Title
Outcome-Driven Dynamic Refugee Assignment with Allocation Balancing

Journal
Operations Research, 72(6)

ISSN
0030-364X

Authors
Bansak, Kirk
Paulson, Elisabeth

Publication Date
2024-11-01

DOI
10.1287/opre.2022.0445

Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Outcome-Driven Dynamic Refugee Assignment with Allocation Balancing

Kirk Bansak, Elisabeth Paulson

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

OPERATIONS RESEARCH
Vol. 72, No. 6, November–December 2024, pp. 2375–2390
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Crosscutting Areas

Outcome-Driven Dynamic Refugee Assignment with Allocation Balancing

Kirk Bansak,[a] Elisabeth Paulson[b,*]

[a] Department of Political Science, University of California, Berkeley, California 94720; [b] Technology and Operations Management Unit, Harvard Business School, Boston, Massachusetts 02163

*Corresponding author

Contact: kbansak@berkeley.edu, https://orcid.org/0000-0003-0229-9694 (KB); epaulson@hbs.edu, https://orcid.org/0000-0002-8318-0937 (EP)

Received: August 28, 2022
Revised: May 7, 2023; November 23, 2023
Accepted: February 13, 2024
Published Online in Articles in Advance: March 25, 2024

Area of Review: OR Practice

https://doi.org/10.1287/opre.2022.0445

Abstract. This study proposes two new dynamic assignment algorithms to match refugees and asylum seekers to geographic localities within a host country. The first, currently implemented in a multiyear randomized control trial in Switzerland, seeks to maximize the average predicted employment level (or any measured outcome of interest) of refugees through a minimum-discord online assignment algorithm. The performance of this algorithm is tested on real refugee resettlement data from both the United States and Switzerland, where we find that it is able to achieve near-optimal expected employment, compared with the hindsight-optimal solution, and is able to improve upon the status quo procedure by 40%–50%. However, pure outcome maximization can result in a periodically imbalanced allocation to the localities over time, leading to implementation difficulties and an undesirable workflow for resettlement resources and agents. To address these problems, the second algorithm balances the goal of improving refugee outcomes with the desire for an even allocation over time. We find that this algorithm can achieve near-perfect balance over time with only a small loss in expected employment compared with the employment-maximizing algorithm. In addition, the allocation balancing algorithm offers a number of ancillary benefits compared with pure outcome maximization, including robustness to unknown arrival flows and greater exploration.

Funding: Financial support from the Charles Koch Foundation, Stanford Impact Labs, the Rockefeller Foundation, Google.org, Schmidt Futures, the Stanford Institute for Human-Centered Artificial Intelligence, and Stanford University is gratefully acknowledged.
Supplemental Material: The online appendix is available at https://doi.org/10.1287/opre.2022.0445.

Keywords: dynamic assignment algorithms • stochastic programming • load balancing • refugee matching • machine learning

1. Introduction

Host countries have, in recent years, been faced with increasing flows of refugees and asylum seekers. Currently, the United Nations Refugee Agency estimates that there are over 35 million refugees worldwide (United Nations 2023). In most countries that accept refugees and/or asylum seekers, refugees and asylum seekers are assigned and relocated across various localities by migration authorities. The capacities or target distributions of refugees across the localities are determined by authorities on a yearly or other regular basis.

The goal of host countries is to help these new arrivals achieve economic self-sufficiency and other positive integration outcomes. Accordingly, a number of countries have begun to explore and implement outcome-based geographic matching in their refugee resettlement and/or asylum programs. Therefore, recent research studies the problem of efficiently assigning refugees to localities in order to maximize outcomes such as employment (Bansak et al. 2018, Ahani et al. 2021). This research falls within a broader area of policy interest as national resettlement programs seek new approaches to help ever-increasing flows of refugees and asylum seekers to better integrate (e.g., find employment) in their host countries (e.g., Andersson et al. 2018, Mousa 2018, Gölz and Procaccia 2019, Olberg and Seuken 2019, Acharya et al. 2022, Ahani et al. 2023).

Outcome-based matching was introduced in the context of refugee and asylum-seeker assignment by Bansak et al. (2018), with the goal of leveraging administrative data to improve key refugee outcomes (e.g., employment in the host country) by optimizing refugees' geographic assignment within a country. To do so, machine learning methods are used to predict refugees' expected outcomes in each possible landing location as a function of the refugees' personal characteristics. Those expected outcomes are then used as inputs into constrained matching procedures to determine a location recommendation for each refugee.

A greedy approach to the refugee assignment problem—one that assigns each refugee to the location (among those that are available) with the highest predicted outcome—is suboptimal when the resettlement locations have capacity constraints. This is the case in practice, where each location only has a certain number of slots in a given time period. For the United States, the time period is one year, but this can vary across host countries. Therefore, Bansak et al. (2018) and other previous studies on outcome-based refugee matching (e.g., Gölz and Procaccia 2019, Ahani et al. 2021) have proposed optimal matching approaches to the refugee assignment problem that take into account these capacity constraints.

This paper, along with the concurrent work of Ahani et al. (2023), are the first two papers to consider the *dynamic* aspect of the outcome maximization matching problem. In many countries—including the United States, Switzerland, Sweden, Netherlands, and Norway—refugees and asylum seekers must be assigned to a locality virtually immediately upon being processed by resettlement authorities. As a result, each arriving refugee or asylum-seeker case (an individual or family) is typically assigned in an online fashion, and these assignments cannot be reversed. The dynamic aspect of this problem introduces a key trade-off between immediate and future rewards: assigning a current case to a location results in an immediate reward (namely, the employment score of the current case at that location), but also uses up a slot at that location for future arrivals.

This paper introduces two new dynamic matching algorithms. The first is a "minimum-discord" algorithm that seeks to maximize expected employment (or any alternative outcome of interest) and is currently employed in a pilot implementation in Switzerland, undertaken by the Swiss State Secretariat of Migration in collaboration with academic researchers. Details on the implementation in Switzerland are provided in Section 5. The minimum-discord algorithm achieves near-optimal employment compared with the hindsight-optimal solution on real-world U.S. and Swiss data. However, it can result in an imbalanced allocation to the localities over time, which leads to implementation difficulties and an imbalanced workload for the resettlement offices.

The second algorithm proposed in this paper is an extension that integrates principles of load balancing into the objective. Because each locality has a given amount of resources (e.g., resettlement officers and service providers) that cannot be transferred across localities, maintaining a steady workload is a first-order concern of resettlement agencies. Hence, building on the minimum-discord outcome maximization algorithm and borrowing ideas from queuing theory, the second algorithm incorporates wait time minimization into the assignment process. This allows refugees to be dynamically assigned to localities in a way that improves their expected employment scores, while also maintaining a balanced allocation across the localities over time. Furthermore, the allocation balancing algorithm also offers ancillary benefits. In particular, it naturally handles the real-world scenario in which the total number of arrivals in a given period is not known in advance and helps to improve the resilience of the underlying learning system through greater exploration.

This paper uses data from both the U.S. and Swiss contexts to demonstrate the expected performance of the proposed approaches. Furthermore, we discuss real-world constraints, phenomena, and difficulties that arose during Swiss implementation and our proposed solutions.

1.1. Contributions

1. **Minimum-discord outcome-maximizing dynamic assignment algorithm.** We propose a "minimum-discord" online algorithm that assigns arriving refugees to locations within a host country. The goal of the algorithm is to maximize the sum of individual outcomes along a horizon, while obeying the capacity constraints of each location. This is accomplished through a Monte Carlo-sampling-based method that seeks to minimize the probability of choosing the "wrong" assignment in each time period compared with an offline benchmark. The proposed algorithm is a special case of the Bayes Selector algorithm (Vera and Banerjee 2020).

2. **Allocation balancing dynamic assignment algorithm.** We demonstrate that an outcome-maximizing assignment (even a hypothetical implementation of the hindsight-optimal solution) can result in severe imbalance across the localities over time because of clustered arrivals of refugees with similar characteristics. Thus, we develop a second online algorithm that explicitly balances the trade-off between outcomes and having a balanced allocation to the localities over time using a single parameter, γ, that controls the weight placed on allocation balancing versus outcome maximization.

3. **Results on real refugee resettlement and asylum-seeker data.** The results of the proposed methods are tested on real asylum-seeker data from Switzerland and refugee-resettlement data from one of the largest resettlement agencies in the United States. In both cases, the proposed algorithms are able to improve upon the status quo assignment procedures by roughly 40%–50% and achieve 95%–98% of the hindsight-optimal solution. Using the allocation balancing algorithm, we demonstrate the trade-off between total employment and having a balanced allocation over time as γ varies. In both contexts, we find that near-perfect balance over time can be achieved with little loss in employment.

4. **Implementation details.** We describe practical constraints and learnings that arose during implementation in Switzerland. For example, we discuss how capacity updating throughout the year (resulting from uncertainty about the total number of individuals that will arrive each year) and a requirement to balance the geographic distribution of certain nationalities are treated in practice.

1.2. Related Literature

This paper is related to the existing literature on refugee assignment, online stochastic bipartite matching, and matching with queues. In what follows, we provide an overview of the most relevant literature from each stream.

