Online Bipartite Matching with Random Arrivals: An Approach Based on Strongly Factor-Revealing LPs

Mohammad Mahdian
Yahoo! Research
4301 Great America Pkwy
Santa Clara, CA, USA.
mahdian@alum.mit.edu

Qiqi Yan*
Stanford University
460 Gates Building, 353 Serra Mall
Stanford, CA, USA
qiqiyan@cs.stanford.edu

ABSTRACT

In a seminal paper, Karp, Vazirani, and Vazirani [9] show that a simple ranking algorithm achieves a competitive ratio of $1 - 1/e$ for the online bipartite matching problem in the standard adversarial model, where the ratio of $1-1/e$ is also shown to be optimal. Their result also implies that in the random arrivals model defined by Goel and Mehta [6], where the online nodes arrive in a random order, a simple greedy algorithm achieves a competitive ratio of $1-1/e$. In this paper, we study the ranking algorithm in the random arrivals model, and show that it has a competitive ratio of at least 0.696, beating the $1 - 1/e \approx 0.632$ barrier in the adversarial model. Our result also extends to the i.i.d. distribution model of Feldman et al. [5], removing the assumption that the distribution is known.

Our analysis has two main steps. First, we exploit certain dominance and monotonicity properties of the ranking algorithm to derive a family of factor-revealing linear programs (LPs). In particular, by symmetry of the ranking algorithm in the random arrivals model, we have the monotonicity property on both sides of the bipartite graph, giving good "strength" to the LPs. Second, to obtain a good lower bound on the optimal values of all these LPs and hence on the competitive ratio of the algorithm, we introduce the technique of strongly factor-revealing LPs. In particular, we derive a family of modified LPs with similar strength such that the optimal value of *any single one* of these new LPs is a lower bound on the competitive ratio of the algorithm. This enables us to leverage the power of computer LP solvers to solve for large instances of the new LPs to establish bounds that would otherwise be difficult to attain by human analysis.

*Supported by a Stanford Graduate Fellowship. Part of this work was done while the author was at Yahoo! Research, Santa Clara.

STOC'11, June 6–8, 2011, San Jose, California, USA.

Categories and Subject Descriptors

G.2.2 [**Discrete Mathematics**]: Graph algorithms

General Terms

Algorithms, Theory

Keywords

online matching, random arrival, factor-revealing LP

1. INTRODUCTION

Bipartite matching is among the most fundamental problems in combinatorial optimization. This is the problem that a match-maker faces when matching a group of boys with a group of girls subject to compatibility constraints given by a bipartite graph, and the goal is to maximize the number of matched pairs. The online version of this problem also has many applications, from matching patients and donated organs to allocating online advertisement space [9, 15, 4, 11]. The main focus of this paper will be on online bipartite matching.

Karp, Vazirani and Vazirani [9] were the first to formulate the online bipartite matching problem. In their model, the nodes on one side of the bipartite graph (say boys) arrive online with the incident edges, and must be matched upon their arrivals (to the girls), if possible. Karp, Vazirani, and Vazirani [9] showed that the following simple randomized algorithm achieves a competitive ratio of $1-1/e$: rank all girls randomly according to a random permutation, and whenever a boy arrives, match him to the highest-ranked available girl who is connected to the boy (i.e., is compatible with him). The competitive ratio of $1-1/e$ means that even if an adversary determines the order of the boys' arrivals, the expected number of matched pairs under this algorithm is at least $1 - 1/e \approx 0.632$ times the number of pairs that an offline algorithm with the full knowledge of the input would match.

Worst-case analysis in the adversarial model often leads to over-pessimistic results. Goel and Mehta [6] proposed to study the random arrivals model, where the boys arrive in a random order rather than an adversarial order. This assumption smooths out certain bizarre worst-case examples, while still being much weaker than an i.i.d. distribution assumption that is common in average-case analysis. It turns out that the result of [9] also shows that in the random arrivals model, the greedy algorithm that simply matches each boy to the first available compatible girl (if one exists) according to a fixed tie-breaking rule achieves a competitive ratio of $1 - 1/e$. A natural question that was left open is whether the any algorithm can achieve a competitive ratio better than $1 - 1/e$ in the random arrivals model. Our paper answers this question affirmatively, and the algorithm is simply the ranking algorithm.

We prove that the competitive ratio of the ranking algorithm in the random arrivals model is at least 0.696. In addition to answering a natural theoretical question, our result strengthens the result of Feldman et al. [5] and the follow-up work by Manshadi et al. [14]. Feldman et al. [5] show that when each boy is sampled i.i.d. from a *known* distribution, there is an online algorithm that achieves a ratio of 0.67. This ratio was recently improved to 0.702 by Manshadi et al. [14]. Compared to Manshadi et al. [14], we get a slightly worse factor (0.696), but without the assumption that the algorithm has prior knowledge of the distribution. Furthermore, the ranking algorithm is significantly simpler than the algorithms in [5, 14].

Our Technique.

Our proof is based on a twist on the idea of using *factor-revealing linear programs (LPs)*. We start by forming a family of exponential size factor-revealing LPs based on a dominance property and a monotonicity property of the ranking algorithm, such that the infimum of the optimal values of the LPs in this family would be a lower bound on the competitive ratio of the algorithm. These two properties have been used previously to prove the competitive ratio of $1 - 1/e$ for the ranking algorithm in the adversarial setting, or for the greedy algorithm in the random arrivals model. Our LPs utilize monotonicity property from both sides of the bipartite graph, which gives good strength to the LPs to beat $1 - 1/e$.

Next, we relax these exponential size LPs into a family of polynomial size LPs. To analyze the infimum of the solutions of these polynomial size LPs, we form a related family of LPs, which we call *strongly factor-revealing LPs*, and prove that the optimal value of *any single one* of these LPs is a lower bound on the competitive ratio. This enables us to use a computer LP solver to solve large instances of the strongly factor-revealing LPs numerically, and obtain good lower bounds for the competitive ratio. Note that whereas for standard factor-revealing LPs, computer LP solvers merely provide guidance for the human-generated proof [7], in our case the solution produced by the LP solvers is actually part of the proof. Despite its simplicity, this is a powerful trick for analyzing factor-revealing LPs and has applications beyond the present problem. For example, in [12] we have used this technique to significantly simplify and in some cases improve the analysis of the factor-revealing LPs used to bound the approximation factor of facility location algorithms in [7, 13].

Related Work.

