https://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Vol. 72, No. 3, May–June 2024, pp. 1087–1104
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Crosscutting Areas

Dynamic Placement in Refugee Resettlement

Narges Ahani,[a] Paul Gölz,[b,*] Ariel D. Procaccia,[c] Alexander Teytelboym,[d] Andrew C. Trapp[e]

[a] Bank of America, Charlotte, North Carolina 28202; [b] Simons Laufer Mathematical Sciences Institute, Berkeley, California 94720; [c] School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138; [d] Department of Economics, University of Oxford, Oxford OX1 3UQ, United Kingdom; [e] WPI Business School and Data Science Program, Worcester Polytechnic Institute, Worcester, Massachusetts 01609

*Corresponding author

Contact: narges.ahani@bofa.com, https://orcid.org/0000-0002-6012-1707 (NA); mail@paulgoelz.de, https://orcid.org/0000-0002-8101-6818 (PG); arielpro@seas.harvard.edu, https://orcid.org/0000-0002-8774-5827 (ADP); alexander.teytelboym@economics.ox.ac.uk, https://orcid.org/0000-0002-6570-1903 (AT); atrapp@wpi.edu, https://orcid.org/0000-0003-0143-9093 (ACT)

Received: August 15, 2021
Revised: June 6, 2022; January 25, 2023
Accepted: March 16, 2023
Published Online in Articles in Advance: September 28, 2023

Area of Review: Policy Modeling and Public Sector OR

https://doi.org/10.1287/opre.2021.0534

Abstract. Employment outcomes of resettled refugees depend strongly on where they are initially placed in the host country. Each week, a resettlement agency is allocated a set of refugees by the U.S. government. The agency must place these refugees in its local affiliates while respecting the affiliates' annual capacities. We develop an allocation system that recommends where to place an incoming refugee family to improve total employment success. Our algorithm is based on two-stage stochastic programming and achieves over 98% of the hindsight-optimal employment, compared with under 90% of current greedy-like approaches. This dramatic improvement persists even when we incorporate a vast array of practical features of the refugee resettlement process including inseparable families, batching, and uncertainty with respect to the number of future arrivals. Our algorithm is now part of the Annie^{TM} MOORE optimization software used by a leading American refugee resettlement agency.

Funding: This work was supported by the UK Economic and Social Research Council [Grant ES/R007470/1]; the Office of Naval Research [Grant N00014-20-1-2488]; and the National Science Foundation's Division of Civil, Mechanical and Manufacturing Innovation [Grant CMMI-1825348], Division of Computing and Communication Foundations [Grants CCF-1733556 and CCF-2007080], Division of Mathematical Sciences [Grant DMS-1928930], and Division of Information and Intelligent Systems [Grant IIS-2024287].

Supplemental Material: The e-companion is available at https://doi.org/10.1287/opre.2021.0534.

Keywords: refugee resettlement • matching • stochastic optimization • integer optimization • humanitarian operations

1. Introduction

As of 2022, over 35 million people are seeking refuge outside their country of origin due to war, violence, or persecution (United Nations High Commissioner for Refugees 2023b). Of these refugees, the United Nations High Commissioner for Refugees (UNHCR) considers 2.4 million to be in need of resettlement, that is, permanent relocation from their country of asylum to a third country (United Nations High Commissioner for Refugees 2023b). Resettlement is mainly targeted at the most vulnerable refugees, such as children at risk, survivors of violence and torture, and those with urgent medical needs. Despite resettlement efforts by dozens of countries, global resettlement falls far short of what would be required. In 2022, for example, only around 114,000 refugees were resettled (United Nations High Commissioner for Refugees 2023a), whereas the projected need for resettlement grew by 400,000 over a similar time frame (United Nations High Commissioner for Refugees 2023b). Given this dearth of resettlement capacity, countries need to use their limited places as effectively as possible in order to maximize refugee welfare.

Historically, countries taking in resettled refugees have paid little attention to where inside the country these refugees are placed. This policy might be worth reconsidering in light of ample evidence that the local resettlement destination dramatically affects key socioeconomic outcomes for refugees (Åslund and Rooth 2007; Åslund and Fredriksson 2009; Åslund et al. 2010, 2011; Damm 2014; Bansak et al. 2018; Martén et al. 2019). One specific outcome impacted by community placement is whether and when resettled refugees find employment (e.g., Åslund and Rooth 2007), which plays a key role in the successful integration of a refugee by "promoting economic independence, planning for the future, meeting members of the host society, providing

opportunity to develop language skills, restoring self-esteem and encouraging self-reliance" (Ager and Strang 2008, p. 170).

To help more refugees find employment, the U.S. resettlement agency HIAS (founded as the Hebrew Immigrant Aid Society) started in 2018 to match refugees to communities with the software Annie™ MOORE (Matching and Outcome Optimization for Refugee Empowerment) developed by Ahani et al. (2021). Based on past arrival data, Annie MOORE estimates how likely a refugee is to find employment in each community soon after arrival. The software then suggests where to place the refugee to maximize the expected total employment, subject to not exceeding community capacities and to ensuring that refugees have access to services they need.

A key limitation of the existing software, however, was that it solved an *offline* optimization problem, whereas refugee allocation is an *online* problem: whereas Annie MOORE optimized a one-shot matching of refugees to communities, organizations like HIAS continuously allocate refugees over the year as they are cleared for resettlement, and they aim to maximize the total employment across the year. Because of this mismatch, resettlement practitioners employed Annie MOORE as a greedy algorithm; that is, Annie MOORE myopically maximized the employment of the current batch of refugees, without considering whether the current assignment would negatively impact the employment of future arrivals in the same fiscal year by prematurely using up community capacity.

In this paper, we design and deploy an *online* algorithm for refugee allocation. This algorithm achieves higher employment by explicitly accounting for the value that a community's capacity has for the employment of future arrivals, which we refer to as the community's *potential*. In fact, we design *two* closely related algorithms, defined by different ways of computing potentials from the dual values of a matching linear program. One of these potentials is motivated by stochastic programming and the other by Walrasian equilibrium. We study these algorithms in a rich model that captures all of the relevant practical features of the refugee resettlement process, including inseparable families of refugees, batching, and unknown numbers of refugee arrivals. Evaluating our algorithms on HIAS data from 2014 to 2019, we show that they achieve over 98% of the hindsight-optimal employment in all years, compared with a typical employment of around 90% for the greedy baseline. We then describe how we implemented our algorithms within Annie MOORE to create Annie 2.0, which has been well-received by HIAS leadership: "*Annie*™ 2.0 is a game-changer for our pre-arrivals processes, allowing us to plan and optimize our pre-arrival strategy a year rather than a week ahead." The code for our experiments is available at https://github.com/pgoelz/dynamicrefugees, and Section EC.1 of the e-companion contains a detailed descriptions of our data set, as well as details on data preprocessing.

1.1. Related Work

Our paper extends a line of work initiated by Bansak et al. (2018), which aims to increase refugees' employment outcomes through data- and optimization-driven placement. This approach consists of two components: using machine learning to estimate the probability that a given refugee placed at a given community would find employment, and using mathematical programming to perform the optimization. Ahani et al. (2021) adopted a similar approach to develop Annie MOORE; they also pointed out the practical relevance of inseparable families and the possibility of batching. Both papers seek to maximize employment with respect to a current batch of refugees, without considering future arrivals. In this sense, we think of the previously deployed algorithms as *greedy*, and that is indeed our benchmark in this paper.

Though our dynamic refugee placement problem generalizes the classic *edge-weighted online bipartite matching* problem, most algorithms in the theory literature are not promising for our application because they are optimized for overly pessimistic arrival scenarios. Whereas competitive analysis was quite successful for unweighted online bipartite matching (Karp et al. 1990), no constant-factor approximation algorithm is possible for the weighted setting if arrivals are adversarial (Fahrbach et al. 2020). In the random-order arrival model, a $1/e$-approximation is possible (Kesselheim et al. 2013), but the algorithm is impractical; in particular, it leaves the first 37% of arrivals unmatched. Even if arrivals are drawn independent and identically distributed from a known distribution, Manshadi et al. (2012) show that no online algorithm can obtain a better approximation ratio than 0.823, far below the performance of even the greedy baseline in our setting. Because this impossibility is based on highly adversarial arrival distributions, many papers additionally assume that arrivals belong to finitely many types determining their edge weights. In this setting, constructing matchings that are optimal up to lower-order terms (with high probability) is not difficult (see Alaei et al. 2013), and multiple papers obtain such results, often in generalizations of edge-weighted online bipartite matching (Alaei et al. 2012, 2013; Vera and Banerjee 2021). What limits the applicability of these algorithms to our setting, however, is that these algorithms require the distribution over types explicitly in their input, and are often constructed based on the assumption that multiple arrivals of each type will occur in a single run of the algorithm. By contrast, we estimate employment scores based on 20 independent features, which means the number of refugee "types" is too large to enumerate, and we do not expect to see identical refugees.

Our allocation algorithms work by simulating sequences of future arrivals, computing the optimal matchings for these simulated futures, and then allocating the current arrival based on the shadow prices of these optimal matchings. This approach can be seen as an instantiation of the *Bayes selector*, an algorithmic paradigm that takes in a prediction of future arrivals and then performs the action (in our setting, chooses the affiliate for the current arrival) that seems most likely to coincide with the action taken by an optimal benchmark. Across various problems with stochastic online arrivals, algorithms following this paradigm have yielded impressive theoretical and empirical results (Freund and Banerjee 2019, Banerjee et al. 2020, Vera and Banerjee 2021, Vera et al. 2021, Sinclair et al. 2023). Specifically, under some regularity conditions on arrivals, these algorithms obtain *constant regret*; that is, the expected difference between the algorithm's performance and that of the optimal benchmark does not grow with the size of the problem. The prediction of future arrivals often takes other shapes, but it can be a sampled trajectory of arrivals as in our algorithms (Banerjee et al. 2020). In most papers, the choice of action is based on how *often* the optimal benchmark would take an action in the simulated future, rather than, as in our work, on the marginal effect of an action on the optimal value. In recent work, however, Sinclair et al. (2023) analyzed the same variant of the Bayes selector (the "hindsight planning policy") as our Equation (1) and showed that it gives constant regret for the problem of stochastic online bin packing. Even though we do not provide theoretical guarantees in this paper, the success of the Bayes selector across related settings partially explains our good empirical performance.

Our use of shadow prices for guiding online refugee allocation mirrors earlier applications of shadow prices to a variety of online decision problems in, among other contexts, advertising (Vazirani et al. 2005, Devanur and Hayes 2009, Vee et al. 2010), revenue management (Talluri et al. 2004), worker assignment (Ho and Vaughan 2012, Johari et al. 2021), and resource allocation (Asadpour et al. 2020). Agrawal et al. (2014) develop a dynamic learning approach where prices are calculated in a similar manner to ours, but whereas they update their match scores upon every doubling of the arrival history, we update our match scores upon every batch. Ho and Vaughan (2012) extend the advertising context of Devanur and Hayes (2009) to assign workers to tasks when match scores are initially unknown and must be learned. Like Ho and Vaughan (2012), Johari et al. (2021) also consider the worker-to-job context, but learn scores while matching via an explore-then-exploit approach. In our setting, our scores are known in advance independent of arrivals (Ahani et al. 2021).

