An Algorithm for Redistributing Disproportionate Numbers of Political Asylum Applications

Samuel Stanton

Advised by Dr. Renaud Chicoisne

A Thesis Presented On

May 1, 2017

In partial fulfillment of the

requirements for the degree of

Bachelor of Science in Applied Math

with Departmental Honors

University of Colorado Denver

DEDICATION

This work is dedicated to my parents, Gerald and Heather Stanton,
for teaching me to rise to every challenge I face.

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor, Dr. Renaud Chicoisne, for his unwavering support. His assistance in formulating, proving, and writing this thesis has been invaluable.

CONTENTS

1. INTRODUCTION

Since 2015, political strife and regional instability in the Middle East have caused significant portions of the populations of countries such as Syria to flee en masse. This migration, arising out of desperation, often occurs through irregular methods, without documentation and via hazardous methods of travel. This swell in the worldwide refugee population has overwhelmed the existing infrastructure of the countries hit the hardest by the surging waves of supplicants. This has prompted a widespread interest in international cooperation to mitigate the crisis. This is particularly true among the member nations of the European Union. Under an existing rule known as the Dublin Regulation, the member country responsible for the examination of an irregularly migrating asylum applicant is designated to be the first point of irregular entry into the EU. The flaws in the Dublin System have become increasingly apparent as the responsibility for numbers of applications far in excess of most countries' asylum infrastructure was assigned *de facto* to EU countries occupying accessible entry points to Europe, such as Greece. In 2015 alone, the number of refugees crossing into Europe exceeded one million (BBC News 2016).

All member nations of the EU must abide under Common European Asylum System (CEAS) regulations, which sets standards for the application decision process. Since the EU is an area of open borders and freedom of movement, once granted political asylum, a refugee is considered under the protection of the EU, and hence all its member nations. However the process of reviewing asylum applications can be lengthy and expensive, since all refugees must be interviewed and EU countries are required to provide shelter for all applicants until a decision is reached. Refugees are not required to apply for asylum at their point of entry, and many prefer to apply for asylum in northern EU nations. Many of these northern countries successfully argued from the Dublin Regulation that the responsibility for these applicants lay with the country through which the refugees first entered the EU, and demanded that applicants return to their point of entry for consideration (Open Society Foundations 2016).

In an attempt to relieve beleaguered EU border nations buckling under the volume of applicants, both those newly entering and those rejected from interior EU nations, the European Commission proposed the Dublin IV Regulation in an attempt to reform the CEAS. This regulation includes proposals for fairness mechanisms in order to correct serious imbalances of economic burden among the EU member nations. In particular, if a country is handling a disproportionate number of political asylum applications, all future asylum applications are to be redirected to other member nations with excess capacity (European Commission 2017). This raises the question of what method should be used to determine where these refugees should be reassigned. In this work, we will examine a methodology that allows refugees to exercise some agency in this process, while ensuring an equitable distribution of economic burden among haven countries. We will further extend the model to assign refugees not just to countries in general, but to specific locations based on location specific capacities.

This work is not intended to be an exhaustive examination of every factor that should be considered when determining where to reassign asylum applications. Rather, we seek to explore a general formulation of the problem and discuss an algorithmic approach to solving such a formulation. To begin, we have two main objectives. The first is clearly balancing the load of asylum application management equitably among the EU member countries. The second is attempting to maximize refugee satisfaction with the country to which they are redirected. Discontent refugee populations can exacerbate political tensions that typically already exist between nationals and the expatriated. Attempting to match asylum applicants with their preferred haven country can help ease the difficult asylum application decision process, and also reduces the amount of post-decision migration, since refugees are already in the purview of one of their preferred haven countries. This will improve the ability of these haven countries to manage the eventual resettlement of these refugees. Hence taking refugee preferences for their settlement location into account when determining how to reallocate asylum applicants promises significant benefits to the efficiency and humanity of the application system.

A complete reassignment process would determine not only which country should assume responsibility of the asylum application, but would also direct the refugee to a specific application processing location. For the purposes of this work, these locations are taken to be the cities in the EU member nations with population over 200,000. The capacity of a city's ability to process asylum application is assumed to be proportional to its population. This approach helps ensure that the processing load is evenly distributed within the countries, as well as across the EU countries. Ideally this model would also take into account the economic prosperity of the city, but implementing such a factor is outside the scope of this work. However, altering the capacity of a city to a more accurate fixed metric would not change the structure of the problem, and thus can only improve the existing model.

Quantifying refugee preferences presents a challenge, since such preferences are highly individual and subject to rapid change. In this work, we begin with a formulation for deterministic refugee preferences, and then proceed to develop a robust approach that accounts for variation in preference. One possible method of determining refugee preference would be requiring asylum applicants to rank a number of secondary choices of haven country in the case of reassignment. Refugees could then be grouped by preference and reassigned as needed. Clearly such an approach to quantifying refugee preference is unavailable to the author at this time. However, the data on asylum applications to the EU member countries by country of origin is easily accessible through Eurostat. Consequently we take the number of asylum applications from some origin country to a particular haven country relative to the total number of applications from that origin to be a reasonable proxy of the preference of refugees from that origin for that country. We further assume that the actual asylum processing location is inconsequential to the refugee, as long as it lies in the preferred country. This proxy is taken to be the actual preference in the deterministic model, and is used as a nominal value in the robust model.

The balance of a particular assignment of refugee applications is more easily quantified. Given observations of the burden of asylum applications for each city, the variance of those burdens is a reasonable measure of the balance, since it is in essence a measure of the average deviation of burden for any city from the mean burden. Thus an assignment with a low variance of burden will correspond to an assignment that places on each city a burden close to the mean on average. We will see that measuring balance in terms of burden variance is well suited for the algorithm we will propose, and is thus to be preferred over other possible candidates, such as a worst-case sensitive max norm.

The reassignment problem then, is a matter of assigning refugees grouped by origin, and thus preference, to a city in a haven country for application processing. We will attempt to send a refugee to his most preferred destination country, but will avoid sending refugees to overburdened countries. Our specific formulation and algorithm is presented in the next section. We subsequently present the results of implementing our model on a dataset taken from Eurostat databases.

2. PREFERENCE-ONLY OPTIMIZATION

2.1 Refugee Preferences

Let I be the set of refugee origin countries and J be the set of haven cities. Consider the bipartite graph $G := (V, E)$ where the nodes V are taken to be the elements of I and J, and every $i \in I$ is fully connected to the nodes in J by the set of edges $E := \big\{(i, j) \in I \times J\big\}$.