Mehta et al. [15] formulated a generalization of the online bipartite matching problem motivated by allocating online advertisement space, sparking renewed attention to this problem in the theory community. Goel and Mehta [6] used factor-revealing LPs to give a simpler proof for the competitive ratio of greedy for this generalization in the random arrivals model, and corrected a mistake in the original proof of Karp, Vazirani, and Vazirani [9] (originally found by Krohn and Varadarajan [10]). Birnbaum and Mathieu [3] gave a significantly simpler proof for the online matching problem using a direct charging argument, and Aggarwal et al. [1] took this result one step further by proving a $1 - 1/e$ bound for the vertex-weighted version of online bipartite matching.

Feldman et al. [5] were the first to give an algorithm with a competitive ratio better than $1 - 1/e$ in a model where nodes on one side of the graph arrive i.i.d. according to a known distribution with integral arrival rates. Their algorithm crucially uses the prior knowledge of this distribution in order to pre-compute matchings that guide the online assignment. In follow-ups to this work, Manshadi et al. [14] gave an algorithm with an improved competitive ratio of 0.702 for arbitrary arrival rates, and Bahmani and Kapralov [2] also proved several upper bounds in this model.

Simultaneously and independently of this work, Karande, Mehta, and Tripathi [8] proved that the ranking algorithm has an approximation ratio better than $1 - 1/e$ in the random arrivals model. They prove a bound of 0.653 for general graphs, and a bound of $1 - o(1)$ for graphs that have $\omega(1)$ disjoint perfect matchings. They also give a family of examples which shows that the ranking algorithm has an approximation factor no better than 0.727 in the random arrivals model. This shows that our 0.696 bound is at most 0.031 away from the best possible.

2. PRELIMINARIES

Setting.

We have an undirected bipartite graph $G = (L, R, E)$ with n girls $1, \ldots, n$ as left hand side nodes and n boys $1, \ldots, n$ as right hand side nodes. An edge of the graph indicates whether a girl and a boy can be matched to each other. The graph is not revealed to us in advance. We assume that the girls are fixed, and the boys arrive online. Upon the arrival of a boy, the edges incident to the boy are revealed, and the algorithm must match the boy to an unmatched girl immediately, if possible. Note that the assumption that the number of nodes on the two sides are equal is without loss of generality, since the smaller side can always be padded with isolated nodes. We denote the order of arrival of the boys by a permutation π of $[n] := \{1, \ldots, n\}$, i.e., $\pi(r)$ is the r-th boy that arrives. We also say that boy $\pi(r)$ has rank r.

Following the possibly confusing convention of the literature, we assume that a smaller rank number means a higher rank. E.g., 1 is the highest rank, and n is the lowest rank.

Competitive Analysis: Adversarial vs Random.

We evaluate an online algorithm by comparing the (expected) number of pairs it matches against the maximum matching. This is often done in an *adversarial model*, i.e., assuming that the input graph G and the order of arrivals of the boys are picked by an adversary, and the worst-case ratio between the expected size of the algorithm's solution and the size of the optimal matching is called the *competitive ratio* of the algorithm. More formally, let $MM(G)$ denote the size of the maximum matching in G. An algorithm has competitive ratio α ($0 \le \alpha \le 1$) if the expected size of the matching it produces is at least $\alpha \cdot MM(G)$ for every bipartite graph G and every ordering π of boys' arrivals. A less strict model is the *random arrivals model.* In this model the ordering π of the boys' arrivals is assumed to be a random permutation of $[n]$. An algorithm has competitive ratio α in the random arrivals model if the expected size of the matching it produces is at least $\alpha \cdot MM(G)$ for every bipartite graph G, where the expectation is over the random coin flips of the algorithm as well as the random choice of the arrival order π.

The Greedy Algorithm.

Given a fixed tie-breaking rule among the girls, the greedy algorithm matches each arriving boy to the first available girl according to the tie-breaking rule, if such a girl exists. Greedy produces a maximal matching, and hence has a competitive ratio of at least $\frac{1}{2}$ in the adversarial model.

The Ranking Algorithm.

The ranking algorithm of [9] first randomly gives a ranking on the girls, and then runs the greedy algorithm with ties broken in favor of the highly ranked girls. To be specific, at the beginning of the algorithm, it randomly picks a permutation σ which assigns ranks from 1 to n to the girls, where girl $\sigma(l)$ is given rank l for $l = 1, \ldots, n$. Then in running the greedy algorithm, tie-breaking is in favor of the girls with higher ranks (recall that this means smaller rank numbers).

We use Ranking(G, σ, π) to denote the matching produced by the ranking algorithm on graph G when the ranking over the girls is σ and the order of arrivals of the boys is π. The rest of this paper is devoted to analyzing the ranking algorithm in the random arrivals model. That is, we seek to bound E[Size of Ranking(G, σ, π)] where the expectation is over the random choice of permutations σ and π, and we prove that it is at least $0.696 \cdot MM(G)$ for every bipartite graph G.

3. DERIVING A FAMILY OF FACTOR REVEALING LPS

In this section, we first derive a family of exponential size factor-revealing LPs, and then relax them into polynomial size LPs. We start with the following duality principle, which is also used in the original paper of Karp, Vazirani, and Vazirani [9].

LEMMA 3.1 (DUALITY PRINCIPLE). *Let $G = (L, R, E)$ be an arbitrary bipartite graph and G' be the mirror of G, i.e., $G' = (R, L, E')$ with $E' = \{(a, b) : (b, a) \in E\}$. Then for any two permutations σ and π, Ranking(G, σ, π) is the inverse of Ranking(G', π, σ).*

We leave the proof of the above lemma to the full version of this paper. Intuitively, this lemma states that boys and girls play symmetrical roles in the ranking algorithm with random arrivals. This allows us to apply any inequality that we prove for the boys to the girls' side and vice versa.

3.1 A Family of Exponential Size LPs

Throughout this section, we fix a bipartite graph G and an optimal matching OPT in G. Let $x(\sigma, \pi, l, r)$ be the binary indicator variable for the event that Ranking(G, σ, π) matches the girl at rank l to the boy at rank r. We prove two classes of inequalities, based on *dominance* and *monotonicity* properties of the ranking algorithm. These inequalities will be the constraints of our factor-revealing LPs.

3.1.1 Dominance Property

Fix two permutations σ and π. Assume the optimal matching OPT matches girl $\sigma(l)$ with boy $\pi(r)$. Then when the boy with rank r arrives, the ranking algorithm either matches him to some girl with rank at most l, or if it fails to do so, it must be the case that the girl at rank l was already matched to some boys of smaller rank by the algorithm. In other words,

LEMMA 3.2. *For any two permutations σ, π and for all $l, r \in [n]$, we have:*

$$\sum_{l'=1}^{l} x(\sigma, \pi, l', r) + \sum_{r'=1}^{r-1} x(\sigma, \pi, l, r') \ge \mathbf{1}_{(\sigma(l), \pi(r)) \in OPT}, \quad (1)$$

where $\mathbf{1}_{(\sigma(l), \pi(r)) \in OPT}$ is the binary indicator variable for the event $(\sigma(l), \pi(r)) \in OPT$.