In independent and concurrent work, Bansak (2020) also considers dynamic refugee resettlement, albeit in a model of the problem that is more stylized than ours. Our model is more representative of practical refugee matching through explicit inclusion of nonunit family sizes, incompatibilities between families and communities, and uncertain arrival numbers. Out of the four assignment algorithms studied by Bansak (2020), the first two are closely related to algorithms we develop. (The other two algorithms lead to substantially worse employment in Bansak's (2020) experiments, which is why we do not discuss them here.) Bansak's (2020) algorithm 1, which achieves the best employment in his experiments, is a straightforward sampling implementation of the two-stage stochastic programming formulation in Section 4.1. As we explain in that section, our algorithm PM(Pot1) is *functionally equivalent* to Bansak's (2020) algorithm 1 in his model; thus, PM(Pot1) would obtain the same great employment outcomes as in Bansak's (2020) experiments. An important difference, however, is that our algorithms are orders of magnitude faster than Bansak's (2020) algorithm 1, as shown in Section 6. This allows us to evaluate our algorithms on entire fiscal years of arrivals (whereas Bansak's (2020) evaluation is limited to individual months) and enables our algorithms to scale to large future increases in arrivals numbers (see Section EC.4.3 of the e-companion). To overcome his algorithm 1's slow running time, Bansak (2020) proposes algorithm 2, another instantiation of the Bayes selector, which has a comparable running time to our algorithms. Whereas our algorithms match a current case based on shadow prices for a large number of simulated arrival trajectories, Bansak's (2020) algorithm 2 matches the current case with the affiliate to which it was most frequently matched in the offline solutions for the arrival trajectory. Given that Bansak's (2020) experiments reveal that his algorithm 2 obtains nearly as much employment as his algorithm 1 (for an unspecified number of arrival trajectories), his algorithm 2 and PM(Pot1) can be expected to be comparable in terms of employment and running time. Recent work by Bansak and Paulson (2022) extends the earlier work by Bansak (2020) by incorporating a secondary objective that seeks to consume capacity at similar rates across affiliates, improving case wait times across affiliates without sacrificing much employment. Even more recently, Freund et al. (2023) employ a bid-price approach related to ours to enforce fairness between subgroups of refugees.

1.2. Organization of This Paper

In Section 2, we provide an overview of the U.S. refugee resettlement process. In Section 3, we outline our model of dynamic refugee matching. In Section 4, we propose our two algorithms and show that they obtain near-optimal employment in a baseline setting that ignores the inseparability of families, batching, and uncertainty about the total number of arrivals. In the next three sections, we layer on complexity toward the setting encountered in practice: families of different sizes (Section 5), batching (Section 6), and unknown arrival numbers (Section 7). In

these sections, we demonstrate that inseparable families and batching do not substantially change our algorithms' employment performance, and that employment remains high unless the number of arrivals widely deviates from the numbers announced by the government. In Section 8, we then explain how we implemented our approach within Annie MOORE, and we conclude in Section 9. In the e-companion, we provide details deferred from the main text for space reasons and additional empirical analyses.

2. Institutional Background

The federal Office of Refugee Resettlement was created by the Refugees Act in 1980. The act established funding rules and authorized the president of the United States to set annual capacities for resettlement. The resettlement process is managed by the U.S. Refugee Admissions Program (USRAP) of the U.S. Department of State, in conjunction with a number of federal agencies across federal departments as well as the International Organization for Migration and the UNHCR.

Applications for the resettlement program take place from outside of the Unites States, typically in refugee camps. The U.S. government conducts security checks and medical screenings, and performs cultural orientation, which can take upward of 18 months (Jones 2015). After clearance, the USRAP decentralizes the process of welcoming refugees to nine nongovernmental organizations known as *resettlement agencies*, one of which is HIAS. Each agency works with their own network of local *affiliates*, each supported by local offices as well as religious entities like churches, synagogues, or mosques, which serve as community liaisons for refugees. Each agency typically works with dozens of affiliates, though the exact number of affiliates fluctuates over time. Some affiliates lack services to host certain kinds of refugees. For example, certain affiliates do not have translators for non-English-speaking refugees, or they might lack support for single-parent families.

Agencies have no influence on which refugees are cleared for resettlement by the USRAP or on when the refugees might arrive. Resettlement agencies meet on a weekly or fortnightly basis to allocate among themselves the refugees that have been cleared by the USRAP.

Refugees are usually resettled with members of their family. Such an inseparable group of refugees is referred to as a *case*. As a family can split when its members are fleeing their home country, some refugees who are applying for resettlement might already have existing relatives or connections in the United States. Such cases *with U.S. ties* are automatically resettled near their existing ties. All other refugees, referred to as *free cases*, can be resettled by any agency into any of the agency's affiliates.

Each affiliate has an assigned annual capacity for the number of individual refugees it can admit in a given fiscal year.[1] These capacities are approved by the USRAP and, in theory, agencies cannot exceed them. In practice, capacities can be slightly adjusted toward the end of the year or, as in recent years, substantially revised in the course of the year. Because capacities limit the number of refugees *arriving* in a fiscal year rather than *allocated* in it, and because there is typically a delay of multiple months between the two events, the U.S. Department of State tells the resettlement agencies an estimated arrival date for each cleared case.

Agencies are assessed annually by the USRAP on their performance in finding employment for refugees within 90 days of their arrival. Data on 90-day employment is therefore diligently collected by the affiliates and monitored by the agencies.

3. Model

An instance of the matching problem first defines a set L of *affiliates*, and each affiliate ℓ has a capacity $c_\ell \in \mathbb{N}_{\geq 0} \cup \{\infty\}$ of how many refugees it can host. We call a vector $\boldsymbol{c} = \{c_\ell\}_{\ell \in L}$ of capacities for all affiliates a *capacity profile* $\boldsymbol{c}$. We write $\boldsymbol{c} - \boldsymbol{e}_\ell$ to describe the capacity profile obtained from $\boldsymbol{c}$ by reducing the capacity of affiliate ℓ by one.

On the other side of the matching problem is a set $N = \{1, \ldots, n\}$ of *cases*. Each case i represents an inseparable family of $s_i \in \mathbb{N}_{\geq 1}$ refugees. Furthermore, each case i, for each affiliate ℓ, has an *employment score* $u_{i,\ell}$, which indicates the expected number of case members that will find employment if the case is allocated to ℓ. Typically, these employment scores $u_{i,\ell}$ are real numbers in $[0, s_i]$, but we will also allow to set $u_{i,\ell} = -\infty$ to express that case i is not compatible with affiliate ℓ. We will refer to the combination of a case's size and vector of employment scores as the *characteristics* of the case. To ensure that the matching problem is always feasible, we will assume that L contains a special affiliate $\perp$ that represents leaving a case unmatched, where $u_{i,\perp} = 0$ for all cases i and $c_\perp = \infty$.[2]

We use the employment scores developed by Ahani et al. (2021), and we give details on data preprocessing and training in Section EC.1.1 of the e-companion. Throughout this paper, we consider these employment scores as ground truth, which means that we evaluate algorithms directly based on the employment scores. An evaluation of how accurately the employment scores predict employment outcomes is outside of the scope of this paper, and has already been performed by Ahani et al. (2021).

The goal of the matching problem is to allocate cases to affiliates such that the *total employment*, that is, the sum of employment scores, is maximized, subject to not exceeding capacities. For a set $I \subseteq N$ and a capacity profile $\boldsymbol{c} = \{c_\ell\}_{\ell \in L}$, define $\textsc{Matching}(I, \boldsymbol{c})$ as the matching integer linear program (ILP) below, where variables $x_{i,\ell}$ indicate whether case $i \in I$ is matched to

affiliate $\ell \in L$:

$$\begin{aligned} \text{maximize} \quad & \sum_{i\in I}\sum_{\ell\in L} u_{i,\ell}\, x_{i,\ell} & \\ \text{subject to} \quad & \sum_{\ell\in L} x_{i,\ell} = 1 & \forall i \in I, \\ & \sum_{i\in I} s_i\, x_{i,\ell} \le c_\ell & \forall \ell \in L, \\ & x_{i,\ell} \in \{0,1\} & \forall i \in I, \ell \in L. \end{aligned}$$

Let $\textsc{Opt}(I, \boldsymbol{c})$ denote the optimal objective value of $\textsc{Matching}(I, \boldsymbol{c})$. The *linear programming* (*LP*) *relaxation* of $\textsc{Matching}(I, \boldsymbol{c})$ is obtained by replacing the constraint $x_{i,\ell} \in \{0,1\}$ by $0 \le x_{i,\ell} \le 1$ for all $i \in I, \ell \in L$. For a fixed matching, we define the *match score* of a case i as its employment score u_{i,ℓ_i} at the affiliate ℓ_i where it is allocated; we will also refer to its *match score per refugee*, $u_{i,\ell_i}/s_i$.

Finally, cases arrive *online*, that is, they arrive one by one, and when case i arrives, the decision of which affiliate to place i in must be made irrevocably, before the characteristics of the subsequent arrivals $i+1, \dots, n$ are known.[3] Thus, although an online matching algorithm must still produce a matching whose indicator variables $x_{i,\ell}$ satisfy the constraints of $\textsc{Matching}(N, \boldsymbol{c})$, the total employment $\sum_{i\in N, \ell\in L} u_{i,\ell}\, x_{i,\ell}$ typically will not attain the benchmark $\textsc{Opt}(N, \boldsymbol{c})$ of the *optimal matching in hindsight*. Although we will not commit to a specific model of how the characteristics of arriving cases are generated, these arrivals should be thought of as *stochastic* rather than worst case, and the distribution of case characteristics as changing slowly enough that sampling from recent arrivals is a reasonable proxy for the distribution of future arrivals.

Note that we use the word "arriving," as is customary in online algorithms, to refer to the time at which a case is presented to and allocated by the algorithm. Unfortunately, this usage disagrees with the terminology of refugee resettlement, where "arrival" refers to a refugees *physical arrival* in the United States, which takes place some time after allocation. As we have described in Section 2, this physical arrival plays a single role in our allocation problem, namely, by determining which fiscal year's capacities a case counts toward. Because we treat the allocation problems for different fiscal years separately, the important point in time associated with a case is its time of allocation, which we will refer to as its *arrival* for the remainder of this paper.

Throughout the following sections, we will consider a sequence of models that incorporate an increasing number of features of the real-world refugee allocation problem. In Section 4, we consider traditional online bipartite matching, which results from requiring $s_i = 1$ in the above model. From Section 5 onward, we allow cases to have arbitrary size. From Section 6 onward, we also allow cases to arrive in batches rather than one by one. In Section 7, we no longer assume that the total number n of arriving cases is known to the algorithm.

4. Online Bipartite Matching ($s_i = 1$)

In this section, we will consider the special case of online bipartite (weighted) matching. We stress that this classic problem does not capture key features of the refugee-allocation problem in practice, which we will add in later sections. Instead, online bipartite matching allows us to more cleanly draw connections to theoretical arguments, which help motivate our algorithm design. Later in the paper, we will empirically show that the approach continues to work well in richer and more realistic settings.

Formally, this section considers the model defined in the previous section, with the restriction that all cases consist of single refugees, that is, that $s_i = 1$ for all $i \in N$. Under this assumption, it is well known that the optimum matching for the ILP $\textsc{Matching}(I, \boldsymbol{c})$ can be found by solving its LP relaxation.

4.1. Algorithmic Approach

To motivate our algorithmic approach, we begin by describing why matching systems currently deployed in practice lead to suboptimal employment. These systems assign cases *greedily*, which—putting aside batching for now—means that an arriving case i is matched to the affiliate ℓ with highest employment score $u_{i,\ell}$ among those that have at least s_i remaining capacity. The main problem with greedy assignment is that it exhausts the capacity of the most desirable affiliates too early. In particular, we observe on the real data that a large fraction of cases have their highest employment score in the same affiliate ℓ^*, but that the size of the employment advantage of affiliate ℓ^* over the second-best affiliate varies. Because it considers only the highest-employment affiliate for each case, greedy assignment will fill the entire capacity of ℓ^* early in the year, including with some cases that benefit little from this assignment. Consequently, cases that would particularly profit from being placed in ℓ^* but arrive later in the year no longer fit within the capacity.

Intuitively, the decision to match a case i to an affiliate ℓ has two effects: the immediate increase of the total employment by $u_{i,\ell}$ but also an opportunity cost for consuming ℓ's capacity, which might prevent profitable assignments for later arrivals. Because greedy assignment considers only the former effect, it leaves employment on the table.

