to a red node.

If u is matched to a red node in $\mathrm{RB}(G^{-v}, C^{-v})$, then adding v back to G^{-v} will not change the color match of u in the original setting. In other words, u will still always be matched to a red node in the original setting. Thus, we have

$$\mathop{\mathbb{E}}_{C_v \sim \{\text{red,blue}\}}[y_u \mid C^{-v}] \geq 1 - g(\text{red}). \tag{2}$$

Again, in this case, we cannot guarantee that v will be matched, so we can only apply a trivial lower bound of 0 for y_v.

Case 3: u is matched to a blue node.

If u is matched to a blue node in $\mathrm{RB}(G^{-v}, C^{-v})$, adding v back may have different outcomes. If we add v back and C_v is red, we can guarantee that v will be matched. Since we also have that u was originally matched to a blue, we can also guarantee that u will be matched. However, if v is blue, we cannot guarantee that v will be matched. Before we give the actual lower bound, we must first make the following claim:

Claim 2.4. *For the potential function g, we must have*

$$g(red) < g(blue).$$

We will later prove the claim, as if this condition is false, we cannot achieve a competitive ratio better than $\frac{1}{2}$.

Using the claim, we have

$$\mathop{\mathbb{E}}_{C_v \sim \{\text{red,blue}\}}[y_u + y_v \mid C^{-v}] \geq \frac{1}{2}(1 - g(\text{blue})) + \frac{1}{2}(g(\text{red}) + 1 - g(\text{blue})), \tag{3}$$

where we used the fact that $g(\text{red}) \le g(\text{blue})$ to get the lower bounds.

We can now set the three expressions equal to each other to solve for a global lower bound. By (1) and (3), we have

$$\frac{1}{2}g(\text{blue}) + \frac{1}{2}g(\text{red}) = 1 - g(\text{blue}) + \frac{1}{2}g(\text{red}) \iff \frac{3}{2}g(\text{blue}) = 1 \iff g(\text{blue}) = \frac{2}{3}.$$

Next, using (2), (3), and $g(\text{blue}) = \frac{2}{3}$, we have

$$1 - g(\text{red}) = 1 - g(\text{blue}) + \frac{1}{2}g(\text{red}) \iff g(\text{blue}) = \frac{3}{2}g(\text{red}) = \frac{2}{3},$$

which gives $g(\text{red}) = \frac{4}{9}$.

This means that we have a global lower bound of

$$\mathop{\mathbb{E}}_{C_v \sim \{\text{red,blue}\}} [y_u + y_v \mid C^{-v}] \ge 1 - g(\text{red}) = \frac{5}{9}. \qquad \square$$

Now, we can finalize the proof. We know that given any choice of C^{-v}, we have $\mathbb{E}_{C_v}[y_u + y_v \mid C^{-v}] \ge \frac{5}{9}$. This means that, by law of total expectation, we have

$$\mathbb{E}\left[y_u + y_v\right] = \mathop{\mathbb{E}}_{C^{-v}}\left[\mathop{\mathbb{E}}_{C_v \sim \{\text{red,blue}\}} [y_u + y_v \mid C^{-v}]\right] = \mathop{\mathbb{E}}_{C^{-v}}\left[\frac{5}{9}\right] = \frac{5}{9}.$$

Thus, the Red-Blue Algorithm has a competitive ratio of $\frac{5}{9}$. $\square$

So why can we not have $g(\text{red}) \ge g(\text{blue})$? It turns out, if we assume this, we cannot hope to achieve a competitive ratio better than $\frac{1}{2}$.

Proof of Claim 2.4. Suppose $g(\text{red}) \ge g(\text{blue})$. Then, we will establish a global lower bound by setting (3) equal to (4). Setting the two expressions equal, we have

$$\begin{aligned} & \frac{1}{2}g(\text{red}) + \frac{1}{2}g(\text{blue}) = 1 - g(\text{red}) \\ \iff\ & \frac{3}{2}g(\text{red}) + \frac{1}{2}g(\text{blue}) = 1 \\ \iff\ & g(\text{red}) \ge g(\text{blue}) = 2 - 3g(\text{red}) && \text{(by assumption)} \\ \iff\ & g(\text{red}) \ge \frac{1}{2}. \end{aligned}$$

Plugging this value of $g(\text{red})$ to (4), we get

$$1 - g(\text{red}) \le \frac{1}{2}.$$

Thus, if $g(\text{red}) \ge g(\text{blue})$, we cannot prove a competitive ratio better than $\frac{1}{2}$. $\square$

2.1 Ranking and Upper Bound

We have shown an algorithm with competitive ratio $\frac{5}{9}$. It turns out that by letting the number of colors go to infinity, the ratio tends to $1 - \frac{1}{e}$ and corresponds to the so-called *ranking* algorithm of Karp, Vazirani, and Vazirani [KVV90]. In Appendix A, we show why the ranking algorithm achieves a ratio of $1 - \frac{1}{e}$. The proof is quite similar in spirit to what we just did. You do not need to know this proof, but it's there in case you want to see the analysis.