These constraints are in fact enough to prove a competitive ratio of $1/2$ for the ranking algorithm.

3.1.2 Monotonicity Property

Given permutations σ, π, consider promoting a girl with rank l_{old} to a higher rank of l_{new} (i.e., $l_{new} < l_{old}$), with relative ranking of the other girls intact. In other words, we define a new ranking for girls $\sigma_{l_{old}}^{l_{new}}$ as follows: $\sigma_{l_{old}}^{l_{new}}(l) = \sigma(l)$ if $l < l_{new}$ or $l > l_{old}$, $\sigma_{l_{old}}^{l_{new}}(l+1) = \sigma(l)$ if $l_{new} \le l < l_{old}$, and $\sigma_{l_{old}}^{l_{new}}(l_{new}) = \sigma(l_{old})$.

With an argument similar to the ones in [9, 6, 3], it can be observed that the symmetric difference between the two matchings Ranking(G, σ, π) and Ranking$(G, \sigma_{l_{old}}^{l_{new}}, \pi)$ is a single alternating path (see Lemma 2 in [3]). If we restrict our attention to girls of rank at most l_{old} in σ, the difference is an alternating path starting from the node $\sigma(l_{old})$, as depicted in Figure 1. This means that for every boy b, if b was matched to some girl with rank at most l in Ranking(G, σ, π), i.e., $\sum_{l'=1}^{l} x(\sigma, \pi, l', r) = 1$, then in Ranking$(G, \sigma_{l_{old}}^{l_{new}}, \pi)$ he is matched to a girl with rank at most $l + 1_{l \ge l_{new}}$, i.e., $\sum_{l'=1}^{l+\mathbf{1}_{l \ge l_{new}}} x(\sigma_{l_{old}}^{l_{new}}, \pi, l', r) = 1$. (the $+1_{l \ge l_{new}}$ term is due to the insertion of a girl at rank l_{new}) Therefore the monotonicity property can be captured with the following inequality.

LEMMA 3.3. *For all σ, π, $l_{new}, l_{old}, l, r \in [n]$ such that $l_{new} < l_{old}$ and $l < l_{old}$, we have:*

$$\sum_{l'=1}^{l+\mathbf{1}_{l \ge l_{new}}} x(\sigma_{l_{old}}^{l_{new}}, \pi, l', r) \ge \sum_{l'=1}^{l} x(\sigma, \pi, l', r). \quad (2)$$

This set of inequalities, together with the inequalities for the dominance property, can be used to prove a competitive

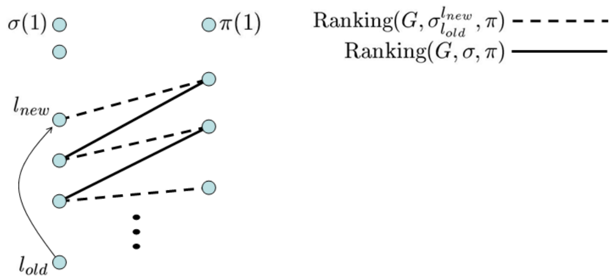

Figure 1: Difference between Ranking(G, σ, π) and Ranking$(G, \sigma^{l_{new}}_{l_{old}}, \pi)$ on girls of rank at most l_{old}

ratio of $1 - 1/e$ for the ranking algorithm in the adversarial setting, as they capture the logic behind the previous proofs of this theorem. In our random arrivals setting, since boys and girls play symmetric roles, the duality principle implies that the following symmetric inequality is also true, which gives us additional strength to beat $1 - 1/e$.

LEMMA 3.4. *For all σ, π, $r_new, r_old, r, l \in [n]$ such that $r_{new} < r_{old}$ and $r < r_{old}$, we have:*

$$\sum_{r'=1}^{r+\mathbf{1}_{r \geq r_{new}}} x(\sigma, \pi^{r_{new}}_{r_{old}}, l, r') \geq \sum_{r'=1}^{r} x(\sigma, \pi, l, r'). \qquad (3)$$

3.1.3 Putting Everything Together

Given $n > 0$ and the number of edges $k > 0$ in the optimal matching OPT, we define the following LP, which we call expLP(n, k), with an exponential number of variables and constraints. The variables of this LP are the $x(\sigma, \pi, l, r)$ variables defined above, except that they are no longer required to be either 0 or 1. We may assume without loss of generality that the optimal matching OPT consists of edges (i, i) for $i \in [k]$. With this assumption, $\mathbf{1}_{(\sigma(l),\pi(r)) \in OPT} = \mathbf{1}_{\sigma(l)=\pi(r) \leq k}$. We can now define the factor-revealing LP expLP(n, k):

$$\begin{array}{rc} \text{minimize} & \frac{1}{(n!)^2 k} \sum_{\sigma,\pi,l,r} x(\sigma, \pi, l, r) \\ \text{subject to:} & (1), (2), (3), \\ & \forall \sigma, \pi, l, r : x(\sigma, \pi, l, r) \geq 0 \end{array}$$

With a slight abuse of notation, we also let expLP(n, k) denote the optimal value of the above LP. The following lemma summarizes what we have so far.

LEMMA 3.5. *The competitive ratio of the ranking algorithm in the random arrivals model is lower-bounded by $\inf_{n \in I} \inf_k$ expLP(n, k), for any infinite subset I of natural numbers.*

PROOF. Suppose we run the ranking algorithm on a graph of size n in the random arrivals model. Let n' be any integer in I that is as large as n. Then by adding isolated vertices to the graph, we ensure the size of the graph is n', without changing the solution of the ranking algorithm, or the size of the optimal matching.

Let $x(\sigma, \pi, l, r)$ be 1 if the girl at rank l is matched to the boy at rank r when the permutations are σ, π respectively, and 0 otherwise. Then as we showed in Lemma 3.2, Lemma 3.3, and Lemma 3.4, x is a feasible solution to expLP(n'), and the objective value of expLP(n') under x equals to the competitive ratio of the algorithm for the graph, i.e., the ratio of the expected size of the solution of the algorithm to the size of the optimal matching. As a linear relaxation, optimal value of expLP(n') is a lower bound on the competitive ratio of the algorithm when graphs have size n. Our lemma follows by taking infimum over all n. □

3.2 A Family of Polynomial Size LPs

The linear program expLP(n, k) has an exponential size, making it hard to solve either analytically or numerically. In this section, we carefully define polynomially many new variables in terms of the variables of expLP(n, k), and write the constraints that the inequalities of expLP(n, k) impose on these new variables. This results in a polynomial size relaxation of expLP(n, k) which is easier to analyze while preserving most of its strength.