A better approach is *two-stage stochastic programming*, which allocates an arriving case i to the affiliate ℓ maximizing the sum of the immediate employment $u_{i,\ell}$ and the expected optimal employment obtainable by matching the future arrivals subject to the remaining capacity. That is, if, at the time of i's arrival, the remaining capacities are given by $\boldsymbol{c}$, two-stage stochastic programming

allocates i to the affiliate

$$\arg\max_{\ell \in L: c_\ell \geq s_i} \quad u_{i,\ell} + \mathbb{E}[\textsc{Opt}(\{i+1,\dots,n\}, \boldsymbol{c} - s_i \cdot \boldsymbol{e}_\ell)],$$

where the expectation is taken over the characteristics of cases $j = i+1,\dots,n$. Because adding a constant term does not change the argmax, this can be rewritten as

$$\begin{aligned}
&= \arg\max_{\ell \in L: c_\ell \geq s_i} \quad u_{i,\ell} - \mathbb{E}[\textsc{Opt}(\{i+1,\dots,n\}, \boldsymbol{c})] \\
&\quad + \mathbb{E}[\textsc{Opt}(\{i+1,\dots,n\}, \boldsymbol{c} - s_i \cdot \boldsymbol{e}_\ell)] \\
&= \arg\max_{\ell \in L: c_\ell \geq s_i} \quad u_{i,\ell} - \mathbb{E}\big[\textsc{Opt}(\{i+1,\dots,n\}, \boldsymbol{c}) \\
&\quad - \textsc{Opt}(\{i+1,\dots,n\}, \boldsymbol{c} - s_i \cdot \boldsymbol{e}_\ell)\big]. \qquad (1)
\end{aligned}$$

Using our assumption that $s_i = 1$, this can be simplified to

$$\begin{aligned}
&= \arg\max_{\ell \in L: c_\ell \geq 1} \quad u_{i,\ell} - \mathbb{E}\big[\textsc{Opt}(\{i+1,\dots,n\}, \boldsymbol{c}) \\
&\quad - \textsc{Opt}(\{i+1,\dots,n\}, \boldsymbol{c} - \boldsymbol{e}_\ell)\big].
\end{aligned}$$

Note that the expected value that is subtracted in either of the last two lines is exactly the expected opportunity cost of reducing the capacity of ℓ by placing case i there. This motivates our algorithmic approach: In every time step, we first compute a *potential* p_ℓ for each affiliate ℓ. Then, rather than myopically maximizing the matching score as does greedy assignment, our algorithm PM ("potential match") myopically maximizes the matching score minus the potential of the capacity used, as shown in Algorithm 1. (Note that an affiliate ℓ can always be defined in Line 5 as, by assumption, $c_\perp = \infty$.)

Algorithm 1 (PM(Potential))

Parameter: a subroutine Potential to determine affiliate potentials

1 initialize the capacities c_ℓ for each affiliate ℓ;
2 **for** $t = 1,\dots,n$ **do**
3 observe the case size s_t and the employment scores $\{u_{t,\ell}\}_\ell$;
4 call Potential() to define a potential p_ℓ for each affiliate ℓ;
5 $\ell \leftarrow \arg\max_{\ell \in L: c_\ell \geq s_t} u_{t,\ell} - s_t p_\ell$;
6 allocate case t to ℓ and set $c_\ell \leftarrow c_\ell - s_t$;

We estimate the expected value of the opportunity cost by averaging over a fixed number k of *trajectories*, each of which consists of randomly sampled characteristics of all arrivals $i+1$ through n. As the characteristics of arriving refugees change over time, and as these changes tend to be gradual, we draw these arrival characteristics uniformly with replacement from the arrivals in the six months prior to the current allocation decision. In Section EC.4.4 of the e-companion, we evaluate different lengths of this sampling window.

For each sampled trajectory, it remains to calculate the potential, which we would like to equal the opportunity cost $\textsc{Opt}(\{i+1,\dots,n\}, \boldsymbol{c}) - \textsc{Opt}(\{i+1,\dots,n\}, \boldsymbol{c} - \boldsymbol{e}_\ell)$. Clearly, this could be computed by solving $\mathcal{O}(|L|)$ matching linear programs, which is what algorithm 1 by Bansak (2020) does.

Instead, we make use of an important observation in matching theory (Leonard 1983) to exactly compute the opportunity costs for *all* affiliates with remaining capacity as the shadow prices of a *single* LP.

Fact 1. Fix a matching-problem instance, in which all cases i have size $s_i = 1$. In the LP relaxation of $\textsc{Matching}(N, \boldsymbol{c})$, let $\{p_\ell\}_{\ell \in L}$ denote the unique element-wise maximal set of shadow prices for the constraints $\sum_{i \in N} s_i x_{i,\ell} \leq c_\ell$. Then, for each ℓ with $c_\ell \geq 1$,

$$p_\ell = \textsc{Opt}(\{i+1,\dots,n\}, \boldsymbol{c}) - \textsc{Opt}(\{i+1,\dots,n\}, \boldsymbol{c} - \boldsymbol{e}_\ell).$$

This suggests the procedure Pot1 for computing potentials, which is shown in Algorithm 2. (One way of finding the element-wise maximal shadow prices is to first solve the dual LP to find its objective value, then add a constraint that constrains the objective of the dual LP to be equal to this optimal objective value, and to finally maximize the sum of dual variables p_ℓ over this new restricted LP.)

Algorithm 2 (Pot1(k))

Parameter: $k \in \mathbb{N}_{\geq 1}$, the number of trajectories per potential computation

Input: remaining capacities $\boldsymbol{c}$, the index t of the last observed case, characteristics of cases arriving in the past 6 months

Output: a set of potentials p_ℓ for all affiliates ℓ

1 **for** $j = 1,\dots,k$ **do**
2 for each $i = t+1,\dots,n$, set s_i and $\{u_{i,\ell}\}_\ell$ to the size and employment scores of a random, recently arrived case;
3 solve the following bipartite-matching LP:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=t+1}^{n} \sum_{\ell \in L} u_{i,\ell} x_{i,\ell} \\
\text{subject to} \quad & \sum_{\ell \in L} x_{i,\ell} = 1 && \forall i = (t+1),\dots,n \\
& \sum_{i=t+1}^{n} s_i x_{i,\ell} \leq c_\ell && \forall \ell \in L \qquad (*) \\
& 0 \leq x_{i,\ell} && \forall i = (t+1),\dots,n,\ \forall \ell \in L.
\end{aligned}$$

4 for each ℓ, set p_ℓ^j to be the maximal shadow price of the constraint $(*)$;
5 set $p_\ell \leftarrow (\sum_{j=1}^{k} p_\ell^j)/k$ for all ℓ;
6 **return** $\{p_\ell\}_{\ell \in L}$;

We also develop a second method, Pot2, for computing potentials, which is based on a slightly different LP and has different theoretical underpinnings:

- whereas the matching LP for Pot1 does not include the current batch of arrivals, the current batch is included in the LP for Pot2;
- whereas Pot1 uses the element-wise maximal set of shadow prices, Pot2 uses the element-wise minimal one; and
- whereas Pot1 is theoretically derived from two-stage stochastic programming, Pot2 is motivated by a connection to Walrasian equilibria.

For conciseness, we defer the formal definition of Pot2 and its connection to the Walrasian equilibrium to Section EC.2 of the e-companion.

4.2. Empirical Evaluation

We evaluate the employment of our potential-based matching algorithm on six real sequences of annual arrivals at HIAS; that is, for each fiscal year, we consider all refugees who physically arrived during this fiscal year, and we consider them in the order in which they were received for allocation by HIAS. For the capacities, we use the year's *final*, that is, most revised, capacities.[4] We also immediately take into account that affiliates have constraints on which nationalities, languages, and family sizes they can accommodate, that not all affiliates can host single parents, and that tied cases can only be allocated to their corresponding affiliate.

The main way in which this experiment deviates from reality is the assumption (made throughout this section) that cases have unit size. To satisfy this assumption, we split each case of size $s_i > 1$ into s_i identical single-refugee cases with a $1/s_i$ fraction of the original employment scores. In subsequent sections, we will repeat the experiments without this modification.

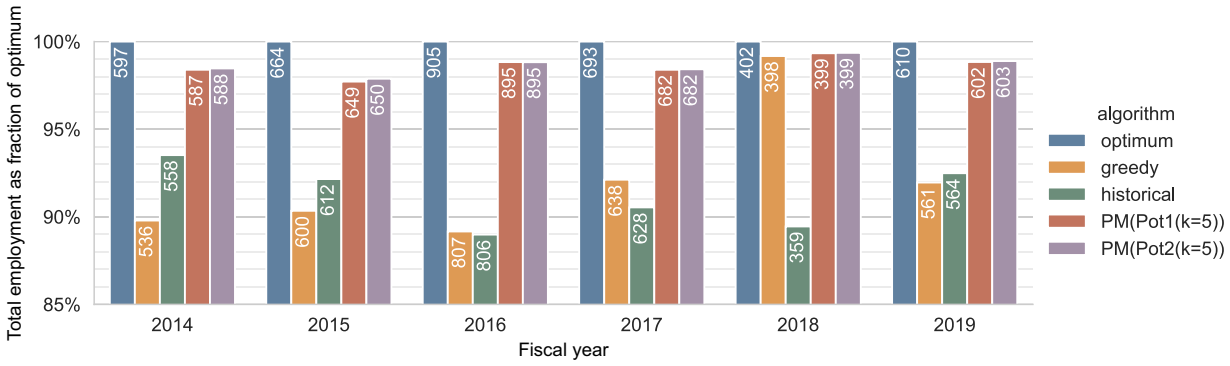
We study six fiscal years, from 2014 to 2019. As affiliates closed and opened across these years, the number of affiliates varies between 16 and 24 (not counting the unmatched affiliate $\perp$). Finally, the number of arriving refugees (respectively, cases) varies between 1,670 (respectively, 640) and 4,150 (respectively, 1,630) across fiscal years. For further metrics of the allocation problem, see Section EC.1.2 of the e-companion.

As shown in Figure 1, even the greedy baseline obtains a total employment of between 89% and 92% of $\textsc{Opt}(N, \boldsymbol{c})$, the optimum matching in hindsight. (One outlier is the year 2018, which we discuss below.) Nevertheless, the greedy algorithm leads to between 50 and 100 fewer refugees finding employment every year compared with what would have been possible in the optimum matching. Our potential algorithms close a large fraction of this gap, obtaining between 98% and 99% of the optimal total employment, both for algorithms based on Pot1 and for those based on Pot2. Because experiments in this model take much longer to run than those in subsequent models, we defer a comparison between the two potential methods and between different numbers k of trajectories to Section 6.1, where we can run the potential algorithms a sufficient number of times to discern smaller differences.

The fiscal year 2018 stands out from the others because the greedy algorithm performs on par with the potential algorithms, at 99% of the hindsight-optimal total employment. This is easily explained by the fact that the capacities are much looser than in other fiscal years: whereas, in all other fiscal years between 2014 and 2019, the number of arriving refugees amounts to between 84% (2019) and 97% (2016) of the final total capacity across all affiliates, this fraction is only 48% in 2018. Because capacity is so abundant, the optimal matching will match a large fraction of cases to their maximum-score affiliate, and the greedy matching is close to optimal.