Let p_{ij} be the preference score of refugees from origin country $i \in I$ for the haven city $j \in J$. As stated earlier, we consider J to be all the cities in the EU with population greater than 200,000. Letting H be the set of countries in the EU, we estimate the preference of refugees from origin country i for each haven country h in the EU by using as a proxy the number of applications a_{ih} for asylum from i to country h divided by the total number of asylum applications from the origin country i to any nation in the EU. Defining $h(j)$ as the country that city $j \in J$ belongs to and assuming the preference of the refugees for any particular city in a haven country to be roughly equivalent, our estimate of the preferences is

$$p_{ij} = \frac{a_{ih(j)}}{\sum\limits_{h \in H} a_{ih}}, \forall j \in h(j) \tag{2.1}$$

Finally, for any source node $i \in I$ let r_i be the number of refugees that have to be assigned.

2.2 A Minimum Cost Flow Model

If the objective is narrowed to be only maximizing the overall refugee preferences, our problem can be reduced to finding a flow sending r_i units of flow from each node $i \in I$ to the nodes in J that yields a maximum overall preference score in the network depicted in Figure 2.1. In this case, we can model the problem as a minimum cost flow problem in a derived graph $G' := \big(V', E'\big)$.

Let $V' := V \cup \{s, t\}$, where s and t are extra source and sink nodes. We link s and t to the nodes in V as follows: A set of extra edges in E' connects s to each $i \in I$ and connects each $j \in J$ to t. In other words:

$$E' := E \cup \big\{(s, i) : i \in I\big\} \cup \big\{(j, t) : j \in J\big\} \tag{2.2}$$

The capacity of the arcs $(s, i) \in E'$ are set to r_i (the supply of asylum applicants at origin node $i \in I$), whereas the capacity of the arcs $(j, t) \in E'$ are set to be the population u_j of city j. We leave edges $(i, j) \in E$ uncapacitated. Setting the offer/demand of every node of N to zero and forcing the source node s to send $\sum\limits_{i \in I} r_i$ units of flow and the sink node t to receive the same amount of flow, we enforce that all the refugees are assigned to a city. We illustrate G' in Figure 2.2)

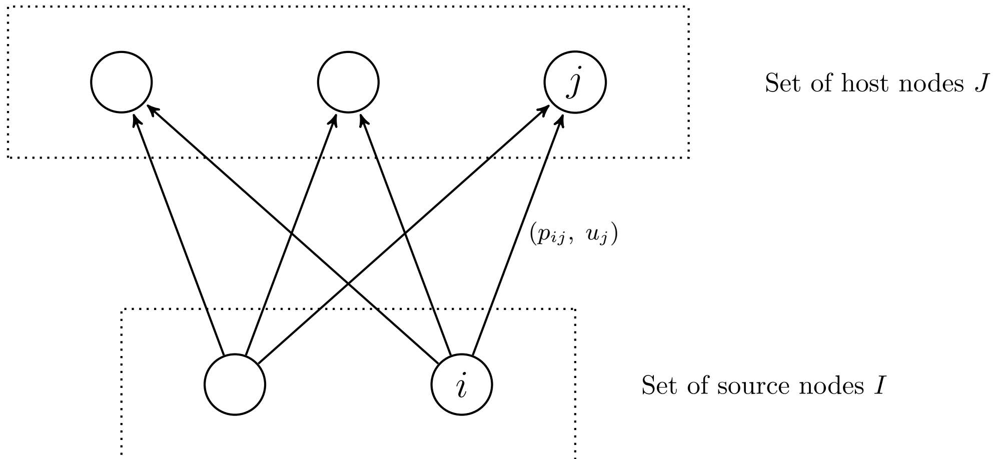

Fig. 2.1: The original bipartite graph G where the edge labels represent (cost, capacity).

It is straightforward that G' is also acyclic. This important observation makes possible the use of any minimum cost flow algorithm to solve our *maximization* problem in G' by minimizing *minus* the preferences. When considering only the preferences, the problem can then be reduced to find an optimal solution to the following minimum cost flow problem:

$$\min - \sum_{(i,j) \in E} p_{ij} x_{ij} \tag{2.3}$$

$$\text{s.t.} \sum_{j:(j,i) \in E'} x_{ji} - \sum_{j:(i,j) \in E} x_{ij} = \begin{cases} \sum_{i' \in I} r_{i'} & \text{If } i = s \\ 0 & \text{If } i \in V = I \cup J \\ - \sum_{i' \in I} r_{i'} & \text{If } i = t \end{cases} \tag{2.4}$$

$$x_{si} \leq r_i \qquad , \forall i \in I \tag{2.5}$$

$$x_{jt} \leq u_j \qquad , \forall j \in J \tag{2.6}$$

$$x_{ij} \geq 0 \qquad , \forall (i,j) \in E' \tag{2.7}$$

Where x_{ij} represents the amount of flow crossing edge $(i,j) \in E'$. This last problem can be rewritten in the following compact form:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & x \in X \end{aligned} \tag{2.8}$$

Where the cost vector c is defined as follows:

$$c_{ij} := \begin{cases} -p_{ij} & \text{If } (i,j) \in E \\ 0 & \text{Otherwise} \end{cases} \tag{2.9}$$