Recall that w.l.o.g. we let OPT consist of edges (i, i) for $i \in [k]$. Let $x_1(l, r, p)$ be the probability (over the random choices of σ, π) that the girl at rank l is matched to the boy at rank r in Ranking(G, σ, π) and the girl at rank p is matched to the boy at rank r in OPT, i.e., $\sigma(p) = \pi(r) \leq k$. In other words,

$$x_1(l, r, p) = E_{\sigma,\pi}[\mathbf{1}_{\sigma(p)=\pi(r) \leq k} \cdot x(\sigma, \pi, l, r)].$$

Similarly, let $x_2(r, l, q)$ be the probability that the boy at rank r is matched to the girl at rank l in Ranking(G, σ, π) and the boy at rank q is matched to the girl at rank l in OPT. In other words,

$$x_2(r, l, q) = E_{\sigma,\pi}[\mathbf{1}_{\sigma(l)=\pi(q) \leq k} \cdot x(\sigma, \pi, l, r)].$$

Note we have chosen the order of indices for x_2 in such a way that $x_1(l, r, p)$ and $x_2(r, l, q)$ have symmetric forms.

We also introduce redundant partial sum variables

$$y_1(l, r, p) = \sum_{l'=1}^{l} x_1(l', r, p) \quad \forall l, r, p \in [n]$$

$$y_2(r, l, q) = \sum_{r'=1}^{r} x_2(r', l, q) \quad \forall r, l, q \in [n],$$

and let $y_1(0, r, p) = 0$ for $r, p \in [n]$ and $y_2(0, l, q) = 0$ for $l, q \in [n]$ for convenience.

We now derive inequalities between these variables based on the inequalities of expLP(n, k). Multiplying both sides of the dominance inequality (1) by $\mathbf{1}_{\sigma(l)=\pi(r) \leq k}$ gives us:

$$\sum_{l'=1}^{l} \mathbf{1}_{\sigma(l)=\pi(r) \leq k} \cdot x(\sigma, \pi, l', r) + \sum_{r'=1}^{r-1} \mathbf{1}_{\sigma(l)=\pi(r) \leq k} \cdot x(\sigma, \pi, l, r') \geq \mathbf{1}_{\sigma(l)=\pi(r) \leq k}.$$

Taking expectation of both sides of this inequality over a random choice of σ and π, and we have:

$$y_1(l, r, l) + y_2(r - 1, l, r) \geq k/n^2 \qquad \forall l, r \in [n] \qquad (4)$$

Similarly, we multiply both sides of the monotonicity inequality (2) by $\mathbf{1}_{\sigma(p)=\pi(r)\leq k}$ for a fixed p, let $l_{old} = n$, and take expectation over a random choice of σ, π. The right-hand side of the resulting inequality is clearly $y_1(l, r, p)$. For the left-hand side, we observe that the permutation $\sigma_{l_{old}}^{l_{new}}$ is a uniformly random permutation, and that $\sigma(p) = \pi(r)$ if and only if $\sigma_n^{l_{new}}(p_{new}) = \pi(r)$, where:

$$p_{new} = \begin{cases} p & p < l_{new} \\ p+1 & l_{new} \leq p < n \\ l_{new} & p = n. \end{cases}$$

Using these observations the inequality (2) implies that for every $l_{new}, l < n$ and r, p,

$$y_1(l + \mathbf{1}_{l \geq l_{new}}, r, p_{new}) \geq y_1(l, r, p). \tag{5}$$

A symmetric argument using (3) implies that for every $r_{new}, r < n$ and l, q,

$$y_2(r + \mathbf{1}_{r \geq r_{new}}, l, q_{new}) \geq y_2(r, l, q), \tag{6}$$

where q_{new} is defined by an equation similar to (3.2) with p and l_{new} replaced by q and r_{new}. Finally, for every l, r we have

$$\sum_p x_1(l, r, p) = \sum_q x_2(r, l, q), \tag{7}$$

as both sides of the equation are equal to $\frac{k}{n}\mathrm{E}_{\sigma,\pi}[x(\sigma, \pi, l, r)]$. Inequalities (4), (5), (6), and (7), together with non-negativity constraints and equations defining y_1 and y_2 in terms of x_1 and x_2 form the constraints of our new LP. The objective of this LP is $\frac{1}{2k}\left(\sum_{l,r,p} x_1(l, r, p) + \sum_{r,l,q} x_2(r, l, q)\right)$, which is equal to the objective of expLP(n, k).

3.2.1 Simplifications

Before analyzing the above LP, we further simplify it in a few steps:

- If x_1, x_2 (together with the corresponding y_1, y_2) is a feasible solution, then $\bar{x}_1 = \bar{x}_2 = (x_1 + x_2)/2$ (and $\bar{y}_1 = \bar{y}_2 = (y_1 + y_2)/2$) is also feasible with the same objective value. Hence by imposing that $x_1 = x_2$, the optimal LP value is the same.
- The value of k does not affect the value of the LP, since by scaling a solution of the LP for k by a factor of k'/k we obtain a solution for k'. So w.l.o.g., we let $k = n$.
- For $p < l_{new}$, we have $p_{new} = p$; hence inequality (5) is trivial for this range of p.
- For $l_{new} \leq p < n$, we have $p_{new} = p + 1$, and inequality (5) becomes $y_1(l + \mathbf{1}_{l \geq l_{new}}, r, p + 1) \geq y_1(l, r, p)$. This constraint is strongest when $l_{new} = p$. So if $l \geq p$, we have $y_1(l + 1, r, p + 1) \geq y_1(l, r, p)$, and if $l < p$, we have $y_1(l, r, p + 1) \geq y_1(l, r, p)$.
- For $p = n$, we have $p_{new} = l_{new}$, and inequality (5) becomes $y_1(l + \mathbf{1}_{l \geq l_{new}}, r, l_{new}) \geq y_1(l, r, n)$. If $l_{new} > l$, this is $y_1(l, r, l_{new}) \geq y_1(l, r, n)$. This, together with the inequality $y_1(l, r, p + 1) \geq y_1(l, r, p)$ from the last paragraph shows that $y_1(l, r, p)$ are equal for $p = l + 1, \ldots, n$. If $l_{new} \leq l$, we get $y_1(l+1, r, l_{new}) \geq y_1(l, r, n)$.