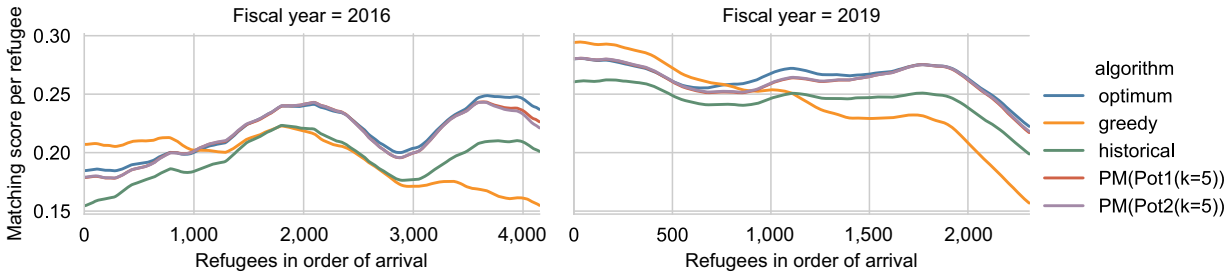
We also compare with the employment obtained by the allocation chosen by HIAS ("historical"). This comparison gives the historical matching a slight advantage, as HIAS sometimes overrides the incompatibility between an

Figure 1. (Color online) Total Employment Obtained by Different Algorithms, Assuming That Cases Are Split into Multiple Cases of Size 1



Notes. Capacities are the final capacities of the fiscal year. For the potential algorithms, total employment is averaged over 10 random runs. The numbers in the bars denote the absolute total employment; the bar height indicates the proportion of the optimum total employment in hindsight.

Figure 2. (Color online) Evolution of the Per Refugee Match Score in Order of Arrival for Fiscal Years 2016 and 2019 in the Experiment in Figure 1 (Split Cases, Final Capacities)



Note. Consecutive match scores are smoothed using triangle smoothing with width 500.

affiliate and a case, which we do not allow any other algorithm to do.[5]

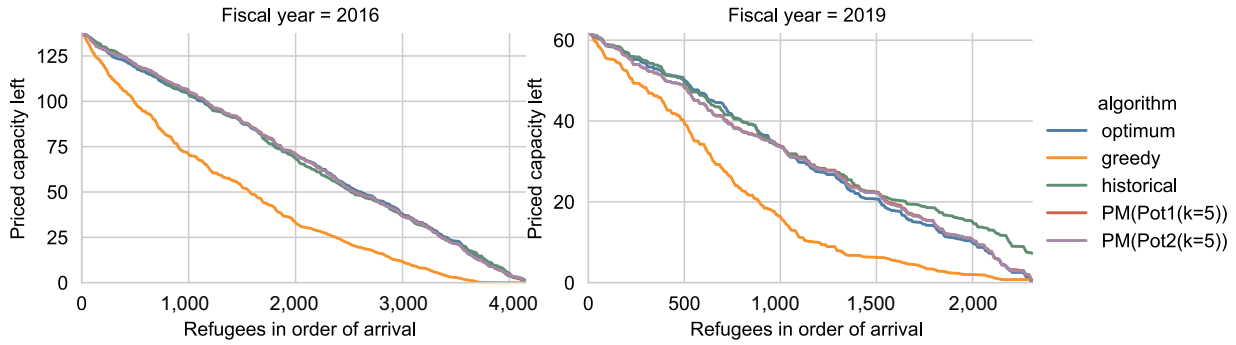
In Figure 2, we investigate how the match score changes over the course of two fiscal years, 2016 and 2019, chosen to contain one year in which the greedy and historical baselines perform relatively poorly (2016) and one in which they perform well (2019). As the match score of subsequently arriving refugees can greatly differ, these graphs are heavily smoothed over time. If arrivals were drawn from a time-invariant distribution, we would expect the curves for the optimum matching in hindsight to be level, because how much employment the optimum matching can extract from a case would be independent of the case's arrival time. Instead, we see that the employment prospects of arrivals fluctuate noticeably over time; in particular, the early refugees in fiscal year 2016 and the late refugees in fiscal year 2019 seem to have worse employment prospects than other refugees in the plot.

The curves for both potential algorithms are nearly indistinguishable from one another, which shows that the algorithms make very similar decisions. In 2016, these curves start out closely tracking the curve of the optimal-hindsight matching, but fall behind for the last arrivals, which we observe in most fiscal years. The similarity of the curves over most of the year indicates that our approach of sampling trajectories from past arrivals is nearly as useful as the optimum algorithm's perfect knowledge of future arrivals and that it leads to a similar trade-off in extracting immediate employment versus preserving capacity for later arrivals. Of course, the imperfect knowledge of the future incurs a small loss toward the end of the fiscal year, likely because the amount of capacity reserved per affiliate does not perfectly match the demand, which explains the gap in total employment between the hindsight optimum and the potential algorithms. This typical end-of-year effect is not very pronounced in fiscal year 2019, likely because the final arrivals of fiscal year 2019 have lower employment probabilities than what would be expected based on past arrivals. Instead, the potential algorithms fall behind the optimum algorithm for some period in the middle of the year, perhaps because they are reserving capacity for late arrivals which the optimum already knows to hold little promise.

The most striking curve is that of the greedy algorithm, which lies above those of all other algorithms in the first quarter of arrivals, but then falls clearly below the other curves in the second half. This observation can be explained by the effect we predicted in the motivation of our potential approach: the greedy algorithm extracts small additional gains in employment early in the arrival period, at the cost of prematurely consuming the capacity of the most desirable affiliates. Then, the lack of capacity limits the match scores of later arrivals, resulting in an overall unfavorable trade-off. This effect can be directly seen in Figure 3, in which we visualize the amount of capacity remaining in the most valuable affiliates. Specifically, looking at all arrivals of the fiscal year, we compute the shadow prices of the matching LP. At any point in time, we can then weight the remaining capacity by these prices to obtain a *priced capacity*. In Figure 3, we see that the optimum-hindsight matching and the potential algorithms use up the priced capacity at a roughly constant pace and essentially consume it all. By contrast, the greedy algorithm uses up the capacity very quickly, such that at the median refugee, only 22% (2016) or 17% (2019) of the priced capacity is left.

The historical matching made by HIAS does not have such obvious defects, but still falls short in terms of total employment. In both reference years, the average employment moves in parallel with the optimum matching, meaning that HIAS does not overly focus on extracting employment at certain parts of the fiscal year at the expense of others. However, the average employment consistently lies below that of the optimum and of the potential algorithms. We see in Figure 3 that, in 2019, HIAS started consuming the priced capacity at a near-constant pace very similar to that of the optimum algorithm. Around the median arrival, however, the historical matching slowed down its capacity consumption and ended up not consuming all priced capacity, which explains some loss in total employment. One reason for this behavior might be that HIAS staff treat the last 9% of

Figure 3. (Color online) Remaining Priced Capacity at the Time of Arrival of Different Refugees, for Fiscal Years 2016 and 2019 in the Experiment in Figure 1 (Split Cases, Final Capacities)



the capacity as a reserve that they are more reluctant to use. In a year such as 2019, in which the overall arrivals were only 84% of the total capacity, this heuristic might have actually kept much of the reserve capacity free, including in the affiliates that could have generated higher employment. By contrast, the total arrivals in 2016 amounted to 97% of the overall capacity, which could explain why nearly all priced capacity was consumed in this year. Despite using up priced capacity in a similar pattern as the optimum matching in 2016, the historical assignment achieved lower matching scores throughout the year. This indicates that the low employment of the historical matching is not just due to a reluctance to use the entire capacity, but that the priced capacity is furthermore inefficiently allocated.

5. Nonunit Cases ($s_i \geq 1$)

The most pressing aspect of refugee matching that we have ignored thus far is that many cases do not consist of individual refugees. Instead, they consist of an entire family of refugees, which has to be resettled to the same affiliate.

To accommodate cases consisting of multiple family members, we will from now drop the assumption that the s_i are one. The main effect of this change is that the LP relaxation of the ILP MATCHING$(I, \boldsymbol{c})$ can now be a strict relaxation. Indeed, the LP relaxation might allow for higher objective values because it allows fractional solutions.[6] As a result, our dual prices will no longer *exactly* compute the marginal value of a unit of capacity. In any case, to retain the exact connection to stochastic programming in Equation (1), PM would have to subtract the opportunity cost of s_i units of capacity from $u_{i,\ell}$, which might exceed s_i times the opportunity cost of a single unit of capacity.

However, as the capacity of most affiliates is much larger than the size of a typical case, both approximations can be expected to be relatively close, which is what we find empirically: we repeat the experiment of the previous section, but without splitting up cases into individual refugees. The results are nearly indistinguishable, which supports our decision to use LP relaxations even in the setting with inseparable cases. The full figures are deferred to Section EC.4.1 of the e-companion.

6. Batching

A second aspect that we have not considered thus far is that HIAS does not actually process arriving cases one by one, but in batches containing one or multiple cases. Most of these batches result from the weekly meetings between the resettlement agencies, but smaller batches with urgent cases are allocated between the weekly meetings.

The fact that cases arrive in batches does not make the problem harder; after all, a matching algorithm that does not support batching can still be used by presenting the cases of each batch to the algorithm one by one. As we will argue, however, batching represents an opportunity to improve on this strategy: there is a (limited) opportunity to increase total employment and a (substantial) opportunity to reduce running time.

Concerning total employment, using a nonbatching algorithm in a batching setting is wasteful because it ignores potentially valuable information. Specifically, when the earliest cases of the batch are allocated, a nonbatching algorithm presumes that the characteristics of the other cases in the batch are not yet known. Arguably, as the sizes of batches tend to be much smaller than the total number of cases n, the amount by which accounting for this information can increase total employment is likely to be limited.

As for running time, given that the matching algorithm receives no new information between the first and last case of a batch, it seems reasonable not to recompute potentials within a batch. As there tend to be 5 to 10 times more cases than batches and as the computation of potentials is the bottleneck in the running time of the potential algorithms, this promises to substantially speed up the algorithm.

In adapting our algorithm PM to batching, we will not change how we compute the potentials p_ℓ. However, the algorithm now allocates all cases in the batch at once, still with the objective of optimizing the immediate score

of the assignment less the sum of potentials consumed. Thus, our extended algorithm PMB ("potential match with batching," Algorithm 4 in Section EC.3 of the e-companion) allocates the current batch according to the solution to a matching ILP, in which matching case i to affiliate ℓ contributes an amount of $u_{i,\ell} - s_i p_\ell$ to the objective. Note that if all batches have size $b = 1$, this algorithm coincides with our previous algorithm PM. Moreover, PMB also generalizes the greedy algorithm previously implemented in Annie MOORE, which can be recovered by setting all potentials p_ℓ to zero.

We can now compare the running time of our algorithms to Bansak's (2020) algorithm 1, which obtained the highest employment in his study. Though this algorithm is closely related to ours, it does not use dual prices to compute opportunity costs and handles batching in a way that does not improve running time. The bottleneck in our algorithms and his is the computation of bipartite-matching linear programs over the trajectories of simulated future arrivals. Whereas we compute a single such program per batch of arrivals, Bansak (2020) solves $|L| \cdot b$ many such linear programs per batch, where $|L|$ is the number of affiliates and b is the number of cases in the batch. In our data set, a typical value of $|L| \cdot b$ is around 150, so these speedups are substantial.

6.1. Empirical Evaluation

We repeat the experiment measuring the total employment obtained by the algorithms, this time with the greedy algorithm and the potential algorithms allocating cases in batches. As shown in Figure 4, the results again look very close to those in the restricted setting of online bipartite matching, confirming that our algorithmic approach generalizes well not only to nonunit case sizes but also to batching as it is used in practice.