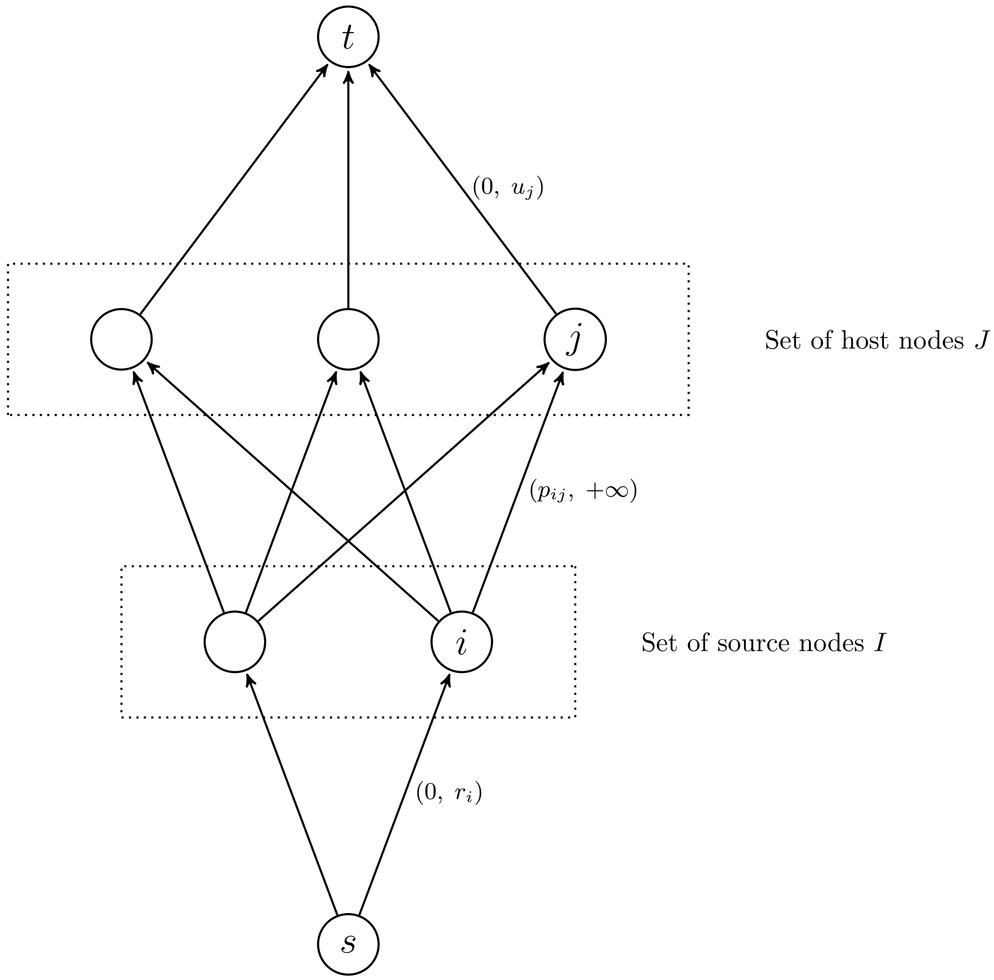

Fig. 2.2: Modified Graph G' where the edge labels represent (cost, capacity).

and x is a decision vector of size $\left|E'\right|$ belonging to the following set:

$$X := \left\{x : \text{ Constraints (2.4)-(2.7) are satisfied}\right\} \tag{2.10}$$

3. A MULTIOBJECTIVE SETTING

3.1 Balance and Preference Joint Optimization

In order to optimize load balance in addition to matching preference, we will utilize a multi-objective emphasis. The new objective function will attempt to minimize $c^\top x$ as before, but will also penalize flows that fail to balance the burden of asylum applicants. As mentioned previously, we will use the variance of the applicant burden as a measure of load balance.

First, given a feasible flow $x \in X$ on G', the refugee burden on city $j \in J$ is $\frac{x_{jt}}{u_j}$. In consequence the mean burden of flow x on the haven cities J is

$$\mu(x) = \frac{1}{n} \sum_{j \in J} \frac{x_{jt}}{u_j} \tag{3.1}$$

Thus, we can write the *balance* of flow x as follows:

$$b(x) := \frac{1}{n} \sum_{j \in J} \left(\frac{x_{jt}}{u_j} - \mu(x) \right)^2 \tag{3.2}$$

Now, the ideal goal would be to simultaneously minimize both the preferences *and* the balance, leading to the following bi-objective problem:

$$\min_{x \in X} \left[c^\top x; b(x) \right] \tag{3.3}$$

In this work, we will restrict our attention on parameterizations of the latter problem: Given a prioritization parameter, $\alpha \in [0, 1]$ we will tackle the following problem:

$$\min_{x \in X} \left\{ f(x) := \alpha c^\top x + (1 - \alpha) \, b(x) \right\} \tag{3.4}$$

3.2 Convexity

Now, we will show how to take advantage of the structure of problem (3.4) to solve it in an efficient way.

Proposition 1: f is convex and differentiable on $\mathbb{R}^n$.

Proof: First we examine the quadratic term,

$$b(x) := \frac{1}{n} \sum_{j \in J} \left(\frac{x_{jt}}{u_j} - \mu(x) \right)^2 \tag{3.5}$$

Let Q be a $|E'| \times |E'|$ matrix, where

$$Q_{(i,j),(i',j')} = 0 \qquad \text{If } j \neq t \text{ or } j' \neq t \tag{3.6}$$

$$Q_{(j,t),(j',t)} = \begin{cases} \frac{-1}{nu_j} & \text{If } j \neq j' \\ \frac{n-1}{nu_j} & \text{If } j = j' \end{cases} \tag{3.7}$$

Then we have

$$b(x) = \frac{1}{n}||Qx||_2^2 \tag{3.8}$$

From Boyd and Vandenberghe 2004, the composition $h(x) = g(Mx)$ of a convex function function $g : \mathbb{R}^n \to \mathbb{R}$ with an affine mapping $M : \mathbb{R}^n \to \mathbb{R}^n$ is convex. Moreover, since $\frac{1}{n} > 0$, $g(x) = \frac{1}{n}||x||_2^2$ is convex on $\mathbb{R}^n$, $b(x) = \frac{1}{n}||Qx||_2^2$ is convex. Second, $x \to c^\top x$ is linear, thus convex. Finally, given that f is the nonnegative weighted sum of two convex functions of x - $b(x)$ and $c^\top x$- then f is convex on $\mathbb{R}^n$. f is also obviously differentiable as the sum of a quadratic function and a linear operator. ■

3.3 Steepest descent algorithm

In this work we solve the refugee assignment problem with the steepest descent algorithm, as the objective is convex and differentiable. This allows to make use of gradients to find the steepest descent direction at each iteration $k \geq 1$ by minimizing the first order approximation of f at a current flow $x^{(k)}$. To do so - and find the steepest descent direction - we solve the following problem:

$$\min_{x \in X} f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^\top \left(x - x^{(k)}\right) \tag{3.9}$$

Removing the terms with no dependence in the decision variable x, the problem reduces to

$$\min_{x \in X} \nabla f\left(x^{(k)}\right)^\top x \tag{3.10}$$

This is exactly equivalent to solving a min cost flow problem on G' with edge costs changed to be the respective components of the gradient of f at the current incumbent $x^{(k)}$. Keeping in mind that G' is acyclic, even though the successive gradients will certainly yield negative edge costs, we can still solve the minimum cost flow problem with a Successive Shortest Path (SSP) algorithm. The acyclic nature of G' ensures that we do not encounter negative cycles when executing SSP.