The above observations together with Lemma 3.5 can be summarized in the following Lemma.

LEMMA 3.6. *The competitive ratio of the ranking algorithm in the random arrivals model is lower-bounded by* $\inf_{n \in I}$ *polyLP*(n) *for any infinite subset* I *of natural numbers, where polyLP*(n) *is the linear program defined below:*

$$\textit{minimize } \frac{1}{n} \sum_{l,r,p \in [n]} x(l, r, p) \textit{ subject to}$$

$$\begin{aligned}
\forall l, r \in [n]: &\quad y(l, r, l) + y(r - 1, l, r) \geq \frac{1}{n} \\
\forall l, r, p \in [n], p \leq l < n: &\quad y(l + 1, r, p + 1) \geq y(l, r, p) \\
\forall l, r, p \in [n], l < p: &\quad y(l, r, p) = y(l, r, l + 1) \\
\forall l, r, p \in [n], p \leq l < n: &\quad y(l + 1, r, p) \geq y(l, r, l + 1) \\
\forall l, r \in [n]: &\quad \sum_p x(l, r, p) = \sum_p x(r, l, p) \\
\forall r, p \in [n], 0 \leq l \leq n: &\quad y(l, r, p) = \sum_{l'=1}^{l} x(l', r, p) \\
\forall l, r, p \in [n]: &\quad x(l, r, p) \geq 0
\end{aligned}$$

Unfortunately, despite its apparently simple form, giving a closed form optimal solution to polyLP(n) seems difficult. In the next section, we propose the technique of strongly factor revealing LPs, which will enable us to prove good lower bounds on $\inf_{n \in I}$ polyLP(n) for some infinite set I using computer LP solvers, which is sufficient for our purpose.

4. STRONGLY FACTOR-REVEALING LPS

For a maximization algorithm, a family of LPs $LP(n)$ is called factor-revealing, if the infimum of the optimal values of LPs in this family is a lower bound on the approximation ratio (or competitive ratio) of the algorithm. For technical convenience, we assume that the approximation ratio of the algorithm is monotonely decreasing, like in the case of ranking algorithm.

If LPs in such a family have a nice form, we can solve each of them analytically and get a tight bound on the ratio of the algorithm (as was done in [15, 11]). When this is not the case, a common approach is to observe patterns in the optimal dual solutions, and then try to extrapolate a class of the dual solutions that are near-optimal. However, this process can be painstaking, and is usually doomed to incur a constant loss in factor (see [7, 13] for an example).

Here we propose a different approach. We say that a family of LPs is *strongly factor-revealing*, if the solution of *any* LP in this family is a lower bound on the approximation ratio of the algorithm. Our approach is based on transforming our family of factor-revealing LPs into a family of strongly factor-revealing LPs of almost the same strength. Once this is done, we can use LP solvers to solve large LPs in this family and obtain good lower bounds on the approximation ratio of the algorithm. Despite its apparent simplicity, this is a powerful technique. For example, we have used this technique [12] to significantly simplify and in some cases improve the analysis of factor-revealing LPs in [7, 13]. These LPs are the basis of the analysis of the currently best known algorithms for the uncapacitated facility location problem, and bounding their solution is the most technically challenging part of the analysis.

To illustrate how one can get strongly factor-revealing LPs from standard factor-revealing ones, we consider the following illustrative example.

4.1 An Illustrative Example

Suppose we have the following family of factor-revealing LPs for some algorithm, denoted by $LP(n)$.

$$\begin{array}{ll} \text{minimize } \frac{1}{n}\sum_{t=1}^{n} x_t & \text{subject to:} \\ 1 - x_t \leq \frac{1}{n}\sum_{s=1}^{t-1} x_s & \forall t = 1, \ldots, n \\ x_t \geq x_{t+1} \geq 0 & \forall t = 1, \ldots, n-1 \end{array}$$

It is in fact easy to solve these LPs analytically: the optimal solution is $x_t = (1 - \frac{1}{n})^{t-1}$ for $t \in [n]$, giving an objective value of $1 - (1 - \frac{1}{n})^n$. Note that the optimal solution, normalized properly, is converging to the continuous function e^{-y} for $y \in [0, 1]$, while the optimal objective value is monotonely decreasing in n, and converges to $1 - 1/e$ from above. However, here we try to bound these LPs by derive a family of strongly factor-revealing LPs in order to illustrate the technique.

Fix a natural number m, and consider any multiple n of m. Let $n = md$. Then we start with the optimal solution $x_1, \ldots, x_n$ to $LP(n)$, and project it into the smaller size m while preserving objective value. Specifically, we define

$$x'_i = \frac{1}{d} \cdot \sum_{j=(i-1)d+1}^{id} x_j \text{ for all } i \in [m].$$

See Figure 2 for a pictorial demonstration. Clearly this transformation preserves the objective value, i.e.,

$$\frac{1}{n}\sum_{t=1}^{n} x_t = \frac{1}{m}\sum_{t=1}^{m} x'_t.$$

But x' is not a feasible solution to $LP(m)$. However, at least when n, m are both large, both solutions are similar in form, and x' only slightly violates the constraints of $LP(m)$.

Now comes our main twist. Instead of trying to fix the projected solution such that it becomes feasible, we modify the constraint to accommodate the projected solution. As the projected solution is only slightly infeasible, our changes to the constraints will be slight too. Specifically, we replace the first constraint of $LP(n)$ by

$$1 - x_t \leq \frac{1}{m}\sum_{s=1}^{t} x_s,$$

and let $LP'(n)$ denote the resulting LP.

To justify our modification, we verify that the variables x' defined above constitute a feasible solution to $LP'(m)$:

$$\begin{aligned} 1 - x'_t &= 1 - \frac{1}{d}\sum_{j=(t-1)d+1}^{td} x_j \leq 1 - x_{td} \\ &\leq \frac{1}{n}\sum_{s=1}^{td-1} x_s \leq \frac{1}{n}\sum_{s=1}^{td} x_s = \frac{1}{m}\sum_{s=1}^{t} x'_s. \end{aligned}$$

Here the first and last equalities are by the definition of x', the first inequality is by the second constraint of $LP(n)$, and the second inequality is by the first constraint of $LP(n)$.