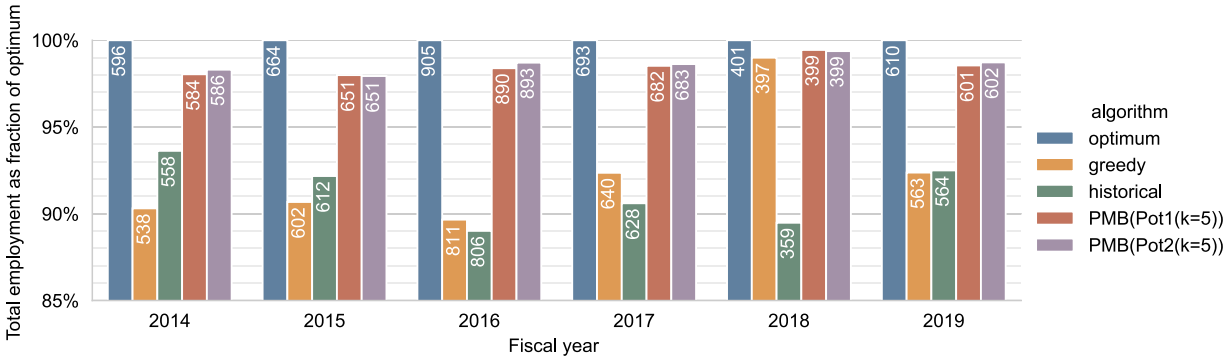
Because processing entire cases in batches is much faster than processing cases (or individual refugees) one by one, we are now in a position to run each potential algorithm many times and analyze the distribution of total employment. As shown in Figure 5, the total employment produced by each potential algorithm is sharply concentrated, especially when the algorithms use $k \geq 3$ trajectories to compute duals.

Running each algorithm many times enables us to compare the relative performance of the potential algorithms. Across both ways of computing potentials and all fiscal years (with the exception of 2018, where everything is very close together), we see a clear tendency that averaging the potentials across more trajectories improves the employment outcome. These effects are somewhat limited, though, as going from a single trajectory to nine trajectories improves the median employment by less than half a percent of the hindsight optimum. As is to be expected, increasing k exhibits diminishing returns.

For k held constant, we observe that the Pot2 variants quite consistently outperform the Pot1 variants, again with the exception of 2018, in which a small inversion of this trend can be seen. Although all potential algorithms perform very well, based on these results, we recommend the Pot2 potentials with a relatively large k for practical implementation. Of course, increasing k increases the running time of the matching algorithm. However, because a resettlement agency computes only one set of potentials per day, the algorithm runs in few seconds even for $k = 9$ (see Section EC.4.3 of the e-companion).

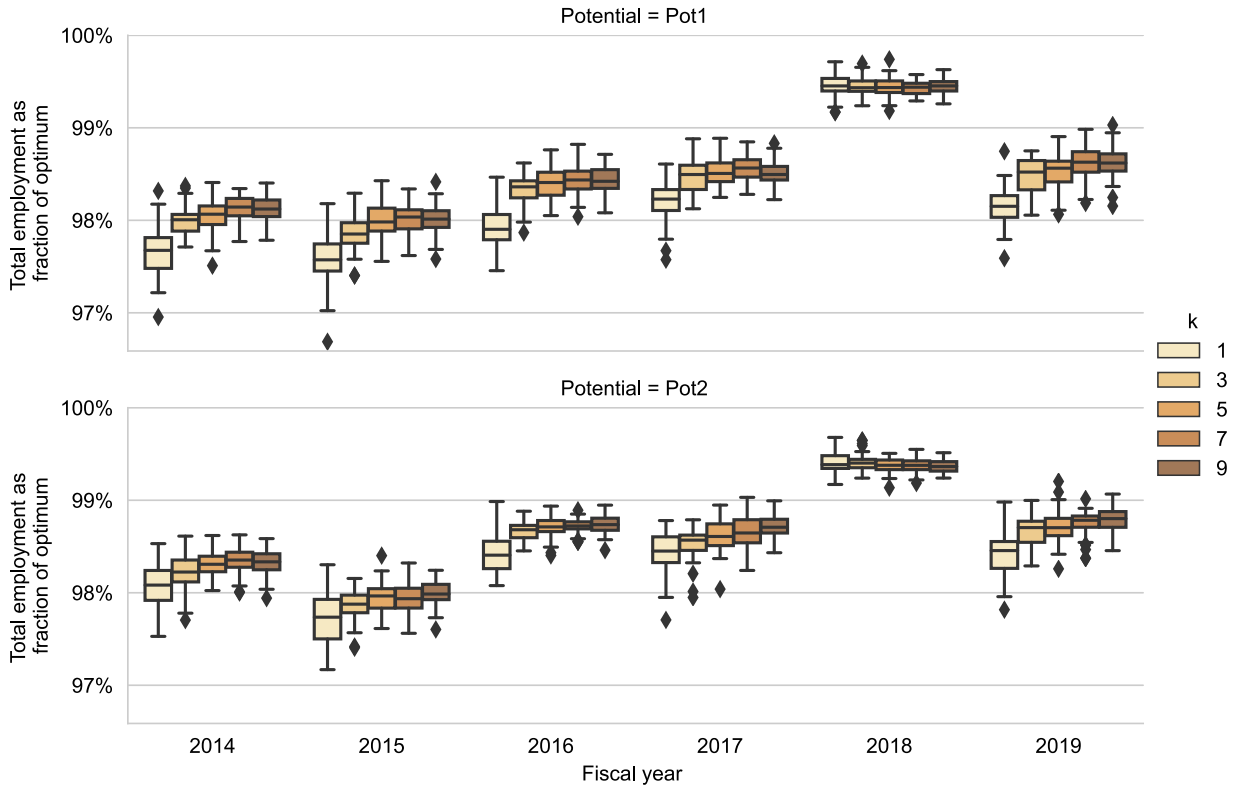
To additionally support our observation that the potential algorithms outperform the greedy algorithm and the historical matching, we repeat the experiment from Figure 4 for additional arrival sequences derived from the historical data. As we show in Section EC.4.6 of the e-companion, we obtain similar employment performance as in Figure 4 if the arrival sequence for each year is reversed, or if we consider shifted yearly arrival periods from, say, April to the March of the following year rather than fiscal years (from October to September). In Section 7.2, we also evaluate the algorithms on bootstrapped arrivals. While we discuss more specific observations there, the potential algorithms perform similarly

Figure 4. (Color online) Total Employment, Where Cases Are Not Split and Arrive in Batches



Notes. Capacities are the final fiscal year capacities. In contrast to Figure 1, cases are treated as inseparable, cases arrive in batches, and the batching variants of the greedy and potential algorithms are used. For the potential algorithms, the mean employment across 50 random runs is shown.

Figure 5. (Color online) Distribution of the Total Employment Obtained by Instantiating PMB with Different Potential Methods and Different k in the Experiment in Figure 4 (Whole Cases, Batching, Final Capacities) and Over 50 Random Runs per Algorithm



well or slightly better in that setting, consistently at 99% of the hindsight optimum.

7. Uncertainty in the Number of Future Arrivals

Given that our algorithm PMB supports nonunit sized cases and batching, it might seem that we are ready to replace the greedy algorithm in Annie MOORE by our potential algorithm. However, our algorithm crucially relies on one piece of input that the greedy algorithm did not need, namely, the total number of cases arriving in the fiscal year. This number determines the length of the sampled trajectories, which can greatly impact the shadow prices and, thus, how the algorithm allocates cases.

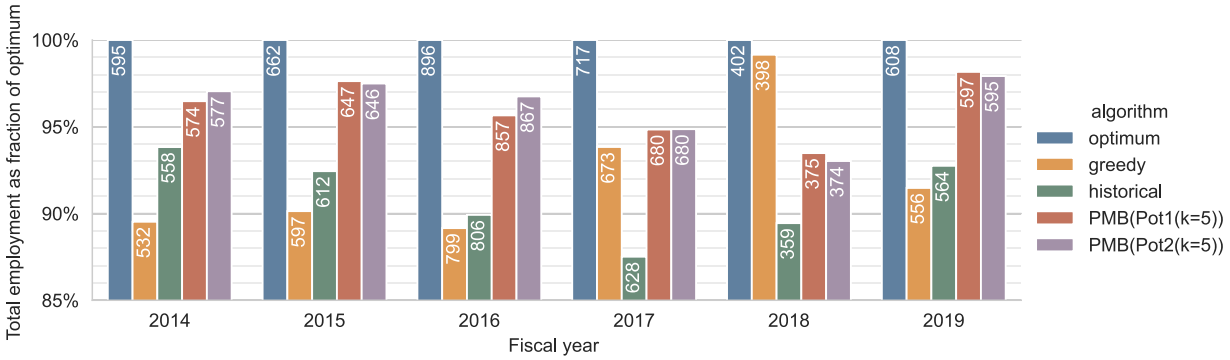
In principle, the information given to resettlement agencies should provide a fairly precise estimate of how many cases are expected to arrive. Indeed, before the start of each fiscal year, the U.S. Department of State announces how many refugees it intends to resettle in that fiscal year, and resettlement agencies are instructed to prepare for a certain fraction of this total number. In fact, HIAS sets its affiliate capacities to sum up to 110% of this number of announced refugees, which is intended to give local affiliates a good idea of how many refugees they will receive while affording the resettlement agency some freedom in its allocation decisions.

7.1. Relying on Capacities

It is thus natural to run our potential algorithms under the assumption that the number of arriving refugees will be $1/(110\%) \approx 91\%$ of the total announced capacity.[7] The result of this strategy is shown in Figure 6. Because these experiments use the initial, unrevised capacities, the employment scores of the hindsight optimum and the greedy algorithm may differ from those in previous experiments, which used the most revised capacities.[8] In all fiscal years other than 2017 and 2018, the imprecise knowledge of future arrivals deteriorates the approximation ratio of the potential algorithms, but the potential algorithms continue to clearly outperform the greedy baseline overall, and they outperform the historical matching in every single year.

Setting aside the outlier years of 2017 and 2018 for the moment, we investigate the fiscal years 2016 and 2019, in which arrivals were otherwise highest and lowest relative to the announced capacity. In fiscal year 2016, the total arrivals were particularly large relative to the initial capacity: the arrival numbers added up to 100% of the initial capacity rather than 91%, which means that our potential algorithms expected around 3,770 refugees to

Figure 6. (Color online) Total Employment, Where Cases Are Not Split Up and Arrive in Batches

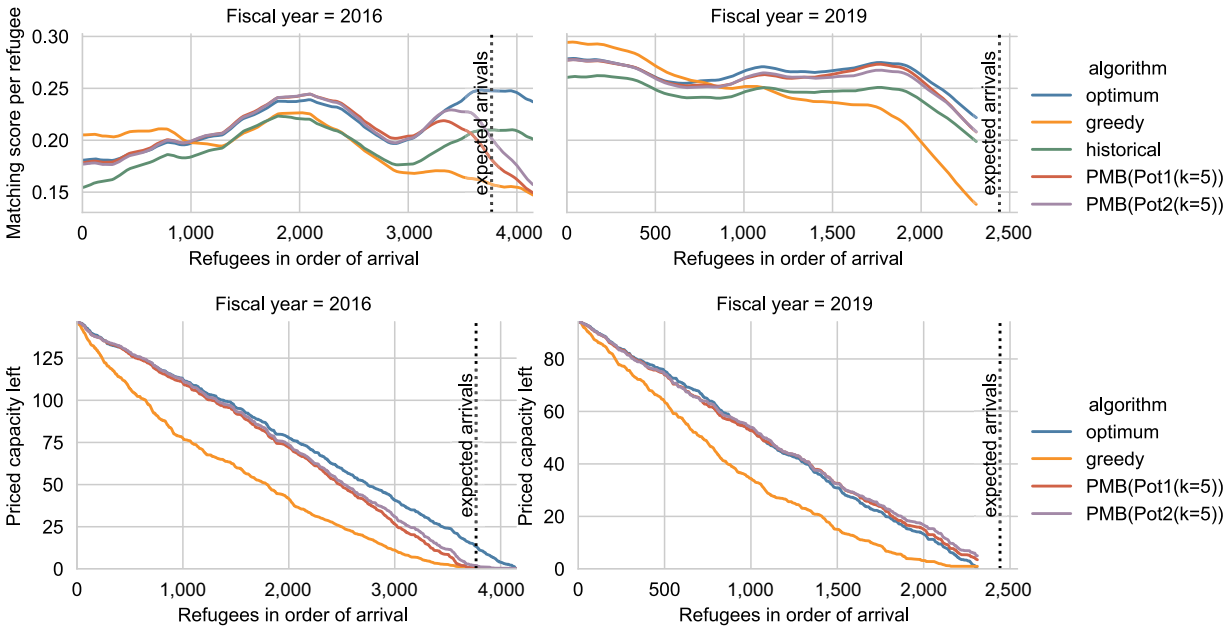


Notes. The potential algorithms no longer have access to the true number of arriving cases but assume that the arriving refugees amount to 91% of the total capacity. Capacities are the *initial* capacities of the fiscal year (except for historical). For the potential algorithms, the mean employment across 50 random runs is shown.