Proposition 2: The gradient of the objective at some flow $x \in X$ is:

$$\frac{\partial f}{\partial x_{si}}[x] = 0 \qquad , \forall i \in I \tag{3.11}$$

$$\frac{\partial f}{\partial x_{ij}}[x] = \alpha c_{ij} \qquad , \forall (i,j) \in I \times J \tag{3.12}$$

$$\frac{\partial f}{\partial x_{jt}}[x] = \frac{2(1-\alpha)}{n}\left(\sum_{j' \in J}\left(\frac{x_{j't}}{u'_j} - \mu(x)\right)\left(\frac{\delta_{jj'}}{u_j} - \frac{1}{nu_j}\right)\right) \qquad , \forall j \in J \tag{3.13}$$

Proof: Since for any $i \in I$, x_{is} does not appear in the objective function, we clearly have:

$$\frac{\partial f}{\partial x_{si}}[x] = 0 \qquad , \forall i \in I \tag{3.14}$$

Second, because $f(x) = \alpha c^\top x + (1-\alpha)b(x)$, and no x_{ij} with $(i,j) \in E$ appears in the expression of $b(x)$, we have

$$\frac{\partial f}{\partial x_{ij}}[x] = \alpha \frac{\partial}{\partial x_{ij}} \left(c^\top x \right) = \alpha c_{ij} \tag{3.15}$$

Finally, for any $j \in J$, x_{jt} only appears in $b(x)$. We then have:

$$\frac{\partial f}{\partial x_{jt}}[x] = (1-\alpha) \frac{\partial b}{\partial x_{jt}}[x] \tag{3.16}$$

Recalling that

$$b(x) = \frac{1}{n} \sum_{j' \in J} \left(\frac{x_{j't}}{u_{j'}} - \mu(x) \right)^2 \tag{3.17}$$

we then have

$$\frac{\partial b}{\partial x_{jt}}[x] = \frac{2}{n} \left(\sum_{j' \in J} \left(\frac{x_{j't}}{u_j'} - \mu(x) \right) \left(\frac{\delta_{jj'}}{u_j} - \frac{\partial \mu}{\partial x_{jt}}[x] \right) \right) \tag{3.18}$$

Finally,

$$\frac{\partial \mu}{\partial x_{jt}}[x] = \frac{\partial}{\partial x_{jt}} \left(\frac{1}{n} \sum_{j' \in J} \frac{x_{j't}}{u_{j'}} \right) = \frac{1}{n u_j} \tag{3.19}$$

Hence:

$$\frac{\partial b}{\partial x_{jt}}[x] = \frac{2}{n} \left(\sum_{j' \in J} \left(\frac{x_{j't}}{u_{j'}} - \mu(x) \right) \left(\frac{\delta_{jj'}}{u_j} - \frac{1}{n u_j} \right) \right) \tag{3.20}$$

■

From the last proposition we then know that at iteration k, we have to solve the following minimum cost flow problem:

$$\min_{x \in X} \left\{ \alpha c^\top x + (1-\alpha) \sum_{j \in J} \frac{\partial f}{\partial x_{jt}} \left[x^{(k)} \right] x_{jt} \right\} \tag{3.21}$$

Let $\bar{x}^{(k)}$ be an optimal solution to (3.21).

Once the steepest descent direction is computed, we must find the best stepsize to take in that direction. To do this, let $d^{(k)} = \bar{x}^{(k)} - x^{(k)}$, and execute a linesearch to determine the optimal stepsize in this direction, $\lambda^{(k)}$. In other words, solve:

$$\lambda^{(k)} = \arg\min_{\lambda \in [0,1]} f \left(x^{(k)} + \lambda d^{(k)} \right) \tag{3.22}$$

Proposition 3: The linesearch has the following closed form solution:

$$\lambda^{(k)} = \underset{\lambda \in \{0, \bar{\lambda}^{(k)}, 1\}}{\arg\min} \left\{ f\left(x^{(k)} + \lambda d^{(k)}\right) \right\}, \tag{3.23}$$

where

$$\bar{\lambda}^{(k)} = \frac{-\theta\left(x^{(k)}, d^{(k)}\right)}{\phi(d^{(k)})} \tag{3.24}$$

$$\theta\left(x^{(k)}, d^{(k)}\right) = \alpha c^\top d^{(k)} + \frac{2(1-\alpha)}{n} \sum_{j \in J} \left(\mu\left(d^{(k)}\right) - \frac{d_j^{(k)}}{u_j}\right)\left(\mu\left(x^{(k)}\right) - \frac{x_{jt}^{(k)}}{u_j}\right) \tag{3.25}$$

$$\phi\left(d^{(k)}\right) = \frac{2(1-\alpha)}{n} \sum_{j \in J} \left(\mu\left(d^{(k)}\right) - \frac{d_j^{(k)}}{u_j}\right)^2 \tag{3.26}$$

Proof:

Given that the objective function of the lineasearch problem (3.22) is convex (it is the restriction on a line of an already convex function) its global minimum value is attained when

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\left(f\left(x^{(k)} + \lambda d^{(k)}\right)\right) = 0 \tag{3.27}$$

Which is equivalent to solving

$$\theta\left(x^{(k)}, d^{(k)}\right) + \lambda \phi\left(d^{(k)}\right) = 0 \tag{3.28}$$

Taking into consideration the boundary points for λ, we have the result. ∎

The next iterate in the steepest descent algorithm is thus $x^{(k+1)} = x^{(k)} + \lambda^{(k)} d^{(k)}$.

3.4 Complexity

Gradient methods often exhibit linear convergence, i.e. exhibit complexity $\mathcal{O}\left(\frac{1}{\varepsilon}\right)$ (Boyd and Vandenberghe 2004). As we shall see in the next section, in our case our algorithm does not appear to be an exception to this generality. The Successive Shortest Path (SSP) algorithm has pseudo-polynomial complexity $\mathcal{O}(mnC)$ when executed on a network having m edges, n nodes and an upper bound C for the magnitude of the edge costs. In practice this problem is applied on moderate size graphs with $n > 100$. The edge costs are either relative weights and thus have magnitude less than 1, or are gradients of a variance function on a high dimensional space, but are still bounded by some number C. Since for G' we have:

$$n := |I| + |J| + 2 \tag{3.29}$$

$$m := |I| \cdot |J| + |I| + |J| \tag{3.30}$$

Thus the overall complexity of our gradient descent with SSP subproblems is also pseudo-polynomial:

$$\mathcal{O}\left(C \cdot |I| \cdot |J| \frac{|I| + |J|}{\varepsilon}\right) \tag{3.31}$$

4. A ROBUST FORMULATION

4.1 Introducing Uncertainty to the Objective Function

In the last two sections we detailed our approach in the deterministic setting. However some of our parameters, in particular the preference weights p_{ij}, are just proxies for the real preferences, which are highly likely to vary. So far, we used the relative number of political asylum applications as a reasonable estimate of these preferences. We will continue to use this as a nominal value but we now consider optimizing f in a more robust setting by considering that the preferences are uncertain, but belong to an uncertainty set $\mathcal{U}$. Letting $q = |E| = |I| \cdot |J|$, we define our uncertainty set to be

$$\mathcal{U} = \{\widetilde{p} \in \mathbb{R}^q \ : \ \widetilde{p}_{ij} = p_{ij} + u_{ij}\hat{p}_{ij},\ \forall (i,j) \in I \times J \ , u \in \psi\} \tag{4.1}$$

Where $\hat{p}$ is some fixed perturbation term - for example, $\hat{p} = 0.1p$, which correspond to allow a deviation of at most 0.1 times the nominal value - and ψ is the perturbation set defined as

$$\psi := \{u \in \mathbb{R}^q \ : \ ||u||_2 \leq \rho\} \tag{4.2}$$

given some uncertainty allowance $\rho \in \mathbb{R}_+$. This describes an ellipsoidal perturbation set as defined in (Ben-Tal, El Ghaoui, and Nemirovski 2009).