1.2.1. Geographic Assignment of Refugees and Asylum Seekers. Prior research has proposed different schemes for refugee matching both across and within countries based on refugee and/or host location preferences (Moraga and Rapoport 2014, Fernández-Huertas Moraga and Rapoport 2015, Andersson and Ehlers 2020, Nguyen et al. 2021, Delacrétaz et al. 2023). However, the lack of systematic data on preferences has thus far been a barrier to implementing these preference-based schemes.

In contrast, outcome-based matching was introduced in the context of refugee and asylum-seeker assignment by Bansak et al. (2018), with the goal of leveraging already-existing data to improve key refugee outcomes (e.g., employment in the host country). However, the dynamic aspect of the problem is not considered by Bansak et al. (2018), nor by most previous studies on outcome-based refugee matching (Gölz and Procaccia 2019, Ahani et al. 2021, Acharya et al. 2022). Although Andersson et al. (2018) consider dynamically matching asylum seekers to localities, they focus on the goals of Pareto efficiency and envy-freeness across localities, as opposed to outcome maximization.

Ahani et al. (2023) is the closest to this paper. Like this paper, Ahani et al. (2023) propose a dynamic matching algorithm to assign arriving refugees to locations within host countries with the goal of outcome maximization. The *potentials method* proposed in Ahani et al. (2023) is currently implemented by a resettlement agency in the United States. For each newly arriving household, both the algorithm proposed in this paper and that of Ahani et al. (2023) use a sampling procedure to solve many instances of the offline matching problem for the remaining horizon. Ahani et al. (2023) then propose using dual variables from the offline problems to inform the assignment of the current arrival—a method referred to as the *potentials* method. The algorithm proposed in this paper, on the other hand, assigns the current arrival to the location that minimizes the probability of a disagreement between the online algorithm and an offline benchmark. Both methods perform similarly on the data used in this paper. Our "minimum-discord" method, however, both is easily explainable and extends naturally to include allocation balancing, which is the focus of this paper.

Recent work also considers the relationship between the prediction and matching stages of dynamic refugee assignment (Bansak et al. 2023, Kasy and Teytelboym 2023) and group-fairness concerns (Freund et al. 2023).

1.2.2. Stochastic Online Bipartite Matching. Refugee matching is a special case of stochastic online bipartite matching, which has been a focus of operations and computer science researchers since the seminal work of Karp et al. (1990).

Two key features differentiate the refugee-matching setting from the classic online matching problem. First, it is a weighted matching problem. Second, there is effectively an infinite number of arrival "types," because of the large number of underlying covariates used to predict the outcome weights. Although weighted online matching problems are well-studied, most existing methods rely on an assumption of finite types (Devanur and Hayes 2009, Vee et al. 2010, Jaillet and Lu 2012, Bumpensanti and Wang 2020). Although, in theory, the covariate domain could be discretized and adapted to a finite-type setting, this is undesirable. Although there is prior research on distribution-free resource allocation problems, the performance guarantees of these algorithms nonetheless rely on a stationarity assumption (Devanur et al. 2019), which would not hold in practice in our setting. Rather, we seek to develop explainable methods that perform well and do not focus on theoretical performance guarantees. The proposed method bears similarities to recent work by Vera and Banerjee (2020).

Vera and Banerjee (2020) introduce a new framework for designing online policies, given access to an offline benchmark. This framework is used to develop a meta-algorithm ("Bayes Selector") for implementing low-regret online decisions across a broad class of allocation problems, including the assignment problem. In each state, the Bayes Selector chooses an action at each time interval that minimizes the likelihood of disagreement with an offline benchmark.

When the number of arrival types is finite, Vera and Banerjee (2020) show that the Bayes Selector algorithm achieves constant regret for many special cases of the online assignment problem. This result is also proven in Arlotto and Gurvich (2019) for the multisecretary problem. In this paper, we propose an outcome maximization algorithm that can be thought of as a special case of a Bayes Selector with infinite arrival types. When arrival types are drawn from a continuous distribution, Bray (2019) shows that the multisecretary problem—which is a special case of the refugee matching problem with only two locations—no longer has bounded regret. Additionally, Freund and Banerjee (2019) extend the methods introduced in Vera and Banerjee (2020) to more general decision-making problems, in particular, showing that the uniform regret bound does not hold in settings with large uncertainty about the time horizon, which is likely to be the case in the refugee matching context.

1.2.3. Allocation Balancing. This paper develops an online matching algorithm that not only improves outcomes for refugees, but also balances the allocation to receiving locations (or, more generally, assignment options) over time. This aspect of the paper is related to one-sided matching with queues. In our setting, each location can be thought of as having a dedicated queue because location assignments are made immediately and cannot be changed.

A subset of online bipartite matching literature considers queuing systems. The topology of the queuing system is critical to the analysis method, and most research in this area either focuses on optimally designing the underlying topology or has topology that is substantially different from the refugee matching context (e.g., Vera et al. 2020, Afeche et al. 2022, Leshno 2022).

Balseiro et al. (2021) propose an algorithm for online resource allocation that combines a welfare-maximizing objective with an arbitrary regularizer on the total consumption of each resource. This regularizer term can model what they call "load balancing"—ensuring that the total level of consumption of each resource is balanced at the end of the horizon. Although this has a similar flavor to our problem, we are interested in maintaining evenness in the allocation *throughout* the horizon.

The kidney-exchange literature also considers queueing models. For example, Ünver (2010) develops an online mechanism for allocating kidneys with the goal of reducing wait time. Bertsimas et al. (2013) develop online kidney-allocation policies that balance efficiency, fairness, and wait times. Recent work by Ding et al. (2018) also considers trade-offs between efficiency and fairness. However, unlike in our setting, the kidney-exchange problem has a single queue.

Because of the structure of the refugee matching problem (namely, the fact that each location has its own queue, and decisions are irrevocable), the allocation balancing problem bears similarity to load balancing in computer science (Azar 1998). However, the utility of load-balancing algorithms is limited in our setting because of our additional goal of outcome maximization. Thus, in this paper, we develop a new approach that combines the objective of maximizing employment outcomes with achieving a balanced allocation over time.

The remainder of the paper is organized as follows. Section 2 provides background on the refugee resettlement processes in the United States and Switzerland and more details on the data sets used in this study. Section 3 defines notation and describes the assumptions of the model and dynamics. Section 4 formulates the offline outcome maximization assignment problem, proposes an algorithm for the online setting, and demonstrates the performance of the method using the U.S. and Swiss data. Section 5 provides further details on the implementation in Switzerland, including practical constraints and challenges. Section 6 introduces the allocation balancing component of the problem and proposes a new heuristic that balances employment outcomes and wait time. Section 7 concludes.

2. Settings and Data

This section provides more detail on the two specific contexts from which data are used in this paper: the refugee resettlement process in the United States and the asylum procedure in Switzerland. The proposed methods are also applicable to many other countries where refugees and asylum seekers must be dynamically assigned to localities, including Sweden, Netherlands, and Norway.

2.1. Settings and Dynamics

In the U.S. context, we focus on the resettlement of United Nations High Commissioner for Refugees (UNHCR) refugees, who are granted refugee status in the United States prior to their arrival. In the United States, the target number of refugees that will be resettled each year is determined by an annual cap set in advance of the start of the year. Refugees who are accepted into the United States are then distributed across ten nongovernmental resettlement agencies. Finally, each of those agencies maintains its own network of localities to which they assign newly arrived refugees, with capacities for each locality also determined in advance.

In the Swiss context, we focus on asylum seekers, who request admission and asylum at a port of entry after entering a host country. In Switzerland, asylum seekers whose claims are not rejected are assigned on a case-by-case basis by the Swiss State Secretariat for Migration (SEM) to one of the 26 Swiss cantons. The assignment of asylum seekers across the cantons must follow an annually mandated proportionality key, which dictates the cantons' relative capacities to receive asylum seekers as a function of their population sizes.

In both the U.S. and Swiss contexts, the geographic placement for some refugees/asylum seekers is predetermined for reasons of family reunification, medical needs, or other special circumstances. For refugees and asylum seekers whose placement is not predetermined, decisions in both the United States and Switzerland are driven primarily by capacity constraints at the locations, without a systematic attempt to optimize with respect to refugee/asylum-seeker outcomes. In Switzerland, the assignment to cantons is explicitly done on a quasi-random basis, subject to the proportionality key.

Finally, the assignment batch size also varies by country. In Switzerland, the assignment is done on a one-by-one basis for each family after their postarrival processing, and a number of other countries (e.g., Netherlands) follow a similar procedure. In the United States, assignment decisions are made on a weekly basis. Although the paper focuses on one-by-one assignment, Online Appendix EC.4 discusses how the proposed methods can be readily extended to settings with batching.

2.2. Data and Scope

For the U.S. context, we use (de-identified) data on refugees of working age (ages 18–64) who were resettled in 2015–2016 into the United States by one of the largest U.S. refugee resettlement agencies. For the Swiss setting, we use (de-identified) data on adult asylum seekers geographically assigned in 2015–2016 who eventually received full protection status specified under the Geneva Convention, as well as those whose claim for Geneva protection status was rejected, but were awarded subsidiary protection.