arrive rather than the 4,150 that ended up arriving. As a result, the potential algorithms consume the priced capacity at an approximately constant rate, consuming it all around the expected number of expected refugees (Figure 7, bottom left). Up to this point, the potential algorithms are more generous in consuming capacity than would be ideal given the actual number of arriving cases, which is why the potential algorithms obtain a slightly higher average employment over the first three-quarters of arrivals (Figure 7, top left) than the optimal matching in hindsight. For refugees arriving after the 3,770 expected refugees, however, the capacity in the best affiliates is used up, which is why the averaged employment sharply drops after this point.[9]

In 2019, by contrast, fewer refugees arrived than expected, only 86% of the total capacity. At the bottom right of Figure 7, it is visible that the potential algorithms consume priced capacity at a slightly lower rate than the optimal algorithm in hindsight, as they aim to use up the capacity around 2,440 refugees rather than the 2,310 who ended up arriving. This effect is reflected in the average employment rates (top right), which lie below that of the optimal algorithm throughout most of the year.[10]

Figure 7. (Color online) Evolution of the Per Refugee Match Score and Remaining Priced Capacity in Order of Arrival, for Fiscal Years 2016 and 2019 and One Run per Algorithm in the Experiment in Figure 6 (Whole Cases, Batches, Initial Capacities, Potential Algorithms Do Not Know n)



Notes. Dotted lines show how many refugees the potential algorithms expect. Smoothing is as in Figure 2. Priced capacity is not shown for "historical" because it uses different capacities.

The fiscal years of 2017 and 2018 stand out because the total number of arriving refugees fell far short of the announced number reflected in the approved capacities: in 2017, arrivals amounted to 65% of the approved capacities, whereas they amounted to only 46% in 2018. Both of these years fall into the beginning of the Trump administration, which not only sharply reduced the announced intake of resettled refugees, but furthermore abruptly halted the intake of refugees from six (predominantly Muslim) countries starting from early 2017.

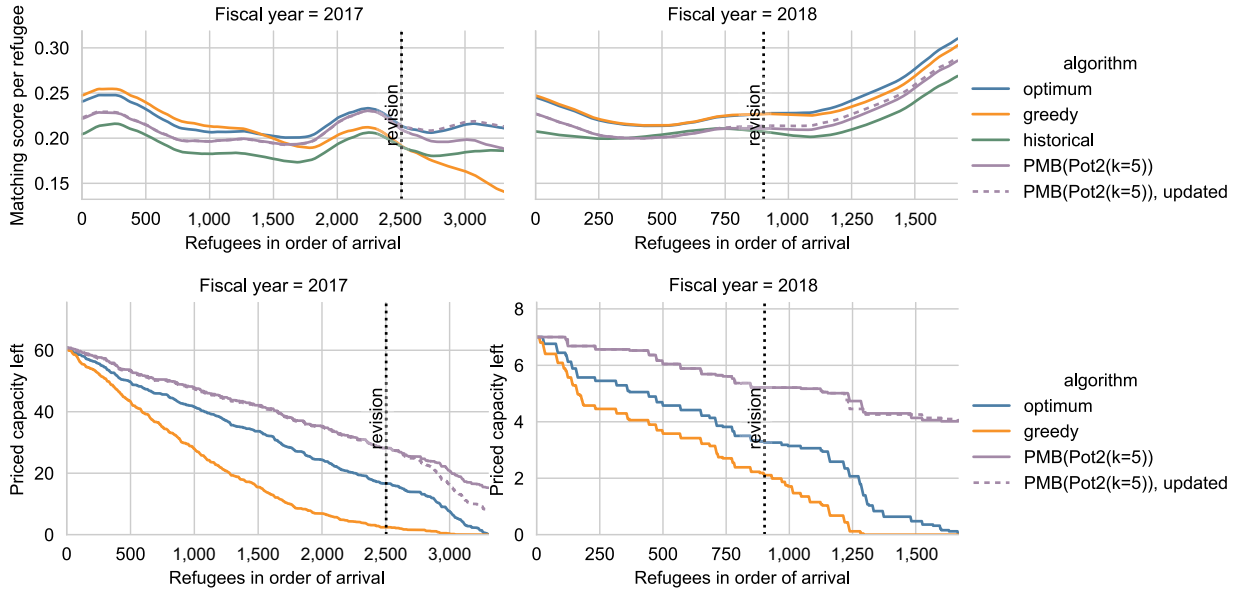
As the potential algorithm depicted in Figure 8 severely overestimates how many cases will arrive, it holds back much more priced capacity than would be optimal (bottom, solid lines). This causes the potential algorithms to extract less employment throughout the year than the optimal algorithm (top, solid lines). As observed in Section 4.2, the capacities in 2018 are so loose that the greedy algorithm performs close to optimal.

In these two years, the U.S. Department of State eventually reacted by correcting the expected arrivals downward and instructing the resettlement agencies to reduce their capacities. In fiscal year 2017, this revision came quite late and ended up underestimating the arrivals: where the arrivals amounted to only 65% of the initial capacities, they exceeded the revised total capacity at a level of 103%, rather than amounting to the 91% that was intended. Even if imperfect, this signal that arrivals are much lower than originally announced is still useful to the potential algorithms. Indeed, in Figure 8, the dashed curve corresponds to a potential algorithm that still starts out expecting 91% of the initial capacities to arrive, but expects only 91% of the revised capacities to arrive from the point on where they were announced (vertical line). Although this information comes late, the algorithm in fiscal year 2017 uses the new information to burn through the remaining priced capacity more aggressively (bottom left), which allows for higher employment among refugees arriving after the revision of arrival numbers (top left). As a result, the employment reaches 97% of the optimum in hindsight, exceeding the value of 95% without the updated information that we showed in Figure 6.

By contrast, the revision in fiscal year 2018 did not yield much useful information; whereas the arrivals amounted to 46% of the initial capacities, they still amounted to 48% of the revised capacities. This seems to indicate that even after half of the fiscal year's refugees had already been allocated, the administration overestimated the number of arriving refugees by a factor of two. Because the revision barely changed the number of expected arrivals, giving the potential algorithm access to this revised information does not have much effect (Figure 8, right).

Although we have considered the informational value of revisions above, our experiments have not considered that these revisions actually reduced the allowable capacities. Although we include a variant of the experiment in Section EC.4.7 of the e-companion, it is difficult to meaningfully compare the employment achieved by different algorithms if the parameters of the matching problem are changed so drastically during the matching period. One particular challenge is that, although the amount of reduction was extraneously decided, HIAS was involved

Figure 8. (Color online) Evolution of the Per Refugee Match Score and Remaining Priced Capacity in Order of Arrival, for Fiscal Years 2017 and 2018 in the Experiment in Figure 6 (Whole Cases, Batches, Initial Capacities, Potential Algorithms Do Not Know n)



Note. Dotted lines show evolution if potential the algorithm updates its expected arrival number at the time of capacity revision.

in deciding which capacities to decrease, which was done in a way that depended on previous allocation decisions.[11] Because we know only the revised capacities that were agreed upon, and not the counterfactual revision of capacities that would have been made, the greedy algorithm and the potential algorithms might have already exceeded a reduced capacity before it was announced. This means that the experiment rewards algorithms for greedily using up the capacity in the best affiliates before the revision, which we do not expect to be a good policy in practice. More generally, a substantial change in capacities is an exceptional situation, outside of our model, and cannot be addressed by our algorithm alone without manual intervention.

7.2. Arrival Misestimation on Bootstrapped Data and Incorporating Uncertainty

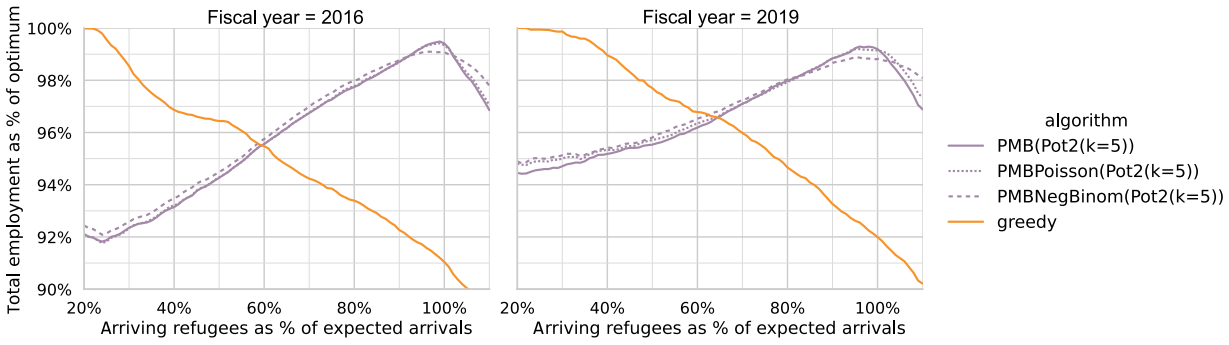
To obtain more systematic insights into the robustness of potential algorithms to misestimated arrival numbers, we study bootstrapped case arrivals, which allows us to simulate varying numbers of arrivals. The results of this experiment are displayed in Figure 9 (results for other fiscal years are deferred to Section EC.4.2 of the e-companion). As a baseline, consider the greedy algorithm, which obtains optimal employment when the number of arrivals is much lower than the total capacity (say, 25% of the expected arrivals, which is (25%/110%) ≈ 23% of the capacity), but becomes more and more suboptimal the more refugees arrive.

By contrast, the potential algorithms perform best (around 99% of the optimal employment) when the number of arriving refugees matches what the algorithm expects. On average, this number is around half of a percentage point higher than in the corresponding nonbootstrapped experiments (Figure 4). Such an increase is to be expected, as the bootstrapping setup ensures that the algorithm draws trajectories from the same distribution from which the arrivals are generated. In particular, the real arrival sequence used for Figure 4 might contain a drift in refugee characteristics or a seasonality not captured by our algorithm, and the lack of these features in the bootstrapped experiment allows for slightly higher employment. It is just as noticeable, however, that this increase is *only* half a percentage point, revealing that a drift of arrival characteristics and seasonality does not account for most of the remaining optimality gap of our algorithm.

The further the actual arrival number deviates from this expectation, the further the relative employment performance of the potential algorithm decreases. Noticeably, the performance more quickly deteriorates when the arrival numbers exceed the expectation, versus falling short. This sharp decline makes sense for two reasons. First, the algorithms aim to exploit all useful capacity exactly at the expected number of refugee arrivals; thus, only a subset of the affiliates remain available for subsequent arrivals. Second, once the number of arrivals exceeds the expectation, the trajectories in the potential algorithms add no cases beyond those that have already arrived, which means that the algorithm serves subsequent arrivals greedily. In the six fiscal years we observe, arrivals below the expectation seem like a more urgent problem than arrivals above the expectation, but overarrivals might well become a problem under different political circumstances or when applying potential algorithms to other matching settings.