For clarity, let us split the decision vector x in two parts:

$$x = \begin{pmatrix} x^E \\ x^{\bar{E}} \end{pmatrix} \tag{4.3}$$

Where x^E corresponds to the edges in E - i.e. the ones actually affected by the uncertainty - and $x^{\bar{E}}$ corresponds to the edges linking the source and sink s and t to the rest of the nodes. In consequence our robust formulation becomes

$$\min_{x \in X, z} \alpha z + (1-\alpha) b(x) \tag{4.4}$$

$$\text{s.t.} \tag{4.5}$$

$$z \geq \begin{pmatrix} -p \\ 0 \end{pmatrix}^\top \begin{pmatrix} x^E \\ x^{\bar{E}} \end{pmatrix} \qquad , \forall p \in \mathcal{U} \tag{4.6}$$

Or simply put:

$$\min_{x \in X, z} \alpha z + (1-\alpha) b(x) \tag{4.7}$$

$$\text{s.t.} \tag{4.8}$$

$$z \geq -p^\top x^E \qquad , \forall p \in \mathcal{U} \tag{4.9}$$

4.2 A Convex Formulation for the Robust Program

Proposition 4: Solving (4.7)-(4.8) is equivalent to solving the following problem:

$$\min_{x \in X} \left\{ \hat{f}(x) := \alpha \left(-p^\top x + \rho \sqrt{\sum_{(i,j) \in E} \left(\hat{p}_{ij} x_{ij} \right)^2} \right) + (1-\alpha) b(x) \right\} \tag{4.10}$$

Proof:

First, having enforced that

$$z \geq -\widetilde{p}^\top x^E, \forall \widetilde{p} \in \mathcal{U} \tag{4.11}$$

is equivalent to have

$$z \geq \sum_{(i,j) \in E} \left(-p_{ij} - u_{ij} \hat{p}_{ij} \right) x_{ij}, \forall u \in \psi \tag{4.12}$$

satisfied for any perturbation $u \in \psi$. Defining $v_{ij} = -\hat{p}_{ij} x_{ij}, \forall (i,j) \in E$ the latter system is equivalent to:

$$z \geq -p^\top x^E + \max_{||u||_2 \leq \rho} u^\top v \tag{4.13}$$

Since $u^\top v$ is the dot product the optimal solution lies in the direction of v with magnitude ρ. So

$$\pi \geq -p^\top x + \rho \frac{v^\top}{||v||_2} v \tag{4.14}$$

$$= -p^\top x + \rho ||v||_2 \tag{4.15}$$

$$= -p^\top x + \rho \sqrt{\sum_{(i,j) \in E} \left(\hat{p}_{ij} x_{ij} \right)^2} \tag{4.16}$$

Thus we can reformulate (4.7)-(4.8) as follows:

$$\min_{x \in X, z} \alpha z + (1-\alpha) b(x) \tag{4.17}$$

$$\text{s.t.} z \geq -p^\top x + \rho \sqrt{\sum_{(i,j) \in E} \left(\hat{p}_{ij} x_{ij} \right)^2} \tag{4.18}$$

Since we are minimizing and z is independent of the $x \in X$ constraints, at an optimal solution z will be tight with its lower bound, allowing us to replace z in the objective function with the lower bound, yielding the result. ■

The objective function of (4.10), $\hat{f}$, is again quadratic convex, and thus we solve again by the gradient descent method as described in the last section, with some modifications. First, the gradient of the objective for the edges $(i,j) \in E$ has a different expression:

$$\frac{\partial \hat{f}}{\partial x_{ij}}[x] = \alpha \left(-p_{ij} + \rho \frac{\hat{p}_{ij}^2 x_{ij}}{\sqrt{\sum_{(i,j)\in E} \left(\hat{p}_{ij} x_{ij}\right)^2}} \right), \forall (i,j) \in E \tag{4.19}$$

Additionally, $\hat{f}$ remains convex and differentiable and thus we are guaranteed a unique solution for the line search problem

$$\arg\min_{\lambda\in[0,1]} \{\hat{f}(x^{(k)} + \lambda d^{(k)}\} \tag{4.20}$$

The closed form of the solution to the linesearch is tedious. In consequence we find our optimal stepsize through a binary search.

5. RESULTS

The implementation of the algorithm detailed in the previous sections was written in C++, and all results were obtained on a computer with the following specifications: Intel(R) Core(TM) i7-4700HQ CPU @ 2.40 GHz x64 processor, 16.0 GB RAM. EU city populations were taken from Eurostat's database *Population on 1 January by age groups and sex - cities and greater cities.* In particular we considered three cases, where we took J to be all EU cities over 50,000, 100,000, and 200,000. EU political asylum application data was taken from Eurostat's database *Asylum and first time asylum applicants by citizenship, age and sex Annual aggregated data* from the year 2015. For our set of refugee origin countries, we took the six countries that originated the most asylum applications in 2015, including Syria, Afghanistan, and Iraq.

It is perhaps helpful to note that the quantitative values of our objectives are not easily interpretable, since they represent values for intangible objectives, preference maximization and the equity of burden distribution. As a result we will focus on relative changes in objective values as we present our quantitative results. Another challenge that presented itself was properly scaling terms in our objective function. Since the gradients of $b(x)$ are gradients of a variance function on a large set they tend to take on very small values, and thus must be scaled in order to compete with the larger preference terms.

5.1 Deterministic Version

5.1.1 Execution Time

We first examine our execution times for a selection of different values of α, n, and ε. The algorithm is fairly sensitive to its starting point, and choosing one randomly can severely impair the speed of the algorithm. In our deterministic formulation, we have a linear term which can be optimized quickly, and the quadratic balance term, which is significantly slower to optimize. Consequently we choose to set our starting flow to be a well-balanced feasible flow. As a result some of the behavior that we would expect out of a gradient

$\alpha = \mathbf{0.1}$	n		
ε	309	634	934
10^{-4}	1	20	62
10^{-8}	2	24	67
10^{-16}	2	26	69

$\alpha = \mathbf{0.5}$	n		
ε	309	634	934
10^{-4}	9	82	418
10^{-8}	11	83	418
10^{-16}	11	84	419

$\alpha = \mathbf{0.9}$	n		
ε	309	634	934
10^{-4}	159	1096	2845
10^{-8}	198	1113	2849
10^{-16}	197	1131	2844

Tab. 5.1: Execution times (seconds) of $f(x)$.

descent method does not surface. In particular our algorithm appears to be indifferent to the value of ϵ if the algorithm is started with a balanced, though not optimal, flow, as shown in (5.1).