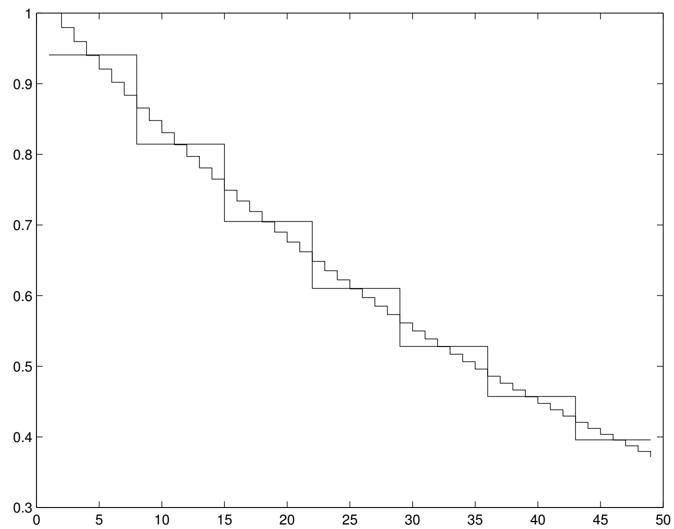

Figure 2: Projecting Solutions

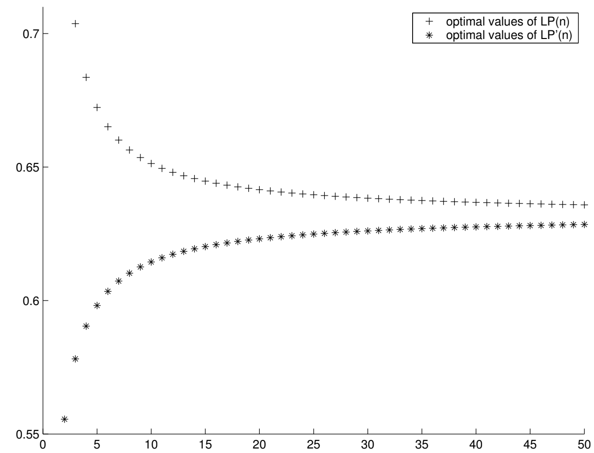

Figure 3: Optimal Values of $LP(n)$ and $LP'(n)$

As a consequence, $LP'(m)$ is a lower bound on $LP(n)$ for all n that is a multiple of m. In fact, even the assumption that n is a multiple of m is not necessary, since we can first pad an optimal solution of $LP(n)$ with $n' - n$ zeros where $n' = m\lceil n/m \rceil$ and then scale the solution up by a factor of n'/m. This gives us a feasible solution of $LP(n')$ with the same objective value, and the above transformation can be applied to this solution to get a feasible solution of $LP'(m)$.

Hence $LP'(m)$ is a family of strongly factor-revealing LPs. Thus, the value of $LP'(m)$ for any m gives a lower bound on $\inf_n LP(n)$. As it turns out, in this case, the optimal solution of $LP'(m)$ is precisely $1 - (1 + 1/m)^{-m}$, and therefore the lower bound given by $LP'(m)$ converges to the correct value of $\inf_n LP(n)$ from below, which is $1 - 1/e$. Figure 3 shows the values of $LP(n)$ and $LP'(n)$ for $n \leq 50$.

4.2 The Technique

The notion of strongly factor-revealing LPs is most useful when the standard factor-revealing LPs in question are hard to solve analytically, but their optimal solutions converge to some continuous curve or surface. When this is the case, the LP constraints often are robust to slight alternations (e.g. shift of indices).

To get strongly factor-revealing LPs from standard ones, we fix $LP(m)$, take the solution of $LP(n)$ for any large n that is a multiple of m, and project it down to construct a candidate solution to $LP(m)$ while preserving the objective value. This usually violates some of the constraints of $LP(m)$, but only slightly, similar to a rounding error. Then we define a slightly relaxed version $LP'(m)$ of $LP(m)$ in such a way that the projected solution becomes feasible. It follows that the optimal value of $LP'(m)$ is a lower bound for $LP(n)$ for all multiples n of m, and hence $LP'(m)$ is strongly factor-revealing. Often, as we set m to be a large number, such rounding error becomes less significant, and $LP'(n)$ and $LP(n)$ are similar in both forms and "strength".

This is a heuristic meta-recipe for constructing a strongly factor-revealing LP. There does not seem to be a general theorem for this approach, and the solution often requires specific understanding of the structure of the original factor-revealing LP. Sometimes, trying to repair the constraints by merely shifting the indices will hurt the strength of the LPs by too much. For such cases, what we can do instead is to write the strongest constraints possible that are feasible for the projected solution. For example, as we will see in Section 4.3, in the case of polyLP(n) we will need to replace one inequality with n relaxed inequalities. This can sometimes be difficult and requires nontrivial understanding of the structure of the factor-revealing LPs, but it is still considerably easier than solving the LPs analytically.

There are several advantages in using our strongly factor-revealing LP approach to analyze factor-revealing LPs. First of all, our approach often gives good bounds that are as close as computational resources allow, which is the second best thing to having an analytical solution. Secondly, (the manual part of) the resulting proof is significantly simpler than the technically messy arguments based on estimating the dual (as in [7, 13]). Finally, often one faces the trade-off between having factor-revealing LPs with good strength and having such LPs that are amenable to analysis. The simplicity of our approach allows us to focus on the former, and let computers take care of the latter.

4.3 Strongly Factor-Revealing LPs for Ranking

Our strongly factor-revealing LP polyLP$'(n)$ is defined to be the same as polyLP(n) except with the first constraint replaced by:

$$\forall\, l,r,p \in [n]: \quad y(l,r,l) + y(r,l,p) \geq 1/n. \tag{8}$$

We remark that the choice of this modification is not obvious. It is the strongest constraint that we are able to write based on our technical understanding of the structure of polyLP(n). All the other natural modifications we tried do not preserve the strength of the LPs as well as this one, as suggested by computer experiments.

Lemma 4.1. *The family polyLP$'(n)$ defined above is a family of strongly factor-revealing LPs for the ranking algorithm in the random arrivals model. In other words, for every m, polyLP$'(m)$ is a lower bound on the competitive ratio of the ranking algorithm in the random arrivals model.*

Proof. Fix integer $m > 0$. For any instance size n that is a multiple of m, we show that the competitive ratio of the ranking algorithm in the random arrivals model on instances of this size is lower-bounded by polyLP$'(m)$.