A natural way to make the potential algorithms more robust to inaccurate arrival estimates is to treat arrival estimates not as exact predictions but as subject to some uncertainty. Concretely, we adapt the potential algorithms by sampling trajectories of different lengths, each drawn from a "prior" distribution whose mean is the arrival estimate, conditioning this distribution such that trajectory lengths are never less than the number of refugees who have already been allocated. Conceivably, these adapted trajectories could generate potentials that are robust across a wider range of arrival numbers, and the adapted algorithm could therefore lead to higher employment when the official arrival numbers are

Figure 9. (Color online) Employment Achieved by Different Algorithms as a Function of How Many Refugees Arrive



Notes. Refugee arrivals are bootstrapped over each fiscal year's historical arrivals, and the number of arriving refugees is given as a fraction of the historical arrivals. Capacities are 110% of historical matching. Employment is measured as a ratio of the optimal hindsight employment for the same set of arriving refugees. Curves are averaged over 10 arrival sequences.

inaccurate. The most obvious distribution is perhaps a Poisson distribution. As shown by the dotted line in Figure 9, using Poisson trajectories hardly changes the employment outcomes for any of the experiments relative to the baseline of fixed trajectory sizes. This is most likely due to the low variance of the Poisson distribution. For a quite typical mean of 3,000 arriving refugees, 95% of the probability mass lies within a distance of only 3.6% of the mean. For this reason, we also try a distribution with overdispersion, specifically, a negative binomial distribution parameterized to have its mean equal to the expected arrivals and its standard deviation equal to 10% of the expected arrivals. For example, if again 3,000 arrivals are expected, 95% of the probability mass deviates up to 20% from the mean. As the figure shows, negative-binomial trajectories lead to decent improvements in employment when more refugees arrive than expected. When fewer refugees arrive than expected, using random trajectory lengths helps more often than not, though with different degrees of success. Overall, negative-binomial arrivals seem to make the potential algorithms marginally more robust to misestimated arrival numbers, though not by enough to make misestimation less of an overall concern. However, this additional robustness comes at a non-negligible cost when arrival estimates are accurate.

7.3. Better Knowledge of Future Arrivals

In Section 7.1, we demonstrated that, even without outside supervision, our potential algorithms lead to substantial employment increases over the baselines, unless the announced capacities miss the eventual arrival numbers by an extreme margin. Even in these typical years, however, more accurate arrival predictions could increase the total employment on the order of percentage points of the hindsight optimum. Obviously, more accurate information about arrivals would be even more useful in years like 2017 and 2018, in which the official information is unreliable.

One approach would be to use time-series prediction to estimate the number of arrivals. For instance, when the U.S. Department of State revised the capacities for the fiscal year 2018 in January 2018 (several months into the fiscal year), the announcement that 2.5 times more refugees were still to come than had already arrived might have raised some doubts. However, the graph of monthly arrivals in Figure 10 shows that late increases in arrival rates may actually happen as they did in fiscal year 2016.[12]

A fundamental challenge that any data-driven approach faces is that there are very little data to learn from. Indeed, although HIAS has data on hundreds of thousands of refugees, they have data on only 15 fiscal years, which is, moreover, incomplete and smaller scale in earlier years. Thus, there is a limited foundation on which to learn about how arrival patterns change between years. This task becomes especially difficult given that arrival numbers are heavily influenced by external events such as elections, the emergence of humanitarian disasters, and changes in immigration policy, which cannot be deduced from past arrival patterns. Thus, although a time-series prediction approach might lead to marginal improvements over naïvely expecting 91% of the capacity to arrive, past arrival numbers are unlikely to give enough information to accurately predict future arrival numbers.

Fortunately, resettlement agencies such as HIAS already possess much richer information and insights into the dynamics of refugee arrivals than a pure data approach would consider. In fiscal year 2017, for example, HIAS foresaw a worsening climate for refugee resettlement immediately after the November 2016 election[13] and was aware of concrete plans to drastically reduce refugee intake in January 2017,[14] both before these changes were reflected in arrival numbers and before the capacities were officially updated in March 2017. Similarly, HIAS continuously monitors domestic politics and international crises for their potential impact on resettlement, and, moreover, it has some limited insight into the resettlement pipeline, which allows it to prepare for changes in arrivals. We therefore believe that, rather than building a sophisticated tool for predicting arrivals in a fully autonomous manner, it is preferable to allow HIAS staff to override our prediction with more advanced information.

Figure 10. (Color online) Monthly Number of Allocated Refugees, Disaggregated by Fiscal Year of Arrival

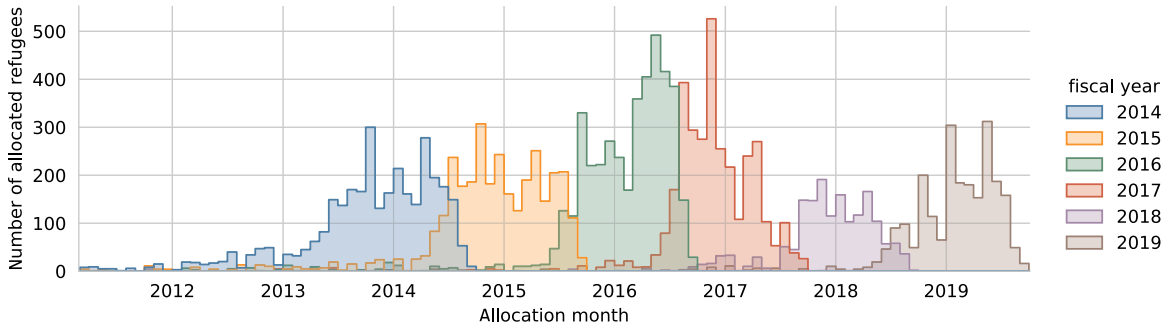
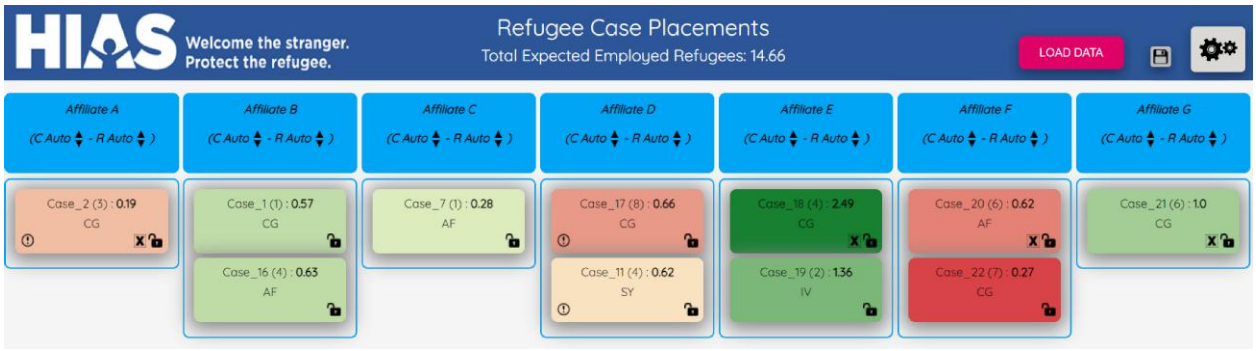


Figure 11. (Color online) Updated Annie Interface



Notes. Family tiles now show both the original numerical employment scores of families in affiliates and the *adjusted* employment scores by their shading. In the user interface, green tiles indicate positive adjusted scores, red tiles indicate negative scores, and darker shades represent greater magnitudes.

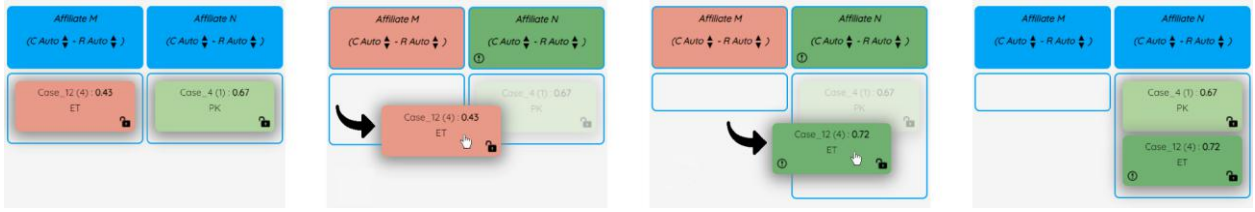
8. Implementation in Annie MOORE

To enable HIAS to benefit from online allocation via potentials, we have integrated new features into its matching software Annie MOORE, which constitute the software's second major release (Annie 2.0). A crucial design requirement is that HIAS staff must be able to override the allocation recommendations of Annie MOORE when they are aware of requirements outside of our model. From an interface-design perspective, the challenge is to visualize the effect of such overrides on total employment, enabling HIAS staff to make informed trade-offs. In the original, static model, this was easy enough: as the quality of a matching was just the total employment of the current batch, the interface labeled each case–locality match with its associated employment score, and staff could drag the case to other localities to see the respective employment scores. In a dynamic setting, however, presenting only the employment scores may unintentionally encourage HIAS staff to greedily use capacity in their overrides, at the expense of future arrivals.

As we illustrate in Figure 11, the new interface of Annie augments the original interface with information about affiliate potentials, thereby taking future arrivals into account. Specifically, the background color of the tile for case i encodes the *adjusted* employment score, that is, the original employment score $u_{i,\ell}$ less the potential $s_i p_\ell$ of the capacity consumed in affiliate ℓ.[15] The fact that the algorithm PMB always maximizes the sum of adjusted employment scores in its allocation of the current batch means that the algorithm is explainable in terms of the information presented to the user. In the interface, the green color spectrum indicates positive adjusted employment scores (meaning that the employment score of the case outweighs the loss in future employment), whereas the red color spectrum highlights negative adjusted scores (where a placement reduces future employment by more than its employment score). Darker shades signify greater magnitudes.

In overriding the allocation recommended by Annie MOORE, HIAS staff should be able to quickly find alternative placements for a case that do not reduce immediate and future employment by more than necessary. To support this workflow, our interface shows the adjusted employment scores of a case across all affiliates at a glance: As shown in Figure 12, upon dragging a particular case tile from its current placement, all other case tiles temporarily fade in appearance, and the shading of every affiliate tile temporarily assumes the adjusted employment score relative to the selected case. By hovering a selected case tile over a new affiliate, the original (numeric) employment score and the adjusted match score (background color of the case tile) dynamically update. Moreover, incompatibilities with affiliates due to nationality, language, family size, and single-parent households can be seen via an exclamation mark in the

Figure 12. (Color online) Moving a Family Tile



Notes. While moving a family tile, tiles belonging to other cases fade, and affiliate tiles are shaded as per their adjusted employment scores, in green (positive) or red (negative). Exclamation marks indicate incompatibilities.

lower left corner of the affiliate tile. After dropping the case tile in a new affiliate, the background color for each affiliate returns to its original blue shade, and all affiliate-tile exclamation marks disappear.

On a separate screen (not shown), Annie 2.0 enables the entry of a prediction for total refugee arrivals, as mentioned in Section 7.3. This estimate can be critical to inform the process of estimating proper shadow prices, as at times HIAS is in a better position to give more accurate case arrival predictions than officially announced capacities.

9. Conclusion

We have developed and implemented online algorithms allocating refugees in a way that promotes refugees' prospects of finding employment. Our algorithms outperform the greedy and historical baselines, even when taking into account how refugee placement in practice deviates stylized online matching problems.

Although we have tested the algorithms as an autonomous system, the success of Annie MOORE in increasing employment outcomes in practice will depend on how it performs in interaction with HIAS resettlement staff. In Section 7.3, we already saw that the allocation decisions of Annie can greatly profit from human decision makers providing better estimates of future arrivals. Human input is equally crucial in dealing with uncertainty in several other places; for example, HIAS staff might intervene by correcting a case's physical-arrival year if the Department of State's estimate seems off, or they might increase certain affiliate capacities late in the year if they anticipate that these capacities will be renegotiated. By allowing all parameters of the matching problem to be changed, Annie MOORE allows HIAS resettlement staff to improve the matching using any available information.

Our hope is that the human-in-the-loop system consisting of the matching algorithm and HIAS staff will combine the strengths of both of its parts: On the one hand, the algorithms in Annie MOORE capitalize on subtle patterns in employment data and manage capacity more effectively over the course of the fiscal year. On the other hand, the expert knowledge of HIAS staff enables the system to handle the uncertainty that is inherent in a matching problem involving the actions of multiple government agencies, dozens of affiliates, and thousands of refugees. In light of the administration's recent increase of the total resettlement capacity from 15,000 to 125,000,[16] we foresee both parts playing a crucial role: the increasing scale of the problem will make data-based algorithms more effective, and human guidance will be necessary to navigate the evolving environment of a rapidly growing operation.