5.1.2 Influence of Prioritization Parameter

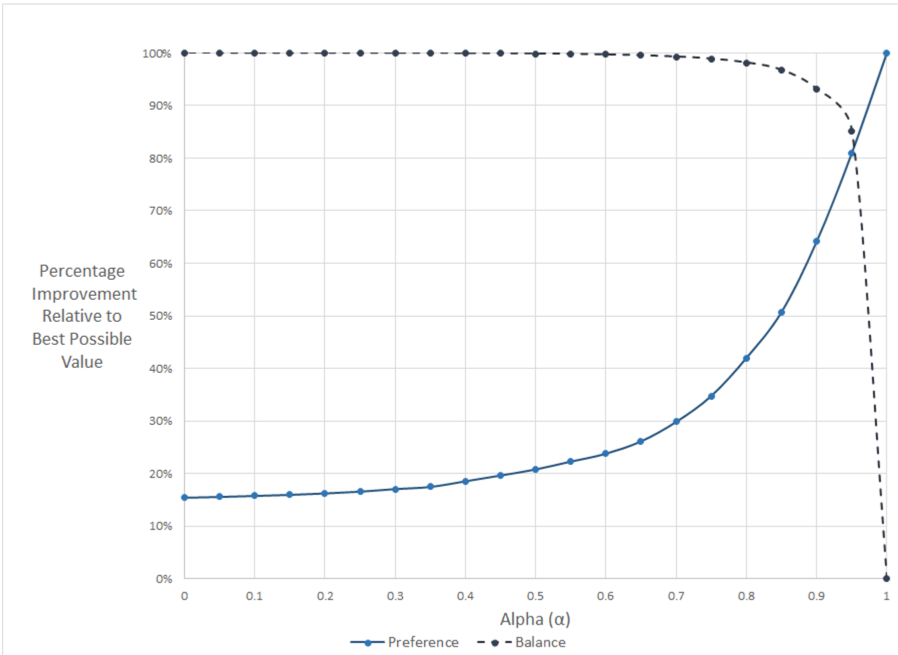

Fig. 5.1: Prioritization Parameter α's impact on each objective.

Next we examine the effect of our choice for the prioritization parameter α on the quality of solution obtained. Given $\varepsilon = .01$, we present our relative individual objective values as a function of α (Fig. 5.1). In particular, we are interested in how the optimal value of each objective at some α relates to its optimal value when we optimize only with respect to one objective.

As expected, choosing α close to 0 causes us to achieve a high quality solution with respect to balance, but does very poorly at satisfying refugee preferences. Similarly, choosing α close to 1 presents no opportunity of finding a balanced optimum. Recalling that we formulated this model as a tool for governments to mitigate burden imbalances, it is reasonable to assume that balance would hold higher priority than refugee preference. However preferences can be accommodated to some extent with very little penalty to the balance of our solution. Choosing $\alpha = 0.80$ results in a 25% improvement in our preference objective value, at the expense of less than 2% of our optimal balance objective value. The gains from improved refugee satisfaction may well be worth such a cost, since small imbalances in the burden of asylum of applicants will inevitably occur in any case from factors outside of CEAS jurisdiction.