In both contexts, placement officers centrally assigned each case (individual or family) in the data set to one of the possible locations—the 26 cantons in Switzerland and about 30 resettlement locations in the U.S. agency's network. Both data sets contain details on the refugees'/asylum-seekers' characteristics (such as age, gender, origin, etc.), their assigned locations, and their employment outcome to be used for optimization. In the U.S. context, the outcome is whether each refugee was employed 90 days after arrival at their assigned location. Refugees' employment status 90 days after their arrival is the key (and only) outcome metric that the resettlement agencies are required to report and that is tracked by the U.S. government. In Europe, labor market integration is typically more challenging and takes longer for asylum seekers, and, hence, we use a longer-term employment outcome in the Swiss context. Specifically, we focus on whether each asylum seeker attained any employment within their first three years after assignment.

In both contexts, only "free cases" (those without prior family ties in their host country) are included in this study. This allows us to present a model and algorithm that aligns with the ongoing Swiss implementation, which is scoped to include only free cases, as will be described in Section 5. However, Online Appendix EC.3 also shows how the proposed approaches can be extended to include cases with family ties, as may be the case in future implementations.

For each case, a vector of employment scores is constructed, where each element corresponds to the average probability for individuals within that case of finding employment (within 90 days for the United States and within three years for Switzerland) if assigned to the particular location. To generate each case's outcome score vector, the same methodology is employed as in Bansak et al. (2018). Specifically, we use the data to generate models that predict the expected employment success of an individual at any of the locations, as a function of their background characteristics. These models were then applied to the cases who were assigned in 2015–2016 ($n = 1{,}919$ for the United States and $n = 4{,}523$ for Switzerland) to generate their expected employment success at each location. This paper assumes that the employment scores are given for each case and evaluates the proposed assignment algorithms relative to these predicted values.

The free cases that were assigned in 2016 ($n = 1{,}175$ for the United States and $n = 1{,}502$ for Switzerland) are treated as the test cohorts in this paper. That is, the proposed algorithms are applied to these particular cohorts, in the specific order in which the families are logged as having actually arrived. The 2015 arrivals are utilized as historical data in the proposed algorithms. To further mimic the real-world process by which these cases would be assigned dynamically to locations, real-world capacity constraints are also employed such that each location can only receive the same number of cases that it actually received.

3. Notation and Model

Throughout, $[K]$ denotes the set of integers $\{1, \dots, K\}$, and $\mathbb{1}\{\cdot\}$ denotes the indicator function. Additionally, e_j denotes a vector with a value of one in the j-th component and zeros elsewhere. For a matrix $\mathbf{W} \in \mathbb{R}^{N_1 \times N_2}$ and vector $\mathbf{w} \in \mathbb{R}^{N_2}$, $[\mathbf{w}; \mathbf{W}] \in \mathbb{R}^{(N_1+1)\times N_2}$ denotes a new matrix whose first row is $\mathbf{w}$.

We will assume throughout most of the paper that the total number of arrivals in a given year is known in advance. This assumption is generally not true and is discussed further in Section 5.3. In reality, the projected arrival numbers determined by resettlement authorities (for instance, the numbers projected by each of the ten U.S. resettlement agencies in consultation with the U.S. State Department) are revised throughout the year. Under this assumption, without loss of generality, we will assume that one case arrives each time period and, thus, let T denote both the number of arrivals and the time horizon. Let M be the number of localities, indexed by j, with capacities/slots s_j, and $\sum_{j=1}^{M} s_j = T$. The capacities represent the number of individuals that each location can accommodate.

The arriving cases are indexed by t. For simplicity of exposition, it will be assumed that each case comprises exactly one individual or, equivalently, that the capacities are set at the case level (instead of the individual level), which aligns with the ongoing implementation in Switzerland. This is further discussed in Online Appendix EC.3.1, which also shows how the proposed methods can be extended to account for varying case sizes, along with individual-level capacities, as may be the situation in future implementations. We will let $a_j(t)$ be the number of cases allocated to location j *after* the allocation at time t and define $\tilde{s}_j(t) := s_j - a_j(t)$ as the remaining slots at location j after time t (i.e., at the start of time $t+1$).

The assignment of case t to location j results in a scalar outcome, w_{tj}. In the U.S. context, the value of w_{tj} represents the probability that case t will find employment within 90 days if assigned to location j, and in the Swiss context, it is the likelihood of finding employment within the first three years. In this paper, the outcome scores w_{tj} are assumed to be known. In practice, they are estimated using a machine learning model that takes a large number of covariates as input (see Bansak et al. 2018). In the online assignment problem, an arriving case is completely defined by its employment score vector, $\mathbf{w}_t$ (which is a function of the case's underlying covariates). Thus, we will use the matrix $\mathbf{W}$ with elements w_{tj} to denote an arbitrary population of T cases. Additionally, let $\mathbf{W}_t$ be shorthand for a population of arrivals from time t through T. We will assume that every free case can be assigned to any location with remaining capacity. In reality, even free cases may have idiosyncratic restrictions on which locations they can be assigned to (e.g., for medical reasons). This is further discussed in Online Appendix EC.3.3.

We will work in the underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where $\omega \in \Omega$ denotes a sample path of arrivals. Thus, there is a one-to-one correspondence between Ω and the set of all matrices $\mathbf{W}$, and fixing ω also fixes $\mathbf{w}_t$ for all $t \in [T]$. The vectors $\tilde{\mathbf{s}}(t-1)$ and set $\{\mathbf{w}_l\}_{l\in[t]}$ fully describe the state of the online assignment problem at time t. Therefore, let $S_t := (\tilde{\mathbf{s}}(t-1), \{\mathbf{w}_l\}_{l\in[t]})$ denote the state at time t. Note that if the arrivals in each time period are assumed to be independent, then the state could be described simply by $\tilde{\mathbf{s}}(t-1)$ and $\mathbf{w}_t$. To formalize the dynamics of the problem, the following features are assumed:

1. **Blind Sequentiality:** The cases are assigned in an order that is exogenously determined and unknown in advance, and each case t must be assigned before case $t+1$ is assigned.
2. **Nonanticipativity:** Each case t is assigned without knowledge of the outcome scores of the future arrivals.
3. **Permanence:** Assignments cannot be changed once they are made.

These features are representative of the real-world dynamics in many countries. Online Appendix EC.4 demonstrates how batching, which violates the nonanticipativity assumption, can be incorporated into the proposed algorithms, resulting in performance gains.

The binary variables z_{tj} are the key decision variables, with $z_{tj}=1$ if case t is assigned to location j and $z_{tj}=0$ otherwise. Let Φ denote a full assignment of cases to locations such that the capacity constraints are satisfied, and let $\phi(t)$ denote the assignment for case t (and, thus, $z_{t\phi(t)}=1$). Therefore, $w_{t\phi(t)}$ is the outcome of case t under assignment Φ, which could also be written as $\sum_{j\in[M]} z_{tj} w_{tj}$. The total employment score of matching Φ is given by

$$w(\Phi) := \sum_{t=1}^{T} w_{t\phi(t)} = \sum_{t=1}^{T}\sum_{j=1}^{M} w_{tj} z_{tj}. \tag{1}$$

3.1. Queueing Model

To capture the allocation balancing problem, each location will be treated as a server with a dedicated queue. Although there are no physical queues, this modeling framework captures the relevant trade-offs. To that end, it is assumed that each location has a *processing rate*, ρ_j, based on the resources (i.e., resettlement officers, service providers, and other related resources) at that location. This is the rate at which location j can handle incoming cases. For example, if $\rho_j = 1/2$, then location j is able to handle one case every two periods, on average. Resettlement officers, service providers, and local community resources cannot be moved across locations. Therefore, for simplicity of exposition, we assume that ρ_j is stationary. However, it is straightforward to adapt the analysis and proposed techniques to settings where ρ_j varies over time or by features of the cases. We will assume throughout that capacities are set to be commensurate with processing rates, so that $\rho_j T = s_j$. Note that this assumption is essentially met by design in the resettlement program, as capacities for each location are programmatically decided on the basis of the resources at each location. However, in practice, the value of ρ_j could also be determined through interviews with case officers, particularly to understand case-level heterogeneities in processing rates.

The *build-up* of location j at time t, for $t>2$, is given by

$$b_j(t) = \max\{0, b_j(t-1) - \rho_j\} + z_{tj}, \tag{2}$$

with $b_j(1) = z_{1j}$ for all $j \in \{1,\dots,M\}$. This is the build-up *up to and including* the assignment at time t, but *before* the processing at time t. This represents the number of cases either waiting or in process at time t. For each location, the ideal build-up level is in the interval $(0,1]$, indicating that the location is actively settling a case, and no cases are waiting. When $b_j(t) > 1$, cases are waiting to be processed at location j, and when $b_j(t) = 0$, location j is idle.

4. Outcome Maximization

This section proposes a minimum-discord online assignment algorithm that seeks to maximize the sum of outcome scores across the horizon. In this section, the build-up at each location is not considered. Section 6 will extend this algorithm by proposing a modified version that additionally seeks to minimize build-up.