Acknowledgments

The authors are grateful to HIAS for providing data and for sharing insights into the practical challenges of refugee resettlement. The authors thank Siddhartha Banerjee, Avrim Blum, Bailey Flanigan, and David Wajc for helpful discussions.

Endnotes

[1] Each fiscal year ranges from October 1 of the previous calendar year to September 30. For example, fiscal year 2017 ranges from October 1, 2016, to September 30, 2017.

[2] For example, allowing cases to be unmatched is necessary because an arriving case might only be compatible with affiliates whose capacity is already exhausted. When these situations occur in practice, such cases do not remain unmatched; instead, capacities can be increased or case–affiliate incompatibilities overruled manually by the arrivals officer. For our sequence of models, we report the fraction of matched refugees in Section EC.4.8 of the e-companion and find that our algorithms do not lead to fewer refugees being matched than in the greedy baseline. To lower the number of unmatched refugees at the cost of reducing employment, one can add a constant reward per refugee to the $u_{i,\ell}$ with $\ell \neq \perp$.

[3] From Section 6 onward, cases will instead arrive in batches, which can be allocated simultaneously.

[4] When the number of refugees resettled in the fiscal year exceeds the official capacity, we use the number of resettled refugees instead. In these situations, HIAS negotiated an increase in capacity that may not be recorded in our data.

[5] In these cases, we estimate the employment achieved by the case using the regression rather than using $u_{i,\ell} = -\infty$.

[6] One can always find a fractional solution that splits cases into $1/s_i$ fractions similarly to what we did in the evaluation of Section 4.2.

[7] To convert the number of remaining refugees into a number of cases, we divide by the average case size of recent arrivals (over the years, this average size fluctuates between 2.4 and 2.6). Although the number of refugees who have arrived is below 91% of the total capacity, this gives us a total number of cases n for the algorithms. Once the number of arrivals exceeds 91% of the total capacity, we make the algorithms assume that the current case is the last to arrive, that is, all subsequently sampled trajectories have length zero.

[8] This means that the comparison with the historical algorithm is not quite on equal terms, because the latter is constrained by a different set of capacities. In all fiscal years except for 2017 and 2018, the final capacities are affiliate-wise larger than the original capacities.

[9] Note that, because of the triangle smoothing, the drop starts dragging down the curve 500 arrivals before its actual start.

[10] The drop in employment probabilities at the end of the fiscal year affects all algorithms including the hindsight optimum and must therefore be caused by an anomaly in arrival characteristics.

[11] Although the sum of capacities did not change much in fiscal year 2018, the capacities of some affiliates were substantially decreased, and those of others were substantially increased.

[12] In fiscal year 2016, the number of arrivals after January 2016 was 1.6 times larger than the number that had arrived so far. In the fiscal year of 2015, the number of refugees arriving after January 2015 was only 75% of that arriving before.

[13] See https://www.hias.org/news/press-releases/hias-calls-president-elect-trump-respect-longstanding-refugee-policy.

[14] See https://www.hias.org/news/press-releases/trumps-planned-action-refugees-betrayal-american-values.

[15] The employment scores of cases in affiliates are prominently retained in text labels.

[16] See https://www.hias.org/news/press-releases/refugee-cap-fy2022-set-125000.

References

Ager A, Strang A (2008) Understanding integration: A conceptual framework. *J. Refugee Stud.* 21(2):166–191.

Agrawal S, Wang Z, Ye Y (2014) A dynamic near-optimal algorithm for online linear programming. *Oper. Res.* 62(4):876–890.

Ahani N, Andersson T, Martinello A, Teytelboym A, Trapp AC (2021) Placement optimization in refugee resettlement. *Oper. Res.* 69(5):1468–1486.

Alaei S, Hajiaghayi M, Liaghat V (2012) Online prophet-inequality matching with applications to ad allocation. Faltings B, Leyton-Brown K, Ipeirotis P, eds. *Proc. 13th ACM Conf. Electronic Commerce* (Association for Computing Machinery, New York), 18–35.

Alaei S, Hajiaghayi M, Liaghat V (2013) The online stochastic generalized assignment problem. Raghavendra P, Raskhodnikova S, Jansen K, Rolim JDP, eds. *Proc. 16th Internat. Workshop Randomization Approximation Techniques Comput. Sci.* (Springer, Berlin), 11–25.

Asadpour A, Wang X, Zhang J (2020) Online resource allocation with limited flexibility. *Management Sci.* 66(2):642–666.

Åslund O, Fredriksson P (2009) Peer effects in welfare dependence: Quasi-experimental evidence. *J. Human Resources* 44(3):798–825.

Åslund O, Rooth DO (2007) Do when and where matter? Initial labour market conditions and immigrant earnings. *Econom. J.* 117(518):422–448.

Åslund O, Östh J, Zenou Y (2010) How important is access to jobs? Old question, improved answer. *J. Econom. Geography* 10(3):389–422.

Åslund O, Edin PA, Fredriksson P, Grönqvist H (2011) Peers, neighborhoods, and immigrant student achievement: Evidence from a placement policy. *Amer. Econom. J.: Appl. Econom.* 3(2):67–95.

Banerjee S, Gurvich I, Vera A (2020) Constant regret in online allocation: On the sufficiency of a single historical trace. Working paper, Cornell University, Ithaca, NY.

Bansak K (2020) A minimum-risk dynamic assignment mechanism along with an approximation, heuristics, and extension from single to batch assignments. Preprint, submitted July 2, https://arxiv.org/abs/2007.03069v2.

Bansak K, Paulson E (2022) Outcome-driven dynamic refugee assignment with allocation balancing. Preprint, submitted January 13, https://arxiv.org/abs/2007.03069.

Bansak K, Ferwerda J, Hainmueller J, Dillon A, Hangartner D, Lawrence D, Weinstein J (2018) Improving refugee integration through data-driven algorithmic assignment. *Science* 359(6373):325–329.

Damm AP (2014) Neighborhood quality and labor market outcomes: Evidence from quasi-random neighborhood assignment of immigrants. *J. Urban Econom.* 79:139–166.

Devanur NR, Hayes TP (2009) The AdWords problem: Online keyword matching with budgeted bidders under random permutations. Parkes DC, Dellarocas C, Tennenholtz M, eds. *Proc. 10th ACM Conf. Electronic Commerce* (Association for Computing Machinery, New York), 71–78.

Fahrbach M, Huang Z, Tao R, Zadimoghaddam M (2020) Edge-weighted online bipartite matching. Irani S, ed. *Proc. 61st Annual IEEE Sympos. Foundations Computer Sci.* (Institute of Electrical and Electronics Engineers, Piscataway, NJ), 412–423.

Freund D, Banerjee S (2019) Good prophets know when the end is near. Preprint, submitted November 1, http://dx.doi.org/ 10.2139/ssrn.3479189.

Freund D, Lykouris T, Paulson E, Sturt B, Weng W (2023) Group fairness in dynamic refugee assignment. *EC '23: Proc. 24th ACM Conf. Econom. Comput.* (ACM, New York), 701.

Ho CJ, Vaughan J (2012) Online task assignment in crowdsourcing markets. Hoffmann J, Selman B, eds. *Proc. 26th AAAI Conf. Artificial Intelligence*, vol. 26 (Association for the Advancement of Artificial Intelligence, Washington, DC), 45–51.

Johari R, Kamble V, Kanoria Y (2021) Matching while learning. *Oper. Res.* 69(2):655–681.

Jones I (2015) Home away from home. *State Magazine* 606:17–20.

Karp RM, Vazirani UV, Vazirani VV (1990) An optimal algorithm for on-line bipartite matching. Ortiz H, ed. *Proc. 22nd Annual ACM Sympos. Theory Comput.* (Association for Computing Machinery, New York), 352–358.

Kesselheim T, Radke K, Tönnis A, Vöcking B (2013) An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. Bodlaender HL, Italiano GF, eds. *Proc. 21st Annual Eur. Sympos. Algorithms* (Springer, Berlin), 589–600.

Leonard HB (1983) Elicitation of honest preferences for the assignment of individuals to positions. *J. Political Econom.* 91(3):461–479.

Manshadi VH, Gharan SO, Saberi A (2012) Online stochastic matching: Online actions based on offline statistics. *Math. Oper. Res.* 37(4):559–573.

Martén L, Hainmueller J, Hangartner D (2019) Ethnic networks can foster the economic integration of refugees. *Proc. Natl. Acad. Sci. USA* 116(33):16280–16285.

Sinclair SR, Frujeri FV, Cheng C-A, Marshall L, Oliveira Barbalho HD, Li J, Neville J, Menache I, Swaminathan A (2023) Hindsight learning for MDPs with exogenous inputs. Krause A, Brunskill E, Cho K, Engelhardt B, Sabato S, Scarlett J, eds. *International Conference on Machine Learning* (PMLR, New York), 31877–31914.

Talluri KT, Van Ryzin G (2004) *The Theory and Practice of Revenue Management*, vol. 1 (Springer, New York).

United Nations High Commissioner for Refugees (2023a) Global Report 2022. Accessed September 7, 2023, https://reporting.unhcr.org/global-report-2022.

United Nations High Commissioner for Refugees (2023b) UNHCR projected global resettlement needs 2024. https://reporting.unhcr.org/unhcr-projected-global-resettlement-needs-2024.

Vazirani U, Vazirani V, Mehta A, Saberi A (2005) AdWords and generalized on-line matching. *Proc. 46th Annual IEEE Sympos. Foundations Computer Sci.* (Institute of Electrical and Electronics Engineers, Piscataway, NJ), 264–273.

Vee E, Vassilvitskii S, Shanmugasundaram J (2010) Optimal online assignment with forecasts. *Proc. 11th ACM Conf. Electronic Commerce* (Association for Computing Machinery, New York), 109–118.

Vera A, Banerjee S (2021) The Bayesian prophet: A low-regret framework for online decision making. *Management Sci.* 67(3):1368–1391.

Vera A, Banerjee S, Gurvich I (2021) Online allocation and pricing: Constant regret via Bellman inequalities. *Oper. Res.* 69(3):821–840.

Narges Ahani is an operations research analyst at the Bank of America (BoA). She is a member of the enterprise operation research and decision science team at BoA and works on pricing optimization problems. She received her PhD in data science from Worcester Polytechnic Institute and focused on the use of analytics for refugee resettlement. She codeveloped the first refugee resettlement decision support software, Annie^{TM} MOORE.

Paul Gölz is a postdoctoral fellow at the Simons Laufer Mathematical Sciences Institute. In Summer 2024, he will join Cornell as an assistant professor in operations research and information engineering. He studies democratic decision making and the fair allocation of resources, using tools from algorithms, optimization, and artificial intelligence. Outside of refugee resettlement, algorithms developed in his work are deployed to select citizens' assemblies around the world.

Ariel D. Procaccia is the Gordon McKay Professor of Computer Science at Harvard University. He works on a broad and dynamic set of problems related to artificial intelligence, algorithms, economics, and society. He has helped create systems and platforms that are widely used to solve everyday fair division problems, resettle refugees, mitigate bias in peer review, and select citizens' assemblies.

Alexander Teytelboym is a professor of economics at the University of Oxford. He works mainly on market design and the economics of networks. He cofounded Refugees.AI, a research network interested in creating systems that use tools from machine learning, optimization, and matching theory to find the best matches between refugees and local communities.

Andrew C. Trapp is an associate professor of operations and industrial engineering at Worcester Polytechnic Institute, with joint appointments in mathematical sciences and data science. He researches the use of prescriptive and predictive analytics, together with algorithms, to effectively allocate scarce resources for systems that serve vulnerable people.