It turns out $1 - \frac{1}{e}$ is the best possible competitive ratio. [KVV90] presented the following input instance and proved that no randomized algorithm can do better than $(1 - \frac{1}{e})$-competitive ratio.

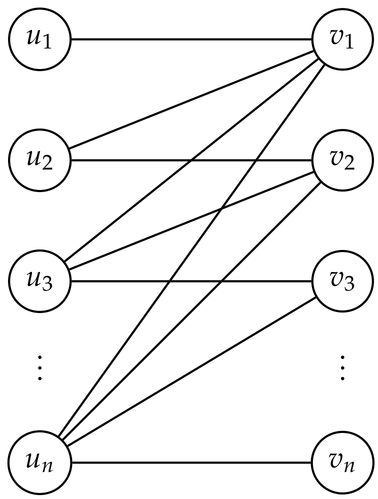

Figure 1: Hard input example from [KVV90].

References

[DJK13] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. "Randomized Primal-Dual Analysis of RANKING for Online Bipartite Matching". In: *Proceedings of the 2013 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2013, pp. 101–107. DOI: `10.1137/1.9781611973105.7`. eprint: `https://epubs.siam.org/doi/pdf/10.1137/1.9781611973105.7` (cit. on p. 6).

[DKR16] Christoph Dürr, Christian Konrad, and Marc Renault. "On the Power of Advice and Randomization for Online Bipartite Matching". In: *24th Annual European Symposium on Algorithms (ESA 2016)*. Ed. by Piotr Sankowski and Christos Zaroliagis. Vol. 57. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016, 37:1–37:16. ISBN: 978-3-95977-015-6. DOI: `10.4230/LIPIcs.ESA.2016.37` (cit. on p. 1).

[KVV90] R.M. Karp, U.V. Vazirani, and V.V. Vazirani. "An optimal algorithm for online bipartite matching". In: *STOC*. 1990 (cit. on pp. 1, 5, 6).

A The Ranking Algorithm

Instead of the original proof by [KVV90], we will go over a much simpler analysis, due to Devanur, Jain, and Kleinberg [DJK13].

A.1 Competitive Ratio of Ranking

As mentioned earlier, for every offline node, we will sample $Y_v \sim \text{Unif}(0,1)$, then on the arrival of an online node, we will match to the smallest label possible. But, we will add an extra step in the algorithm; if we get a match, we will set the dual values y_u and y_v to $1 - g(Y_v)$ and $g(Y_v)$, respectively, where $g : [0,1] \to [0,1]$ is some *potential function*. This will help us later when we analyze the competitive ratio of the algorithm. You can think of g as how we decide to "split" the match value of 1 on the dual values. Throughout the proof, we will list the properties that g needs to satisfy to make the proof work, and towards the end, we will show how we choose this function g.

Algorithm 2 The Ranking Algorithm [KVV90; DJK13]

```
Input: Graph G = (Off ⊎ On, E).
for each v ∈ Off do
    Sample Y_v ~ Unif(0, 1).
end for
Initialize all dual variables to 0.
for each arrival u ∈ On do
    Let UN(u) be the set of unmatched neighbors of u. If UN(u) = ∅, skip to the next vertex.
    Match u to v := arg min_{w∈UN(u)} Y_w.
    Set y_u = 1 − g(Y_v) and y_v = g(Y_v).
end for
return the match set M
```

We can see that this algorithm is called "Ranking", since sampling $Y_v \sim \text{Unif}(0,1)$ imposes a ranking on the offline vertices.

Theorem A.1 ([KVV90])**.** *The Ranking Algorithm achieves an expected competitive ratio of* $1 - \frac{1}{e}$.

Proof. First, we establish how we prove an α-competitive ratio using the dual LP formulation.

Lemma A.2. *For some* $\alpha \in (0,1)$*, if* $\mathbb{E}[y_u + y_v] \geq \alpha$*, then* $\mathbb{E}[|M|] \geq \alpha \cdot OPT$.