First, we introduce the offline version of the outcome maximization problem. For a given set of arrivals **W**, the offline optimization problem is:

$$\begin{aligned} \max_{\mathbf{Z}} \ & \sum_{t=1}^{T}\sum_{j=1}^{M} w_{tj} z_{tj} \\ s.t. \ & \sum_{j=1}^{M} z_{tj} = 1 \quad \forall\, t \in [T] \\ & \sum_{t=1}^{T} z_{tj} = s_j \quad \forall\, j \in [M] \\ & \mathbf{Z} \in \{0,1\}^{T\times M}, \end{aligned} \tag{OutcomeMax}$$

where $\mathbf{Z}$ is the assignment matrix with elements z_{tj}. The solution to OutcomeMax is the outcome maximizing assignment for a population **W**. When a particular population or sample path is specified, we may write this problem as OutcomeMax($\mathbf{W}$) or OutcomeMax(ω), and its optimal objective value represents an upper bound for any assignment of that population or sample path. It is well known that an optimal solution to OutcomeMax can be found by solving the linear programming (LP) relaxation of OutcomeMax (Bertsimas and Tsitsiklis 1997). Thus, solving OutcomeMax is generally fast (e.g., for $T \le 3{,}000$, OutcomeMax can be solved in less than one second). See Online Appendix EC.5 for detailed run-time metrics.

The true online assignment problem is a dynamic program. In other words, the algorithm must make an assignment, given the current state, without knowledge of the outcome score vectors of future arrivals. Because of the online nature of the problem, it is helpful to let the notation OutcomeMax($\mathbf{W}_t, \tilde{\mathbf{s}}(t-1)$) describe solving OutcomeMax for time steps t onward for population $\mathbf{W}_t$, starting with capacities $\tilde{\mathbf{s}}(t-1)$.

In theory, the optimal solution to the dynamic problem could be found by solving Bellman's equation, given by

$$\begin{aligned} V_t(S_t) = \max_{\phi(t)\in[M]} \Bigg(& w_{t\phi(t)} + \int_{\omega\in\Omega} \mathbb{P}(\omega\,|\,S_t) V_{t+1}(\tilde{\mathbf{s}}(t-1) - e_{\phi(t)}, \\ & \{\mathbf{w}_l\}_{l\in[t]} \cup \mathbf{w}(\omega)_{t+1}) \Bigg) \\ s.t.\ & e_{\phi(t)} \le \tilde{s}_j(t-1)\ \forall\, j\in[M]. \end{aligned} \tag{3}$$

The optimal policy is the maximizer of the right-hand side of the equation above. Because of the so-called "curse of dimensionality" (Bellman 1966) arising from the large number of locations and continuous outcome scores, Problem (3) cannot be solved directly, even if the probabilities $\mathbb{P}(\omega|S_t)$ were known. Many heuristics and approximation methods have been proposed to solve Problem (3). Our chosen solution method, a special case of the Bayes Selector method introduced in Vera and Banerjee (2020), is described in the following section.

4.1. Minimum-Discord Online Algorithm

Let

$$\begin{aligned} Q(\phi(t), S_t) := \Bigg\{ & \omega \in \Omega : \phi(t) \notin \arg\max_j (w_{tj} \\ & + V_{t+1}(\tilde{\mathbf{s}}(t-1) - e_j, \{\mathbf{w}_l\}_{l\in[t]} \cup \mathbf{w}(\omega)_{t+1})) \Bigg\}, \end{aligned} \tag{4}$$

be the event that assigning case t to location $\phi(t)$ is not optimal, according to OutcomeMax(ω). This definition allows for the possibility that there are multiple optimal decisions, according to the offline benchmark. Furthermore, let $q(\phi(t), S_t) := \mathbb{P}[Q(\phi(t), S_t)|S_t]$ be the *disagreement probability*. The most general version of the Bayes Selector algorithm proposed by Vera and Banerjee (2020) chooses the location at time t that minimizes $q(\phi(t), S_t)$. The algorithm proposed in this paper chooses the location that minimizes an approximation of these disagreement probabilities in each time period. This approach is referred to as *minimum-discord* because the goal is to minimize the likelihood of disagreement with the offline optimal solution at time t. We note that this method does not take into account the degree of disagreement. An alternative approach could select the location that minimizes the expected optimality gap, as opposed to minimizing the likelihood of making a suboptimal decision. This is elaborated on in Online Appendix EC.8.

Vera and Banerjee (2020) establish performance guarantees for the Bayes Selector algorithm in many settings; however, the assumptions that underlie these guarantees do not hold in our setting, which places no assumptions on the underlying arrival distribution. The focus of this paper is on proposing explainable algorithms with strong empirical performance on the real-world setting. Nonetheless, in Online Appendix EC.1, we provide a characterization of the expected regret of any online algorithm in terms of the disagreement probabilities, following lemma 1 of Vera and Banerjee (2020).

Because there are no assumptions placed on the arrival process, we use a Monte Carlo sampling procedure to estimate $q(\phi(t), S_t)$. The intuition is as follows. When case t arrives, we generate K random trajectories of future arrivals $t+1$ through T, denoted by $\{\mathbf{W}_{t+1}^k\}_{k=1}^K$. For each random trajectory $k \in [K]$, the offline problem OutcomeMax$_t([\mathbf{w}_t; \mathbf{W}_{t+1}^k], \tilde{\mathbf{s}}(t-1))$ is solved.

Let $n_j(t)$ be the number of times that case t is assigned to location j across the K trajectories. The quantity $1 - q(j, S_t)$—namely, the probability that location j is an optimal action—is approximated by $n_j(t)/K$. Therefore, minimizing our approximation of $q(j, S_t)$ is equivalent to assigning case t to location $\arg\max_j n_j(t)$—that is, the location that they were assigned to most often in the random instances. The proposed method is formally defined below.

Method 1 (MinDiscord). *Case t is assigned to location*

$$\phi(t) := \arg\max_{j \in [M]} \sum_{k=1}^{K} z_{tj}^k,$$

with ties broken randomly, where

$$\mathbf{Z}^k = \arg\max_{\mathbf{Z}} \text{OutcomeMax}([w_t; W_{t+1}^k], \tilde{\mathbf{s}}(t-1)).$$

The Monte Carlo sampling approach requires a "sampling population" from which to draw a sample, which we denote by $\mathcal{A}$. In this paper, $\mathcal{A}$ comprises the 2015 arrivals. Algorithm OnlineMinDiscord, defined below, is the online assignment algorithm that employs Method 1 in each time period. We note that the choice of $\mathcal{A}$ should depend on the level of nonstationarity in the arrival process. If the data are highly nonstationary, $\mathcal{A}$ could comprise a shorter, more recent window of arrivals.

Algorithm OnlineMinDiscord (Min-Discord Online Assignment)

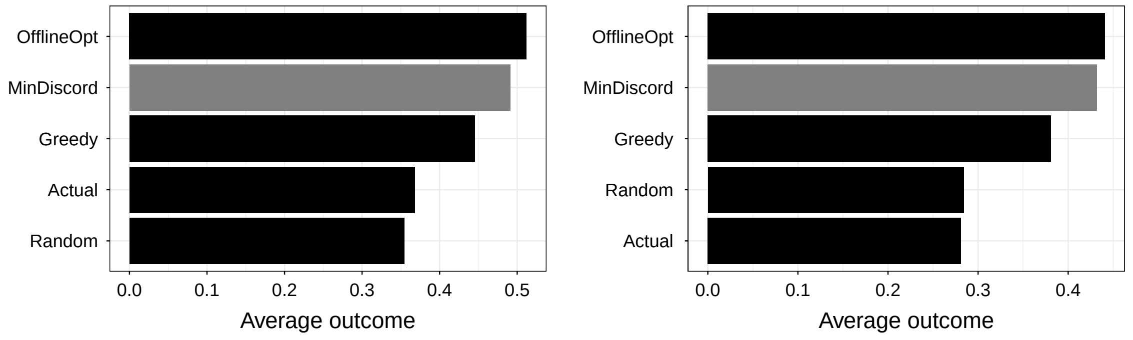
initialize $\tilde{s}_j(0) \leftarrow s_j$ for all $j \in \{1, \ldots, M\}$
for t in $1, \ldots, T$ **do**
  for k in $1, \ldots, K$ **do**
    $\mathbf{W}_{t+1}^k \leftarrow T - t$ randomly drawn cases from set $\mathcal{A}$ with replacement
    $\mathbf{Z}^k \leftarrow \arg\max$ OutcomeMax$([\mathbf{w}_t; \mathbf{W}_{t+1}^k], \tilde{\mathbf{s}}(t-1))$
  end for
  $\phi(t) \leftarrow \arg\max_j \sum_k z_{tj}^k$ (with ties broken randomly)
  $\tilde{\mathbf{s}}(t) \leftarrow \tilde{\mathbf{s}}(t-1) - e_{\phi(t)}$
end for
return $\Phi^{MD} = \{\phi(t)\}_{t=1\ldots T}$

4.2. Performance of OnlineMinDiscord

Figure 1 shows the results of applying OnlineMinDiscord to the 2016 arrivals (both U.S. and Swiss). Throughout the paper, unless otherwise specified, we use $K=5$ for OnlineMinDiscord. We compare OnlineMinDiscord to four benchmarks: the actual historical assignment, the hindsight-optimal solution, greedy assignment, and random assignment. The first benchmark assigns each case to the location to which they were assigned in reality under the status quo procedures. Although, for this benchmark, we could measure employment according to whether the cases actually found employment in reality (because this is contained in the data), for all benchmarks, we measure employment according to the predicted employment scores, $\mathbf{W}$, so that they are all evaluated with respect to the same metric. (We note, however, that using actual employment results in an almost identical total employment score for this benchmark.)