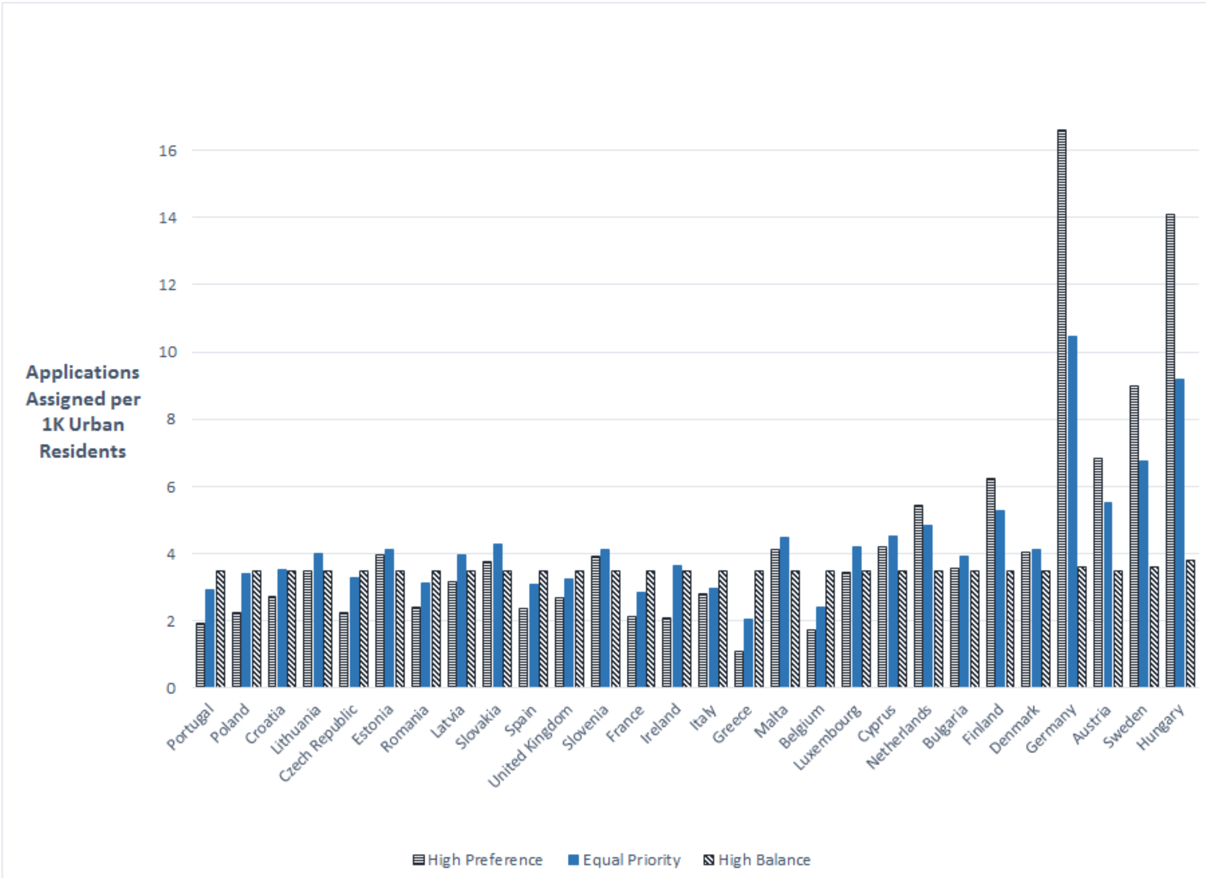

Fig. 5.2: Outcomes of Different Priority Values on Applicant Assignments

5.1.3 Comparison of Applicant Reassignment Outcomes

Though our model in fact assigns asylum applicants to specific cities, observing the aggregate assignments to each EU country is more practical for the purposes of our discussion. As a result, the figures of different refugee assignments present a national view, resulting from totaling the refugee assignments in each city of each country. In Fig. 5.2 we present the asylum applicant assignments resulting from three different priority values for α. High Preference, Equal Priority, and High Balance, corresponding to $\alpha = 0.9,\ 0.8,\ 0.1$, respectively. Hence the High Preference assignment will seek to maximize refugee preference, and the High Balance assignment will strive to achieve an even distribution of economic burden. The Equal Priority α value was chosen based off of our observation from Fig. 5.1 that In order to be consistent with our model, we defined national burden to be the total number of refugees assigned to any city in that country relative to the total population of those cities. Consequently the burden on less urbanized countries is doubtless somewhat overestimated. In Fig. 5.2, the nations are sorted from fewest to most asylum applications received in 2015 relative to their urban population. Hence though Germany received by far the most asylum applications in 2015, less populated countries like Hungary and Sweden, which each received over 100,000 applications (Fig.

5.2), are considered to be more heavily burdened. Table 5.2 at the end of the results section lists the total number of applications to each EU country in 2015 for reference.

Given that we defined refugee preference for a country to be the relative number of applications to that country from the refugee's country of origin, we expect that the High Preference assignment will allot large numbers of refugees to countries that already receive a large number of applications, and we see that this is indeed the case, though the balance term still has a significant impact on the assignment.

5.2 Robust Version

5.2.1 Interpreting Robustness

It is perhaps helpful to begin our discussion on the results of the robust model with this brief reminder: A robust program handles uncertainty by assuming that the unknowns will assume the most sub-optimal values possible, and attempts to find the best solution in that context. In our case, since it is the refugee preferences that we do not know for certain, the sub-optimal values the preferences would take on would be a shift in preference for countries that historically have low numbers of asylum applications. So we should expect our robust program to avoid sending too many applicants to "popular" countries, and to hedge its position by sending some applicants to countries with low application numbers. A robust formulation is particularly relevant in the context of asylum applications, since the problem is fundamentally a humanitarian one. While both refugees and governments stand to benefit from an improved asylum system, the human element must not be forgotten. Robust assignments help prepare governments for the profoundly human element of aiding refugees.

The hedging strategy that a robust program will employ is not without its downside. If the refugee preferences do not change significantly, or they continue to shift in the favor of popular countries, the robust program stands to lose by being unable to take advantage of such a favorable turn in events. It is this opportunity cost that we call the cost of uncertainty. In our formulation the size of ρ determined the type of outcomes we would hedge against. A small ρ would mean we expect only minor deviations from the status quo and thus will not incur a large opportunity cost should the status quo persist. It is worth noting that as α approaches 0, all things being equal, our solution behaves very similarly to the deterministic case, since the shrinking coefficient on the robust term eventually completely mitigates its impact on the objective function.

Figure 5.3 shows how this affects our own model. Consider the loss in the robust objective function $\hat{f}$ relative to the deterministic optimal value of f as a function of ρ for $\alpha = 0.50$ (left) and $\alpha = 0.90$ (right). When ρ is 0 our formulation reduces to the deterministic case, hence it serves as the basis for determining our opportunity cost, which is the percentage of the deterministic optimum we lose through a robust assignment. Fortunately one of the strengths of robust optimization is that often a respectable amount of uncertainty can be allowed while incurring a relatively low opportunity cost.

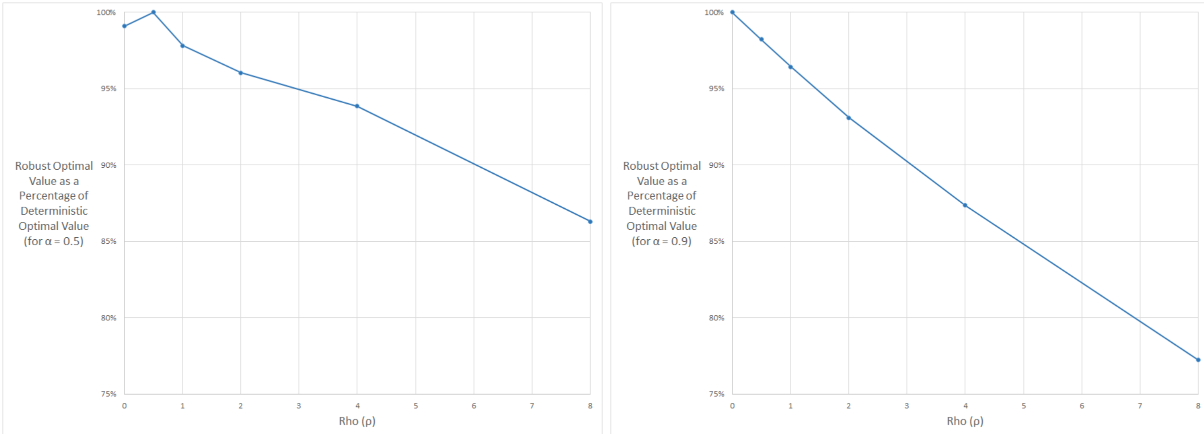

Fig. 5.3: The Cost of Uncertainty

5.2.2 Comparing Deterministic and Robust Reassignments

In Fig. 5.4 we can see the hedging in the robust assignment as it allocates fewer applicants to the popular countries than the deterministic assignment. Applicants are partially redirected away from countries with large numbers of applications towards countries near the bottom of the burden ranking like Portugal and Poland. We observe that because of how we constructed our model this diversification results in improved load balance, but it is not clear that a more refined measure of refugee preference would result in a similar effect. One the next page we provide a table of the number of asylum applications to each EU country in 2015 along with Fig. 5.4.