By Lemma 3.6, the desired competitive ratio is at least $\inf_d$ polyLP(md). As in the example in Section 4.1, we take an optimal solution x, y of polyLP(n) for $n = md$ and project it down to a feasible solution x', y' of polyLP$'(m)$. This projection is defined as follows:

Definition 4.2. *Let x, y be an optimal solution to the linear program polyLP(n) with $n = md$. For any $i, j, k \in [m]$, we define $f(i) = \{d \cdot (i-1)+1, \ldots, d \cdot i\}$, $f(i,j) = f(i) \times f(j)$, and $f(i,j,k) = f(i) \times f(j) \times f(k)$, where $\times$ denotes Cartesian product. For $l', r', p' \in [m]$, we define $x'(l',r',p') = \frac{1}{d}\sum_{(l,r,p)\in f(l',r',p')} x(l,r,p)$ and $y'(l',r',p') = \frac{1}{d}\sum_{(r,p)\in f(r',p')} y(l'\cdot d, r, p)$.*

It is easy to see that x', y' for polyLP$'(m)$ has the same objective value as x, y for polyLP(n).

$$\begin{aligned} & \frac{1}{m}\sum_{l',r',p'\in[m]} x'(l',r',p') \\ = \; & \frac{1}{m}\sum_{l',r',p'\in[m]} \frac{1}{d} \sum_{(l,r,p)\in f(l',r',p')} x(l,r,p) \\ = \; & \frac{1}{n}\sum_{l,r,p\in[n]} x(l,r,p) = \text{polyLP}(n). \end{aligned}$$

We next show that x', y' is a feasible solution to polyLP$'(m)$.

By summing the fifth constraint of polyLP(n) over $(l,r) \in f(l',r')$ we get the fifth constraint of polyLP$'(m)$. Similarly, the sixth constraint of polyLP$'(m)$ can be obtained by summing the corresponding constraint in polyLP(n) over $(r,p) \in f(r',p')$, $l = l' \cdot d$. For $p' \leq l' < m$, we have:

$$\begin{aligned} y'(l'+1, r', p'+1) &= \frac{1}{d}\sum_{(r,p)\in f(r',p')} y(l'd+d, r, p+d) \\ &\geq \frac{1}{d}\sum_{(r,p)\in f(r',p')} y(l'\cdot d, r, p) \\ &= y'(l', r', p'), \end{aligned}$$

where the inequality follows from chaining inequalities $y(l'd + j+1, r, p+j+1) \geq y(l'd+j, r, p+j)$ for $j = 0, \ldots, d-1$. This establishes the second inequality of polyLP$'(m)$. Other monotonicity constraints (the 3rd and 4th inequalities) are also straightforward to verify, and their proofs are left to Appendix A. To verify the new dominance inequality (8), note that for all $l', r', p' \in [m]$

$$\begin{aligned} y'(r', l', p') &= \frac{1}{d}\sum_{(l,p)\in f(l',p')} y(r'd, l, p) \\ &\geq \frac{1}{d}\sum_{l\in f(l')} d \cdot y(r'd-1, l, r'd) \\ &\geq \frac{1}{d}\sum_{(l,r)\in f(l',r')} y(r-1, l, r'd) \\ &= \frac{1}{d}\sum_{(l,r)\in f(l',r')} y(r-1, l, r). \end{aligned}$$

Here the second line follows from the fact that for $p \geq r'd$, $y(r'd, l, p) \geq y(r'd-1, l, p) = y(r'd-1, l, r'd)$, and for $p <$

$r'd$, $y(r'd, l, p) \geq y(r'd-1, l, r'd)$ by the fourth constraint of polyLP(n). The third line follows from the definition of y and non-negativity of x, and the fourth line follows from the third constraint of polyLP(n). Therefore,

$$\begin{aligned} & y'(l', r', l') + y'(r', l', p') \\ \geq\ & \frac{1}{d} \sum_{(r,l)\in f(r',l')} y(l'd, r, l) + \frac{1}{d} \sum_{(l,r)\in f(l',r')} y(r-1, l, r) \\ \geq\ & \frac{1}{d} \sum_{(r,l)\in f(r',l')} y(l, r, l) + \frac{1}{d} \sum_{(l,r)\in f(l',r')} y(r-1, l, r) \\ \geq\ & \frac{1}{d} \cdot d^2 \cdot \frac{1}{n} = \frac{1}{m}, \end{aligned}$$

where the third line follows from the definition of y and non-negativity of x, and the fourth line follows from the first constraint in polyLP(n). □

4.3.1 Results from LP Solvers

Given the results in the previous section, to obtain lower bounds on the competitive ratio of the ranking algorithm in the random arrivals model, it is sufficient to solve polyLP$'(m)$ for as many values of m as possible, and take the best of the results. The numerical results are reported in Appendix B. These results, together with Lemma 4.1, prove the main result of this paper:

THEOREM 4.3. *The competitive ratio of the ranking algorithm in the random arrivals model is at least* 0.696.

We do not know if polyLP gives a tight bound on the competitive ratio of ranking in the random arrivals model. The bipartite graph with the edge set $\{(i,j) : i \leq j\}$ for $n = 7$ shows that this factor is at most 0.796. Note that this is dominated by the family of examples given by Karande, Mehta, and Tripathi [8], which prove an upper bound of 0.727. Also, Manshadi et al. [14] showed that no online algorithm can have a factor better than 0.823, even in the i.i.d. model.

5. REFERENCES

[1] Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.

[2] Bahman Bahmani and Michael Kapralov. Improved bounds for online stochastic matching. In Mark de Berg and Ulrich Meyer, editors, *Proceedings of the 18th Annual European Symposium on Algorithms (ESA)*, volume 6346 of *Lecture Notes in Computer Science*, pages 170–181. 2010.

[3] Benjamin Birnbaum and Claire Mathieu. On-line bipartite matching made simple. *SIGACT News*, 39(1):80–87, 2008.

[4] Niv Buchbinder, Kamal Jain, and Seffi Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proceedings of the 15th Annual European Symposium on Algorithms (ESA)*, 2007.

[5] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. *Proceedings of 50th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 0:117–126, 2009.

[6] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.

[7] Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *Journal of the ACM*, 50:795–824, 2003.

[8] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the 43rd annual ACM symposium on Theory of Computing (STOC)*, 2011.

[9] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of Computing (STOC)*, pages 352–358, New York, NY, USA, 1990. ACM.

[10] Erik Krohn and Kasturi Varadarajan. Private communication reported by Goel and Mehta [6], 2007.

[11] Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. Allocating online advertisement space with unreliable estimates. In *Proceedings of 8th ACM Conference on Electronic Commerce (EC)*, pages 288–294, 2007.

[12] Mohammad Mahdian and Qiqi Yan. Uncapacitated facility location via strongly factor-revealing LPs. unpublished manuscript, 2011.

[13] Mohammad Mahdian, Yinyu Ye, and Jiawei Zhang. Approximation algorithms for metric facility location problems. *SIAM J. Comput.*, 36(2):411–432, 2006.

[14] Vahideh H. Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. In *Proceedings of 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.

[15] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5):22, 2007.