The hindsight-optimal solution, OfflineOpt, is included as a benchmark because, although it cannot be performed in a real-world dynamic context, it sets an upper bound of what is achievable by any algorithm. In the greedy algorithm, each case is assigned sequentially to the location with the highest expected employment score for that case, out of locations with remaining capacity. Finally, the employment score under random assignment for case t is given by $\sum_{j \in [M]} w_{tj} s_j / T$, which we include as a simple reference point. A comparison of OnlineMinDiscord to the method proposed by Ahani et al. (2023) is also included in Online Appendix EC.6,

Figure 1. Results of Online Algorithms and Benchmarks on U.S. Data (Left) and Swiss Data (Right) in 2016



though we note that the methods perform quite similarly.

Figure 1 shows the results. On the U.S. data, OnlineMinDiscord achieves 96% of the employment score of the hindsight-optimal solution. This is compared with the greedy, random, and actual historical assignment benchmarks, which achieve 87%, 69%, and 72% of the hindsight optimal employment levels, respectively. On the Swiss data, OnlineMinDiscord achieves 98% of the employment score of the hindsight-optimal solution. In this case, the greedy, random, and actual historical assignments achieve 86%, 64%, and 64% of the hindsight optimal solution, respectively. Outcomes by certain subgroups (e.g., nationality and sex) are shown in Figure EC.8 in the online appendix.

The optimality gap of OnlineMinDiscord is primarily because of nonstationarity in the arrival processes. When the arrival dates of the cases are randomly perturbed and $\mathcal{A}$ is taken to be the 2016 arrivals—mimicking a stationary process—the optimality percentage of the proposed algorithm compared with the hindsight-optimal solution increases to about 99.5% on both the U.S. and Swiss data. This was calculated as the average optimality percentage across 50 random instances, where in each instance, the arrival dates of the cases are randomly shuffled; in each of these instances, the optimality percentage was between 99.4% and 99.7%. However, the focus of this paper is on the performance of the proposed algorithms on the real, nonstationary, arrival data.

5. Implementation Details

This section provides details on the current pilot implementation of OnlineMinDiscord in Switzerland. Additionally, we discuss implementation complexities that motivated the development of a second algorithm, described in Section 6.

5.1. Background

In coordination with the SEM in Switzerland and a multiuniversity collaboration between researchers from ETH Zurich, Stanford University, Dartmouth College, Harvard University, and the University of California, Berkeley, a multiyear pilot implementation of OnlineMinDiscord is ongoing in Switzerland. The pilot began in January 2020 and is projected to end in 2024. As described further below, the pilot includes a randomized control trial (RCT) and targets the optimization of three-year employment outcomes; for this reason, results are not yet available, and the final results will not be available until three years after the completion of the pilot. The objective of the pilot is to generate rigorous evidence of impact on asylum-seeker employment, based upon which a broader and more permanent implementation of these methods can then be considered by the SEM.

The pilot implementation applies to all adult asylum seekers (or families that include at least one adult) who (a) obtained subsidiary or Geneva Convention protection status (and, hence, who are granted asylum and allowed to stay in Switzerland), (b) who are free to be assigned to any canton (i.e., do not have pre-existing family ties, medical constraints, or other special arrangements), and (c) are part of the "accelerated procedure" track in the Swiss asylum process. The accelerated procedure is used for relatively uncomplicated cases, whose status—whether they will be granted asylum or will be removed from Switzerland—can be designated in a relatively prompt manner, with a target of less than 100 days.

5.2. Pilot Setup

As described earlier, placement officers in Switzerland are in charge of determining the cantonal assignment of asylum seekers. In our pilot implementation, the placement officers have been provided with specialized software that generates a recommended canton for each asylum seeker case (i.e., family or individual). The placement officers maintain the ability to override the recommendation, if necessary, but they are encouraged to take the recommendation; as mentioned, the pilot scope includes only asylum-seeker cases that can be assigned to any canton. The RCT design is simple: each asylum-seeker case that will be assigned is first randomly allocated to either the control or treatment condition. In the control condition, the canton recommendation is generated randomly. In the treatment condition, the canton recommendation is generated via OnlineMinDiscord.

The distribution of asylum seekers in Switzerland follows a cantonal proportional distribution key. Accordingly, in our implementation, the assignment of asylum-seeker cases is subject to canton capacity constraints that follow this proportional distribution key, which is enforced separately for the treatment and control cases. In other words, the treatment and control cases have fully independent capacity at each canton to limit interference in the RCT. The capacity for cases that are out of scope (e.g., cases with family ties) are also independent from the pilot. Furthermore, capacities are set for each of the treatment and control cohorts at the case level in the pilot implementation. Hence, the implementation of OnlineMinDiscord is applied as described in the text with cases as the units of interest, though with one additional consideration: the proportional distribution constraints must be achieved independently for six different nationality groups, in accordance with a Swiss legal requirement. The six groups comprise asylum seekers from (1) Afghanistan, (2) Turkey, (3) Georgia, (4) the Maghreb countries, (5) a handful of specially identified countries (Albania, Benin, Burkina Faso, Bosnia and Herzegovina, Ghana, Guinea, Gambia, India, Moldova, the Republic of North Macedonia, Mongolia, Nigeria, Kosovo, Senegal, and Serbia), and (6) all other countries. The need to achieve proportional distribution independently for each of these six groups is tantamount to—and, hence, is achieved by—implementing OnlineMinDiscord separately and independently for each of these six groups.

5.3. Challenges

Because there is no advance processing prior to the arrival of asylum seekers, flows of asylum seekers can be somewhat unpredictable. Regional and global events, such as conflicts and wars, can lead to sudden changes in the types of asylum seekers who are arriving and their rate of arrival. Relative to the context of assigning UNHCR resettled refugees, this poses a more significant challenge for setting and controlling the capacity constraints, given that the number of cases that will need to be assigned by the end of the year (or within any period of time) is fundamentally uncertain. This uncertainty results in a violation of the modeling assumption that T—the total number of arrivals—is known in advance.

Nonetheless, ensuring that the distribution of the assignments across cantons meets the proportional allocation key by the end of each calendar year is of critical administrative importance, which requires that our capacity targets not exceed the actual number of annual arrivals without knowing what that number will be in advance. We employ intermittent updating of the capacity constraints to deal with this challenge in the pilot implementation. Because resources to process and receive new asylum seekers within each canton are limited and cannot freely move across cantons, it is also important that assignments to any given canton are not too concentrated within a period of time (e.g., if a canton's quota for the entire year were assigned to it in a single month).

To deal with both of these issues, we employ recent trends to project the number of arrivals in shorter intervals (e.g., one to four months) and intermittently add capacity, according to the proportional allocation key, over the course of the year. In doing so, we are able to avoid overloading any canton and protect against a divergence from the proportional allocation key. The cost, however, is inefficiency in two regards. First, the updating process itself entails analyses that cannot be easily automated and, hence, requires additional human labor. Second, introducing smaller chunks of capacity intermittently over time can cut into the ability of the algorithm to maximize gains.

These considerations and learnings from the pilot implementation have thus motivated our proposal for a second algorithm, (OnlineBalance), presented in the following sections, that maintains a balanced geographic distribution over time. By incorporating this allocation balancing component, OnlineBalance not only ensures that all locations have a steady stream of arrivals throughout the year, thus ensuring that local resources in any location are not outstripped by a sudden imbalanced influx of arrivals at any given time, but also naturally ensures that the overall distribution of cases will meet the proportionality targets, regardless of uncertainty in the arrival numbers.

6. Allocation Balancing

Motivated by learnings from the pilot implementation of OnlineMinDiscord, this section presents an extension that strives to maintain a balanced, proportional allocation over time to each locality by considering each locality to be a server with a dedicated queue (see Section 3.1 for the modeling details).

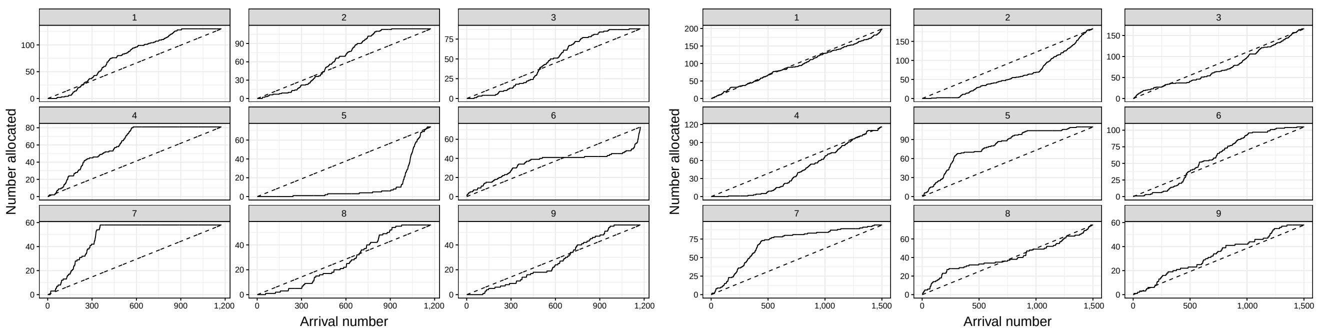
6.1. Imbalance Under Outcome Maximization

Although OnlineMinDiscord performs well in terms of maximizing outcomes, it results in a significant imbalance in the allocation to localities over time. Figure 2 shows the cumulative allocation to the largest nine locations over the horizon for the U.S. data (left) and Swiss data (right) obtained by using OnlineMinDiscord.