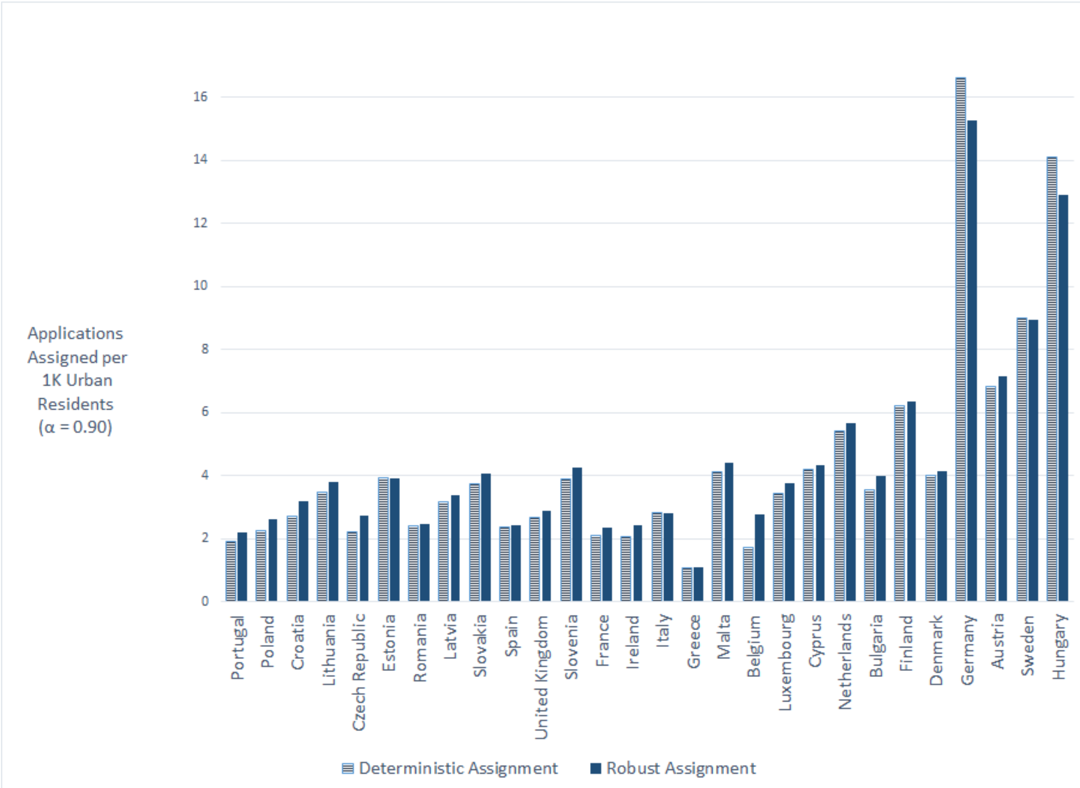

Fig. 5.4: Robust Hedging, $\alpha = 0.9, \ \rho = 1$

Germany	282325	Bulgaria	19695	Luxembourg	1685	Slovakia	220
Hungary	160255	France	17270	Ireland	1615	Slovenia	165
Sweden	123085	Italy	16450	Cyprus	1220	Latvia	130
Austria	69415	United Kingdom	15600	Romania	885	Portugal	115
Netherlands	33165	Denmark	14200	Malta	520	Lithuania	70
Belgium	29820	Greece	7855	Poland	410	Croatia	55
Finland	26835	Spain	6030	Czech Republic	220	Estonia	45

Tab. 5.2: 2015 Asylum Applications by Country

6. CONCLUSION

6.1 Future Work

To begin with, it must be admitted that in order for anything similar to this model to be implemented a great deal of improvement is needed. Of course better measures of asylum infrastructure capacity and refugee desire/capability to be reassigned are necessities. Structurally, the current formulation requires refugee preference to replace the role usually occupied by cost or travel distance, which certainly should be a factor when determining where to reassign an asylum applicant. Of course introducing the geographical element eliminates the bipartite graph structure. Fortunately we only require the graph to be acyclic, though on transportation networks this is certainly challenging enough.

We have another interesting consideration since this formulation can be applied to any acyclic graph in which we are trying to optimize travel cost while maintaining social equilibrium. This is exactly the sort of problem communicating self-driving cars would have to solve to determine which route each car should choose to transport it occupants. It is well understood in traffic routing that often each individual pursuing user equilibrium results in congested highways, i.e. poor load balance. Thus it would be desirable to have self-driving cars coordinate such that social equilibrium is achieved. However it is an inescapable fact that many consumers would still want their vehicle to take the most direct route. Thus we find ourselves faced with a very similar problem to what we have just considered. However it must be admitted that most transportation networks are extremely cyclic by design. Hence we would first have to determine how this formulation could be adapted to cyclic networks. Our current method of solving a Successive Shortest Path problem on a graph with gradient costs would be almost certain to encounter a negative cycle and fail. However if this could be overcome, the approach could be well suited to bi-objective routing of large numbers of cooperating vehicles.

Finally, this work has some similarities to semi-matching problems with load balancing such as those we have seen in (Harvey et al. 2006) and others. In particular in (Low 2006) we find a proof that semi-matching problems on weighted graphs with load balancing is NP-complete. Given unitary capacities the formulation presented in this work has the same structure, but with the relaxation of the integrality constraint of semi-matchings. As such, it is possible this work could have applications in an approximation algorithm for weighted semi-matchings with load balancing.

6.2 Closing Remarks

At the outset, we discussed that our objective was to explore how one could formulate as a quantifiable tractable problem the political challenge of determining when to redirect refugees, who to redirect, and where to redirect them. In this work, we proposed metrics of equity and preference that allowed us to define and solve an optimization problem whose optimal solution can serve as the basis of such administrative action. While this work was certainly inspired by the international challenges created by mass refugee movements, we have attempted to lay out a clear general methodology that can be applied outside of its original context. The model and algorithm we have discussed combine graph and network theory, convex optimization, and robust optimization to a common multiobjective goal.

REFERENCES

BBC News (2016). *Why is EU struggling with migrants and asylum?* http://www.bbc.com/news/world-europe-24583286. Accessed: 04/26/2017.

Ben-Tal, Aharon, Laurent El Ghaoui, and Arkadi Nemirovski (2009). *Robust optimization*. Princeton University Press.

Boyd, Stephen and Lieven Vandenberghe (2004). *Convex optimization*. Cambridge university press.

European Commission (2017). *Country responsible for asylum application (Dublin)*. https://ec.europa.eu/home-affairs/what-we-do/policies/asylum/examination-of-applicants_en. Accessed: 04/26/2017.

Harvey, Nicholas JA et al. (2006). "Semi-matchings for bipartite graphs and load balancing". In: *Journal of Algorithms* 59.1, pp. 53–78.

Low, Chor Ping (2006). "An approximation algorithm for the load-balanced semi-matching problem in weighted bipartite graphs". In: *Information Processing Letters* 100.4, pp. 154–161.

Open Society Foundations (2016). *Understanding Migration and Asylum in the European Union*. https://www.opensocietyfoundations.org/explainers/understanding-migration-and-asylum-european-union. Accessed: 04/26/2017.