APPENDIX

A. MONOTONICITY INEQUALITIES

To prove the third constraint of polyLP$'(m)$, note that for $l' < p' \leq n$ and $r' \leq n$,

$$\begin{aligned} y'(l', r', p') &= \frac{1}{d} \sum_{(r,p)\in f(r',p')} y(l'd, r, p) \\ &= \frac{1}{d} \sum_{(r,p)\in f(r',p')} y(l'd, r, l'd+1) \\ &= \frac{1}{d} \sum_{(r,p)\in f(r',l'+1)} y(l'd, r, p) \\ &= y'(l', r', l'+1), \end{aligned}$$

where both second and the third line follow from the third constraint of polyLP(n). This constraint applies in the second line since $p' > l'$ and hence for every $p \in f(p')$, $p > l'd$. It applies in the third line since every $p \in f(l'+1)$ is at least $l'd + 1$.

For the fourth constraint of $\mathrm{polyLP}'(m)$, we have that for every $p' \leq l' < n$ and $r' \leq n$,

$$\begin{aligned} y'(l'+1, r', p') &= \frac{1}{d} \sum_{(r,p) \in f(r',p')} y(l'd + d, r, p) \\ &\geq \frac{1}{d} \sum_{(r,p) \in f(r',p')} y(l'd + 1, r, p) \\ &\geq \frac{1}{d} \sum_{(r,p) \in f(r',p')} y(l'd, r, l'd + 1) \\ &= \frac{1}{d} \sum_{(r,p) \in f(r',l'+1)} y(l'd, r, p) \\ &= y'(l', r', l'+1). \end{aligned}$$

Here the inequality on the second line follows from the definition of y and non-negativity of x (sixth and seventh constraints of $\mathrm{polyLP}(n)$), the third line follows from the fourth constraint of $\mathrm{polyLP}(n)$ (which applies here since $p \leq l'd$ for every $p \in f(p')$), and the fourth line follows from the third constraint of $\mathrm{polyLP}(n)$ (which applies since for every $p \in f(l'+1)$, $p > l'd$).

B. NUMERICAL SOLUTIONS

Table 1 shows the solutions of $\mathrm{polyLP}(n)$ and $\mathrm{polyLP}'(n)$ for $n = 1, \ldots, 40$ and $n = 50$. Solving these two LPs for $n = 50$ takes about 10 hours on a personal laptop using the CPLEX software, and solving them for a much larger value of n would need considerably more computational resource. These numbers are also plotted in Figure 4. The detailed solutions are available upon request.

Note that optimal value of $\mathrm{polyLP}'(n)$ for $n = 3$ already beats $1 - 1/e$. This gives a "manually verifiable" proof that the ranking algorithm with random arrivals beats $1 - 1/e$. The solution of $\mathrm{polyLP}'(3)$ is presented in Table 2.

Finally, we have computed the value of $\mathrm{expLP}(n, n)$ for $n \leq 6$. These values are

$$1, 0.75, 0.75, 0.742188, 0.737269, 0.732107.$$

There is a small positive gap between $\mathrm{expLP}(n)$ and $\mathrm{polyLP}(n)$ for these values of n. It is not clear if this gap persists asymptotically.

n	$\mathrm{polyLP}(n)$	$\mathrm{polyLP}'(n)$	n	$\mathrm{polyLP}(n)$	$\mathrm{polyLP}'(n)$
1	1	0.5	21	0.704624	0.692120
2	0.75	0.625	22	0.704360	0.692438
3	0.740741	0.641723	23	0.704129	0.692739
4	0.732456	0.657429	24	0.703923	0.693017
5	0.725007	0.667052	25	0.703730	0.693264
6	0.720263	0.673323	26	0.703546	0.693489
7	0.716508	0.677393	27	0.703368	0.693684
8	0.714067	0.680363	28	0.703208	0.693870
9	0.712352	0.682681	29	0.703064	0.694047
10	0.710998	0.684413	30	0.702930	0.694220
11	0.709908	0.685728	31	0.702806	0.694383
12	0.708957	0.686781	32	0.702688	0.694534
13	0.708131	0.687726	33	0.702577	0.694670
14	0.707474	0.688544	34	0.702467	0.694794
15	0.706884	0.689285	35	0.702365	0.694908
16	0.706416	0.689931	36	0.702272	0.695024
17	0.705981	0.690511	37	0.702185	0.695135
18	0.705592	0.691008	38	0.702104	0.695243
19	0.705236	0.691425	39	0.702025	0.695343
20	0.704906	0.691783	40	0.701950	0.695436

n	$\mathrm{polyLP}(n)$	$\mathrm{polyLP}'(n)$
50	0.701357	0.696150

Table 1: Optimal values of polyLP(n) and polyLP$'(n)$ for n up to 40, and for $n = 50$.

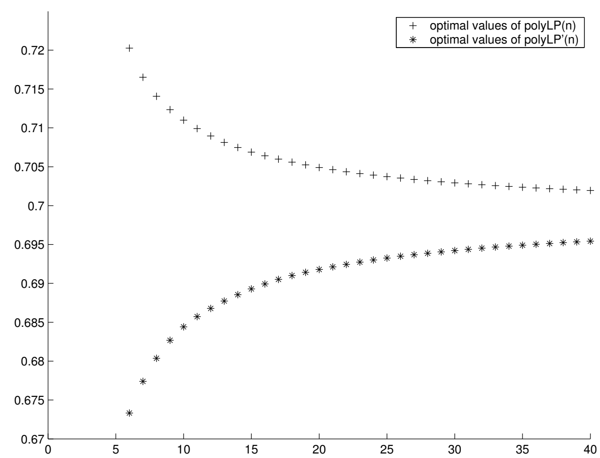

Figure 4: Optimal Values of polyLP(n) and polyLP$'(n)$

$$\begin{bmatrix} \frac{5}{21} & \frac{2}{21} & \frac{2}{21} \\ \frac{2}{21} & \frac{2}{21} & \frac{2}{21} \\ \frac{2}{21} & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} \frac{5}{21} & \frac{5}{21} & \frac{5}{21} \\ \frac{16}{147} & \frac{33}{147} & \frac{16}{147} \\ \frac{16}{147} & \frac{16}{147} & \frac{16}{147} \end{bmatrix}$$

$$y(1,.,.) \qquad y(2,.,.)$$

$$\begin{bmatrix} \frac{5}{21} & \frac{5}{21} & \frac{1}{3} \\ \frac{33}{147} & \frac{33}{147} & \frac{33}{147} \\ \frac{16}{147} & \frac{16}{147} & \frac{33}{147} \end{bmatrix}$$

$$y(3,.,.)$$

Table 2: Optimal solution of polyLP$'(3)$