For the U.S. data, the average queue length (defined as $\max(b_j(t) - 1, 0)$ at location j at time t) across all locations resulting from OnlineMinDiscord is 7.5. For comparison, the average queue length under the actual historical assignment is 3.1. Thus, switching to an optimization approach does indeed lead to longer queues and wait times than under the status quo procedure. Similarly, for the Swiss data, the average queue length of OnlineMinDiscord is 6.3, compared with 3.3 under the actual historical assignments. We note that this is not simply a consequence of the particular choice of online algorithm, nor entirely a consequence of the online nature of the problem: even the hindsight-optimal solution results in imbalance over time (see Figure EC.7 in the online appendix).

This imbalance is primarily driven by nonstationarity in the arrival process. Indeed, when OnlineMinDiscord is applied to the same 2016 data, but with a randomly perturbed arrival sequence (and with $\mathcal{A}$ set to be the 2016 arrival cohort), mimicking a stationary process, the average queue length across five random instances is 1.7 (see Section EC.6 in the online appendix). Because refugee inflows are, in part, because of international events, there can be clustering of arrivals with specific background characteristics—particularly with respect to country of origin, which is one of the predictors that underlies the employment scores. This can lead to clustering in the subsequent assignment, causing imbalance. As described in Section 5,

Figure 2. Allocation to Nine Largest Locations over Time for U.S. Data (Left) and Swiss Data (Right) Using ONLINEMINDISCORD



this phenomenon encourages a conservative capacity updating approach in order to ensure that no location exceeds their proportionality key at the end of the horizon. An imbalanced allocation is also highly undesirable for resettlement service providers who cannot move between locations. The allocation balancing method, described in the following sections, mitigates these issues and provides ancillary benefits.

6.2. Offline Benchmark

Because the number of slots at each location is fixed, minimizing queue length/wait time is effectively equivalent to minimizing wait time *and* idle time. Therefore, we focus on minimizing wait time explicitly, while also noting the subsequent impact of the proposed methods on idle time. For simplicity of exposition, we will assume that the cost of wait time is identical across locations, although extending the algorithm to the nonidentical case is straightforward.

First, consider a new variant of the offline benchmark that penalizes wait time, given by:

$$
\begin{aligned}
\max_{\mathbf{Z},\mathbf{b}} \ & \sum_{t=1}^{T}\sum_{j=1}^{M} w_{tj}z_{tj} - \gamma \sum_{t=1}^{T}\sum_{j=1}^{M} \lceil b_j(t)-1 \rceil \, \mathbb{1}\{b_j(t)>1\} \\
\text{s.t. } & \sum_{j\in[M]} z_{tj} = 1 \quad \forall \ t\in[T] \\
& \sum_{t\in[T]} z_{tj} = s_j \quad \forall \ j\in[M] \qquad \text{(BALANCE)} \\
& b_j(t) = \max\{0, b_j(t-1)-\rho_j\} + z_{tj} \\
& \qquad \forall \ t\in\{2,\ldots,T\},\ j\in[M] \\
& b_j(1) = z_{1j} \quad \forall \ j\in[M] \\
& \mathbf{Z}\in\{0,1\}^{T\times M}.
\end{aligned}
$$

Recall that $b_j(t)$ denotes the build-up at location j at time t, and ρ_j is, again, the processing rate of location j. In the objective function of BALANCE, wait time cost is incurred when $b_j(t) > 1$, and $\lceil b_j(t)-1\rceil$ is the number of cases waiting at time t. The parameter γ (assumed to be nonnegative) is a weight that balances the trade-off between outcomes and wait time and can be thought of as the relative cost of wait time. In practice, this parameter could be set either according to a cost-benefit analysis, such that the units of measure were commensurate with one another, or according to an empirically driven decision on a value that results in acceptable balance across locations over time.

Let BALANCE$(\mathbf{W}_t, \tilde{\mathbf{s}}(t-1), \mathbf{b}(t-1))$ denote solving BALANCE from time t onward, for population $\mathbf{W}_t$ with capacities $\tilde{\mathbf{s}}(t-1)$ and initial build-up $\mathbf{b}(t-1)$. Recall that in ONLINEMINDISCORD, OUTCOMEMAX is solved K times for each new arrival, each time using a randomly generated sample of future arrivals. This same approach will be used to develop the new online allocation balancing assignment algorithm.

However, unlike OUTCOMEMAX, BALANCE cannot be solved to optimality as a linear program. The variables $b_j(t)$ are defined by nonlinear expressions, the objective function of BALANCE is nonlinear, and, finally, the assignment variables are binary. Because of advances in mixed-integer programming (MIP), BALANCE can still be solved using state-of-the-art MIP solvers, and one can obtain partial speed-ups by linearizing and relaxing parts of the problem. However, these approaches nonetheless result in substantially increased run-time compared with OUTCOMEMAX (see Online Appendix EC.5 for further discussion). Thus, instead of using BALANCE as our offline problem, we propose an alternative method that uses a greedy version of BALANCE. We show that this approach results in strong empirical performance and argue why a greedy approach is reasonable for allocation balancing.

6.3. Online Allocation Balancing Algorithm

In this section, we propose a greedy version of BALANCE to use as the offline problem in the online allocation balancing algorithm. In an online setting, the past assignments to each location are readily observable. Thus, at time t, the online algorithm has access to $b_j(t-1)$ for all locations j. Consider the following problem at time t:

$$\begin{aligned} \max_{\mathbf{Z}} \; & \sum_{l=t}^{T}\sum_{j=1}^{M} w_{lj} z_{lj} - \gamma \sum_{j=1}^{M} z_{tj} \left\lceil \frac{b_j(t-1)-\rho_j}{\rho_j} \right\rceil \mathbb{1}\{b_j(t-1)>0\} \\ \text{s.t.} \; & \sum_{j\in[M]} z_{lj} = 1 \quad \forall\, l \in \{t,\dots,T\} \\ & \sum_{l=t}^{T} z_{lj} = \tilde{s}_j(t) \;\; \forall\, j \in [M] \\ & \mathbf{Z} \in \{0,1\}^{N\times M}. \end{aligned} \tag{GBalance}$$

GBalance takes $\mathbf{b}(t-1)$ as input and weights the employment score of case t by the wait time cost incurred by case t. The wait time that case t experiences if assigned to location j is the length of time until all earlier cases are done being processed, starting from time t—namely, $\lceil (b_j(t-1)-\rho_j)/\rho_j \rceil$. Because $\mathbf{b}(t-1)$ is known prior to the t-th arrival, GBalance has a linear objective function. Thus, as with OutcomeMax, the optimal solution to GBalance can be found by solving its LP relaxation. In fact, solving GBalance is as fast as solving OutcomeMax, making this problem appealing for use in an online setting.

To build intuition for GBalance, we present the following lemma, which bridges Balance and GBalance. The proof of Lemma 1 can be found in Online Appendix EC.2.

Lemma 1. *The objective function of* Balance *is equivalent to*

$$\sum_{t=1}^{T}\sum_{j=1}^{M} w_{tj} z_{tj} - \gamma \sum_{t=1}^{T}\sum_{j=1}^{M} z_{tj} \left\lceil \frac{b_j(t-1)-\rho_j}{\rho_j} \right\rceil \mathbb{1}\{b_j(t-1)>0\}. \tag{5}$$

Notice that the objective function of GBalance is a *greedy* version of Expression (5)—namely, it does not calculate wait time for the entire horizon, but does so only for the current arrival (hence the name **Greedy Balance**).

Although the greedy method does not work well when it comes to outcome maximization, it does work well for minimizing wait time. In terms of wait time, taking a slot from location j immediately increases the build-up at location j. This makes it less likely for arrivals in the near-future to be assigned to location j, which could be consequential, especially if, for some of these arrivals, location j is highly desirable. However, this effect is short-lived: it is only relevant if an arrival in the near future (i.e., before the current arrival can be fully processed) would also be assigned to location j. Thus, if there are many locations, or if location j has few slots (implying that the probability of any given arrival being assigned to location j is small), this effect is mitigated.

Accordingly, we propose a new assignment method. Method 2 is similar to Method 1, but assigns the current arrival based on the solution to GBalance, instead of OutcomeMax as in Method 1.

Method 2 (Allocation-Balancing MinDiscord). *Case t is assigned to location*

$$\phi(t) = \arg\max_{j\in[M]} \sum_{k\in[K]} z^k_{tj},$$

with ties broken randomly, where

$$\mathbf{Z}^k = \arg\max_{\mathbf{Z}} \text{GBalance}([\mathbf{w}_t; \mathbf{W}^k_{t+1}], \tilde{\mathbf{s}}(t-1), \mathbf{b}(t-1)).$$

The online algorithm based on Method 2 is presented as Algorithm OnlineBalance.