Proof. Consider any matching $|M|$. We have

$$|M| = \sum_{e \in M} 1 = \sum_{\{u,v\} \in M} y_u + y_v \qquad \text{(since } y_u + y_v = (1 - g(Y_v)) + g(Y_v) = 1\text{)}$$

$$= \sum_{u \in \text{On}} y_u + \sum_{v \in \text{Off}} y_v. \qquad \text{(since } y_u, y_v = 0 \text{ if not matched)}$$

Which means, if we take the expectation of $|M|$, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{Y \sim [0,1]^{|\textsc{Off}|}}[|M|] &= \mathbb{E}\left[\sum_{u \in \textsc{On}} y_u + \sum_{v \in \textsc{Off}} y_v\right] \\ &= \sum_{u \in \textsc{On}} \mathbb{E}[y_u] + \sum_{v \in \textsc{Off}} \mathbb{E}[y_v] \qquad \text{(by linearity of expectation)}\end{aligned}$$

From the lemma statement, we assumed

$$\mathbb{E}[y_u + y_v] \geq \alpha \iff \frac{1}{\alpha}\mathbb{E}[y_u + y_v] \geq 1$$

for every edge. This implies the vector $\frac{1}{\alpha}\mathbb{E}[\vec{y}]$ satisfies all the dual LP constraints and forms a feasible dual solution, and we have

$$\frac{1}{\alpha}\left(\sum_{u \in \textsc{On}} \mathbb{E}[y_u] + \sum_{v \in \textsc{Off}} \mathbb{E}[y_v]\right) \geq OPT_D \geq OPT_P = OPT,$$

since a feasible dual solution is at least as large as the optimal dual solution OPT_D, and the optimal dual value gives an upper bound on the primal solution OPT_P, which is then equal to OPT as we proved in homework 2. Then by rearranging the inequality, we prove the lemma. □

So now, we just have to prove that for all edges $\{u, v\} \in E$, the expected value of the dual values $y_u + y_v$ is at least $1 - \frac{1}{e}$.

Consider an edge $\{u, v\} \in E$. The idea of the proof is to construct an alternate scenario, where node v is excluded, then add back v and randomize over Y_v.

Before we proceed with the detailed proof, we will define some notations below:

- Superscript $-v$: Variables with $-v$ superscript means in the alternate scenario, where v is excluded. For example, G^{-v} is the graph without node v.
- Ranking(G^{-v}): The alternate scenario where v is excluded.
- $h(u)$: The hypothetical matching of u in Ranking(G^{-v}), given that we fix an arbitrary Y^{-v}.

Throughout the proof, I will place subscripts on $\mathbb{E}$ and $\mathbb{P}$, since it is helpful to know what we randomize over.

First, it is possible that $h(u) = \varnothing$. In this case, we can define $Y_{h(u)} = 1$. Since we do not get any match value if $h(u) = 1$, we want to set $y_u = 1 - g(Y_{h(u)}) = 0$. This means that we must have the following property for g:

- $g(1) = 1$.

This will be more clear when we look at the following lemma.

Lemma A.3. *For any $v \in \textsc{Off}$, we have*

$$\mathop{\mathbb{E}}_{Y_v \sim [0,1]}[y_v \mid Y^{-v}] \geq \int_0^{Y_{h(u)}} g(x)\mathrm{d}x.$$

Proof. The main idea of the proof is that, if $Y_v < Y_{h(u)}$, vertex v will always be matched–either by u or an earlier node. Since we cannot guarantee that v will be matched if $Y_v > Y_{h(u)}$, we will instead give a lower bound.

Let $f(x)$ be the PDF of $Y_v \sim \text{Unif}(0,1)$. Then, the rest of the proof simply follows, since

$$\begin{aligned}
\mathop{\mathbb{E}}_{Y_v \sim [0,1]}[y_v \mid Y^{-v}] &= \int_0^1 g(x) \cdot \mathop{\mathbb{P}}_{Y_v \sim [0,1]}[v \in M \mid Y^{-v} \cap Y_v = x] \cdot f(x|Y^{-v})dx && \text{(by LOTUS)} \\
&\geq \int_0^{Y_{h(u)}} g(x) \cdot \mathop{\mathbb{P}}_{Y_v \sim [0,1]}[v \in M \mid Y^{-v} \cap Y_v = x] \cdot f(x|Y^{-v})dx \\
&= \int_0^{Y_{h(u)}} g(x) \cdot f(x|Y^{-v})dx && (Y_v < Y_{h(u)} \text{ implies } v \text{ is matched}) \\
&= \int_0^{Y_{h(u)}} g(x)dx. && (\text{since } Y_v \sim \text{Unif}(0,1))
\end{aligned}$$

This proves the lemma. □

Lemma A.4. *For any $v \in \textsc{Off}$, we have*

$$\mathop{\mathbb{E}}_{Y_v \sim [0,1]}[y_u \mid Y^{-v}] \geq 1 - g(Y_{h(u)}).$$

Proof. Consider the alternate scenario where we exclude v. Intuitively, adding node v back does not make the algorithm perform any worse, i.e. it only helps node u find an open neighbor.