Algorithm OnlineBalance (Allocation-Balancing Online Assignment)

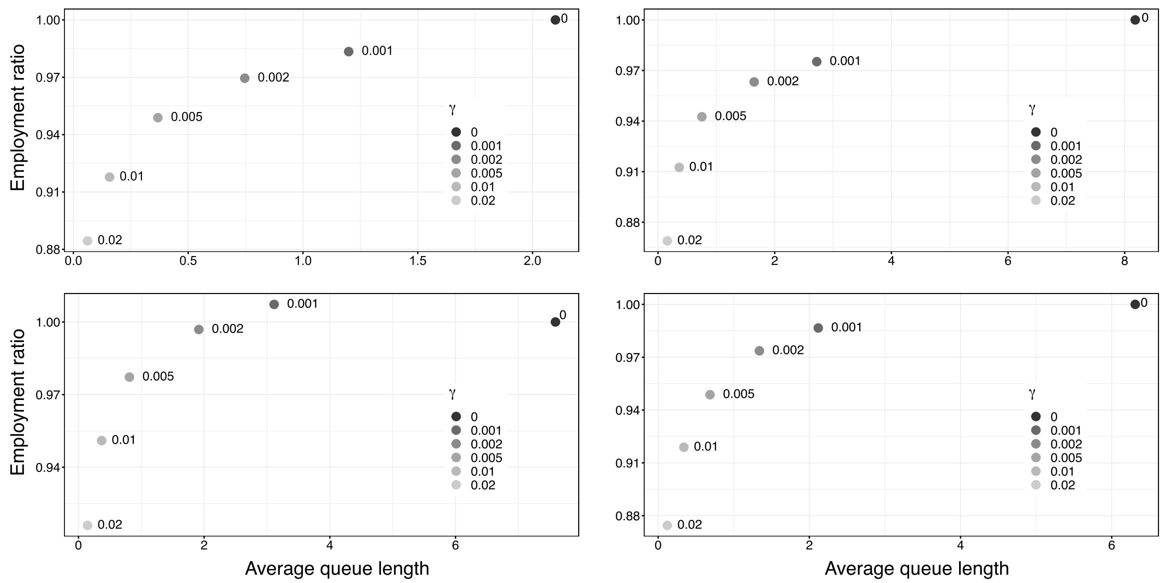
1: **initialize** $\tilde{s}_j(0) \leftarrow s_j$ for all $j \in \{1,\dots,M\}$
2: **for** t in $1,\dots,T$ **do**
3:     **for** k in $1,\dots,K$ **do**
4:         $\mathbf{W}^k_{t+1} \leftarrow T-t$ randomly drawn cases from set $\mathcal{A}$
5:         $\mathbf{Z}^k \leftarrow \arg\max$ GBalance$([\mathbf{w}_t; \mathbf{W}^k_{t+1}], \tilde{\mathbf{s}}(t-1), \mathbf{b}(t-1))$
6:     **end for**
7:     $\phi(t) \leftarrow \arg\max_j \sum_{k\in[K]} z^k_{tj}$ (ties broken randomly)
8:     $\tilde{\mathbf{s}}(t) \leftarrow \tilde{\mathbf{s}}(t-1) - e_{\phi(t)}$
9:     $\mathbf{b}(t) \leftarrow \max\{0, (\mathbf{b}(t-1)-\boldsymbol{\rho})\mathbb{1}_{t>1} + e_{\phi(t)}\}$
10: **end for**
11: **return** $\Phi^{GB} = \{\phi(t)\}_{t=1\dots T}$

6.4. Performance of OnlineBalance

Recall that the parameter γ controls the trade-off between allocation balancing and outcome maximization. Therefore, using historical data, the policymaker can tune this parameter to obtain the desired level of employment and allocation balance. In situations in which the payoff/cost of outcomes, wait time, and idle time can all be measured in or converted to a common metric (such as dollars), policymakers might want to set γ to the specific value that leads to optimization of that common metric.

Figure 3 shows the employment level and average queue length incurred by various values of γ on both the 2015 and 2016 arrivals from United States and Switzerland. The vertical axis of Figure 3 shows the employment level under a particular value of γ divided by the employment level when $\gamma = 0$ (i.e., under pure outcome

Figure 3. Trade-Off Between Outcome Maximization and Allocation Balancing for U.S. Data (Left) and Swiss Data (Right)



Note. The top row shows the results using 2015 data, and the bottom row uses the 2016 data.

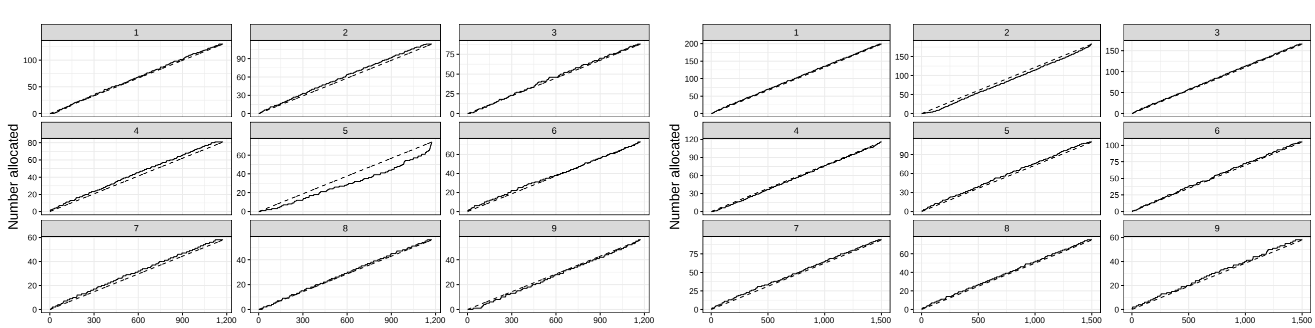
maximization). The x-axis shows the average queue length across affiliates and arrivals. In practice, we would not know the "best" value of γ to choose in a given year in hindsight and would need to base this decision on historical data. Therefore, for the 2016 cohorts, the value of γ should be chosen using the top row of Figure 3 (which uses 2015 data), and the resulting employment and build-up can be seen in the bottom row. Interestingly, in the 2016 U.S. data (Figure 3, bottom left), the highest employment level is not achieved when $\gamma = 0$, but when γ is slightly positive, likely because of idiosyncratic nonstationarities in the arrival process.

As can be seen from Figure 3, build-up can be dramatically reduced with little loss in employment. The ideal region in Figure 3 is the top left—where build-up is minimized, and employment is maximized. Based on the top row of Figure 3, a policymaker could choose an appropriate value of γ to achieve their desired balance of employment versus allocation balancing for the 2016 cohorts.

To illustrate these results in greater detail, Figure 4 shows the allocation of the 2016 arrivals using ONLINEBALANCE with $\gamma = 0.005$. This can be compared with Figure 2. From visual inspection alone, it is clear that ONLINEBALANCE results in a much more balanced allocation over time. Indeed, the average queue length is less than one. Furthermore, the total employment level obtained using ONLINEBALANCE with $\gamma = 0.005$ is 98% of the level obtained with ONLINEMINDISCORD. On the Swiss data, the results are similar: the average queue length is less than one, and the employment level obtained is 95% of the level obtained under ONLINEMINDISCORD.

Thus, with little loss in employment, ONLINEBALANCE is able to achieve a highly balanced allocation over

Figure 4. Allocation to Nine Largest Locations over Time Using ONLINEBALANCE with $\gamma = 0.005$ for U.S. Data (Left) and Swiss Data (Right)



Operations Research, 2024, vol. 72, no. 6, pp. 2375–2390, © 2024 INFORMS

time. A balanced allocation results in an even workload for resettlement officers and immediately solves many issues associated with updating capacities over time, discussed in Section 5. Specifically, the capacity of each location could be set according to an upper confidence bound on the number of arrivals, without running the risk of particular locations exceeding their proportionality key by the end of the horizon.

6.5. Ancillary Benefit: Increased Exploration

Although this paper does not focus on the outcome prediction methodology, the prediction and assignment steps are not independent (as discussed in Kasy and Teytelboym 2023). In this paper, it was assumed that the outcome scores are known. In practice, these outcome scores are estimated from historical data. To use the proposed methods, reliable scores must be determined for every combination of covariates and locations. If these scores are generated via statistical estimation procedures, maintaining some degree of exploration—assigning similar cases to different locations—is crucial to the resiliency of the estimation procedure, given the nonstationarity of the environment. The need for exploration in these situations is a well-known issue and is not unique to the refugee matching context.

A nonstationary constrained contextual bandit framework could be used to formally address this problem. However, without formalizing the bandit version of this problem, we note that OnlineBalance achieves higher levels of exploration than OnlineMinDiscord. Intuitively, because of the balancing component of the objective function in GBalance, the assignment of a case not only depends on their predicted employment score and the remaining capacity vector, but also depends on the current build-up at each location, effectively adding a degree of randomness to the assignment.

To demonstrate this idea, we run OnlineMinDiscord and OnlineBalance 100 times each for the first 100 arriving U.S. cases in 2016, where the arrival order is randomly permuted in each of the 100 instances. Let case i be the case that arrived i-th in the true arrival sequence. In the 100 random instances, they could arrive on any of the 100 days. For each case, we compute the number of times that they are assigned to each location. Let $\ell_{i,1}$ be the location to which case i is most often assigned, $\ell_{i,2}$ be their second most assigned location, etc. Let $n_{\ell_{i,k}}$ be the number of times that case i was assigned to their kth-most-assigned location out of the 100 instances. Figure 5 shows a bar chart of the average value of $n_{\ell_{i,k}}/100$ under OnlineMinDiscord and OnlineBalance. Note that $\sum_k n_{\ell_{i,k}} = 100$ for each case i. If $n_{\ell_{i,1}} = 100$, then $n_{\ell_{i,k}} = 0$ for all $k > 1$, and case i did not "explore" at all. The more uniform the values of $n_{\ell_{i,k}}$, the greater the exploration.