Let's formally prove this. Let $UM_t \subseteq \textsc{Off}$ and $UM_t^{-v} \subseteq \textsc{Off} - \{v\}$ be the set of unmatched nodes prior to timestep t, for the original setting and the alternate setting, respectively. We claim that up to the arrival of u, we have

$$UM_t \supseteq UM_t^{-v}.$$

We prove this inductively. Imagine we run the two algorithms in parallel. The base case is trivially true, since before the arrival of any vertex, we have

$$\textsc{Off} = UM_t \supseteq UM_t^{-v} = \textsc{Off} - \{v\}.$$

Now, assume this is true for some timestep t. We split into two cases: For some $v' \in \textsc{Off}$, both scenarios simultaneously match v', or v' is matched in the original setting, but not in the alternate setting.

The former case subtracts v' from both UM_t and UM_t^{-v}, which still satisfies the claim. The latter case can happen only if v' was not in UM_t^{-v}. By our inductive hypothesis $UM_t \supseteq UM_t^{-v}$, the claim is not violated by subtracting v' from UM_t. This proves the inductive step.

Let $M(u)$ be the match we get for u. This allows us to conclude

$$Y_{M(u)} \leq Y_{h(u)},$$

since the label of the match we get for u in the original setting will be $Y_{h(u)}$ at worst.

Before we conclude the lemma, we must have the following property for g:

- **Property 2**: g is monotonically non-decreasing over $[0,1]$.

Now, we can conclude the lemma, since

$$\mathop{\mathbb{E}}_{Y_v \sim [0,1]}[y_u \mid Y^{-v}] = \mathop{\mathbb{E}}_{Y_v \sim [0,1]}[1 - g(Y_{M(u)}) \mid Y^{-v}] \geq 1 - g(Y_{h(u)}).$$

□

We are now ready to prove the main theorem. So far, we have proven

$$\mathop{\mathbb{E}}_{Y_v \sim [0,1]}[y_u + y_v \mid Y^{-v}] \geq 1 - g(Y_{h(u)}) + \int_0^{Y_{h(u)}} g(x)dx.$$

Which means that for every edge $\{u, v\} \in E$, we have

$$\begin{aligned}
\mathbb{E}[y_u + y_v] &= \mathop{\mathbb{E}}_{Y^{-v}}\left[\mathop{\mathbb{E}}_{Y_v \sim [0,1]}[y_u + y_v \mid Y^{-v}]\right] \\
&\geq \mathop{\mathbb{E}}_{Y^{-v}}\left[1 - g(Y_{h(u)}) + \int_0^{Y_{h(u)}} g(x)dx\right] \\
&\geq \min_{\theta \in [0,1]}\left(1 - g(\theta) + \int_0^{\theta} g(x)dx\right).
\end{aligned}$$

To find the minimum, we can simply take the derivative and solve for 0. But before we take the derivative, we must make sure we have the following property for g:

- **Property 3**: g is convex and differentiable over $[0, 1]$.

$$\frac{\partial}{\partial\theta}\left(1 - g(\theta) + \int_0^{\theta} g(x)dx\right) = -g'(\theta) + g(\theta) = 0.$$

Then, we have that $1 - g(\theta) + \int_0^{\theta} g(x)dx$ is minimized when $g(\theta) = g'(\theta)$.

Before we finalize the proof, we now have everything we need to find what the exact potential function g is. Since we have $g(\theta) = g'(\theta)$, by combining Properties 2 and 3, we have $g(x)$ in the form of $g(x) = e^{x+c}$. Setting $c = -1$ satisfies the first property, so we have

$$g(x) = e^{x-1}.$$

Now, we are ready to conclude the proof. For every edge $\{u, v\} \in E$, we have

$$\mathbb{E}[y_u + y_v] \geq \min_{\theta \in [0,1]}\left(1 - g(\theta) + \int_0^{\theta} g(\theta)dx\right) = \min_{\theta \in [0,1]}\left(1 - e^{\theta-1} + e^{\theta-1} - e^{-1}\right) = 1 - \frac{1}{e},$$

which, in combination with Lemma A.2, proves the main theorem. □