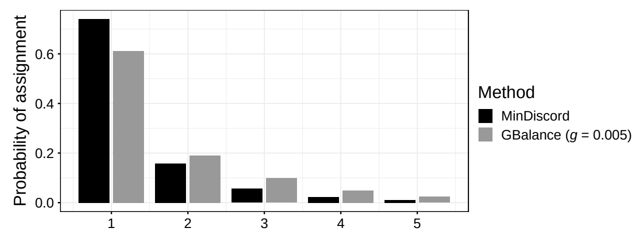
As shown in Figure 5, under OnlineBalance, the average value of $n_{\ell_{i,1}}/100$ is about 0.61 (meaning that a case was assigned to their "top" location 61% of the time), whereas under OnlineMinDiscord, the value is about 0.74. Additionally, the average number of unique locations that the same case was assigned to under OnlineMinDiscord was 3.97, versus 5.31 under OnlineBalance. This suggests that OnlineBalance may be preferable to OnlineMinDiscord from a resiliency perspective, naturally maintaining a higher degree of exploration.

7. Conclusions

This study proposed two assignment algorithms for matching refugees to localities. The first method, OnlineMinDiscord, seeks to maximize the employment scores of all refugees over a horizon by minimizing the probability of disagreement between the online algorithm and an offline benchmark. On the Swiss asylum-seeker data used in this study, this method is able to achieve 98% of the hindsight-optimal employment score. This is a significant improvement over the actual historical assignment, random assignment, and greedy assignment, which achieve 64%, 65%, and 87% of the hindsight-optimal employment scores, respectively. Similar results are found using U.S. data. OnlineMinDiscord is currently employed in a multiyear pilot in Switzerland.

However, OnlineMinDiscord—and any outcome maximizing algorithm—may result in severe periodic imbalance across the localities in the presence of nonstationary arrivals. This creates implementation challenges and an imbalanced workload for the local caseworkers, service providers, and other community members who help each newly arriving family get settled. Furthermore, if local capacities must be revised throughout the year because of larger or smaller arrival numbers than anticipated, imbalance in the allocation over time makes that capacity revision process more challenging. Therefore, we proposed a second assignment algorithm that directly seeks to balance the allocation over time to the localities, while still achieving high employment levels. On the U.S. and Swiss refugee resettlement data used in this

Figure 5. Average Probability of Being Assigned to the kth Ranked Location, Where Locations Are Ranked at the Case Level According to Their Assignment Probabilities



study, the allocation balancing method is able to significantly increase balance with little loss in employment.

By all indications, the challenges and scale of forced migration will continue to grow into the future. The methods presented here build upon recent research on outcome-based refugee assignment and could be integrated into refugee resettlement and asylum programs in many host countries—such as the United States, Netherlands, Switzerland, Sweden, and Norway—to help improve the lives of some of the world's most vulnerable populations.

Acknowledgments

The funders had no role in the data collection, analysis, decision to publish, or preparation of the manuscript. The authors thank Global Refuge and the Swiss State Secretariat for Migration (SEM) for access to data and guidance. The U.S. and Swiss refugee data used in this study were provided under a collaboration research agreement with Global Refuge and the SEM, respectively. These agreements require that these data not be transferred or disclosed. The authors are Faculty Affiliates of the Immigration Policy Lab (IPL) at Stanford University and ETH Zurich. This work is part of IPL's GeoMatch project. Finally, the authors are grateful to Daniel Freund, Jens Hainmueller, and Dominik Hangartner for helpful comments.

References

Acharya A, Bansak K, Hainmueller J (2022) Combining outcome-based and preference-based matching: A constrained priority mechanism. *Polit. Anal.* 30(1):89–112.

Afeche P, Caldentey R, Gupta V (2022) On the optimal design of a bipartite matching queueing system. *Oper. Res.* 70(1):363–401.

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

Ahani N, Gölz P, Procaccia AD, Teytelboym A, Trapp AC (2023) Dynamic placement in refugee resettlement. *Oper. Res.*, epub ahead of print September 28, https://doi.org/10.1287/opre.2021.0534.

Andersson T, Ehlers L (2020) Assigning refugees to landlords in Sweden: Efficient, stable, and maximum matchings. *Scand. J. Econom.* 122(3):937–965.

Andersson T, Ehlers L, Martinello A (2018) Dynamic refugee matching. Working Paper 2018:7, Department of Economics, Lund University, Lund, Sweden.

Arlotto A, Gurvich I (2019) Uniformly bounded regret in the multisecretary problem. *Stochastic Systems* 9(3):231–260.

Azar Y (1998) On-line load balancing. Fiat A, Woeginger GJ, eds. *Online Algorithms*, Lecture Notes in Computer Science, vol. 1442 (Springer, Berlin, Heidelberg), 178–195.

Balseiro S, Lu H, Mirrokni V (2021) Regularized online allocation problems: Fairness and beyond. *Proc. 38th Internat. Conf. Machine Learn.*, vol. 139 (PMLR, New York), 630–639.

Bansak K, Paulson E, Rothenhäusler D (2023) Learning under random distribution shifts. Preprint, submitted June 5, https://arxiv.org/abs/2306.02948.

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.

Bellman R (1966) Dynamic programming. *Science* 153(3731):34–37.

Bertsimas D, Tsitsiklis JN (1997) *Introduction to Linear Optimization*, Athena Scientific Series in Optimization and Neural Computation, vol. 6 (Athena Scientific, Belmont, MA).

Bertsimas D, Farias VF, Trichakis N (2013) Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. *Oper. Res.* 61(1):73–87.

Bray R (2019) Does the multisecretary problem always have bounded regret? Preprint, submitted December 17, https://dx.doi.org/10.2139/ssrn.3497056.

Bumpensanti P, Wang H (2020) A re-solving heuristic with uniformly bounded loss for network revenue management. *Management Sci.* 66(7):2993–3009.

Delacrétaz D, Kominers SD, Teytelboym A (2023) Matching mechanisms for refugee resettlement. *Amer. Econom. Rev.* 113(10): 2689–2717.

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

Devanur NR, Jain K, Sivan B, Wilkens CA (2019) Near optimal online algorithms and fast approximation algorithms for resource allocation problems. *J. ACM* 66(1):1–41.

Ding Y, McCormick S, Nagarajan M (2018) A fluid model for an overloaded bipartite queueing system with heterogeneous matching utility. Preprint, submitted August 22, https://dx.doi.org/10.2139/ssrn.2854492.

Fernández-Huertas Moraga J, Rapoport H (2015) Tradable refugee-admission quotas and EU asylum policy. *CESifo Econom. Stud.* 61(3–4):638–672.

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

Freund D, Lykouris T, Paulson E, Sturt B, Weng W (2023) Group fairness in dynamic refugee assignment. Preprint, submitted January 25, https://arxiv.org/abs/2301.10642.

Gölz P, Procaccia AD (2019) Migration as submodular optimization. *Proc. Conf. AAAI Artificial Intelligence* 33:549–556.

Jaillet P, Lu X (2012) Near-optimal online algorithms for dynamic resource allocation problems. Preprint, submitted August 13, https://arxiv.org/abs/1208.2596.

Karp RM, Vazirani UV, Vazirani VV (1990) An optimal algorithm for on-line bipartite matching. *Proc. Twenty-Second Annu. ACM Sympos. Theory Comput.* (Association for Computing Machinery, New York), 352–358.

Kasy M, Teytelboym A (2023) Matching with semi-bandits. *Econom. J.* 26(1):45–66.

Leshno JD (2022) Dynamic matching in overloaded waiting lists. *Amer. Econom. Rev.* 112(12):3876–3910.

Moraga JF-H, Rapoport H (2014) Tradable immigration quotas. *J. Public Econom.* 115:94–108.

Mousa S (2018) Boosting refugee outcomes: Evidence from policy, academia, and social innovation. Preprint, submitted October 2, https://dx.doi.org/10.2139/ssrn.3259255.

Nguyen H, Nguyen T, Teytelboym A (2021) Stability in matching markets with complex constraints. *Management Sci.* 67(12): 7291–7950.

Olberg N, Seuken S (2019) Enabling trade-offs in machine learning-based matching for refugee resettlement. Technical report, University of Zurich, Zurich.

United Nations (2023) Global trends: Forced displacement in 2022. Accessed March, 2024, https://www.unhcr.org/global-trends-report-2022.

Ünver MU (2010) Dynamic kidney exchange. *Rev. Econom. Stud.* 77(1):372–414.

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 (2020) The Bayesian prophet: A low-regret framework for online decision making. *Management Sci.* 67(3): 1368–1391.

Vera A, Arlotto A, Gurvich I, Levin E (2020) Dynamic resource allocation: The geometry and robustness of constant regret. Working paper, Cornell University, Ithaca, NY.

Kirk Bansak is an assistant professor in the Department of Political Science at the University of California (UC), Berkeley. He is the inaugural faculty director of the Yardi Scholarship at UC Berkeley, a faculty affiliate of the Immigration Policy Laboratory at Stanford University and ETH Zurich, and a faculty affiliate of the Berkeley Interdisciplinary Migration Initiative.

Elisabeth Paulson is an assistant professor in the Technology and Operations Management Unit at Harvard Business School. She is also a faculty affiliate with the Immigration Policy Laboratory at Stanford University.