PREDICTING FORCED DISPLACEMENT USING A GENERALISED AND AUTOMATED AGENT-BASED SIMULATION

A thesis submitted for the degree of Doctor of Philosophy

by

Diana Suleimenova

Department of Computer Science,

Brunel University London

January 2020

Abstract

Within the last decades, international migration demonstrated an escalating growth with more than 68 million people forcibly displaced worldwide. Forced displacement has a huge impact on society today as 40 million people internally displaced within their home country and 25.4 million refugees fled to neighbouring countries. Forcibly displaced people face several concerns, namely, the choice to stay or flee, the choice to flee internally or across borders, and the choice of destination. These decisions are often based on economic, social and/or political push and pull factors in origin and destination countries.

Existing international migration theories frequently cover specific aspects of migration processes, such as why human migration occurs and what effects it has on economies. However, the combination of all the factors and reasons for human movement requires expertise from various disciplines at once. Moreover, existing migration theories are not extensive enough for practical applications, statistical methods are outdated, and usually, do not account for forced population movements. To fulfil gaps within forced displacement predictions, we use computational models as they can contribute to a better understanding of forced displacement patterns and have potential due to their reduced ethical burden.

We propose a generalised simulation development approach (SDA) to predict forced population movements in conflict regions. Our SDA consists of a systematic set of phases to build agent-based simulations, which includes a generic model to define a real system problem, and simulation development and validation for situation-specific scenarios. We also synthesise data from UNHCR, ACLED and Bing Maps to build and validate agent-based simulations of three major African conflicts, namely Burundi, Central African Republic and Mali, and predict the distribution of incoming forced migrants across destination camps. Our simulations consistently predict more than 75% of the population arrivals in camps correctly after the first 12 days. Our agent-based simulation tool can help save migrants' lives by allowing governments and NGOs to conduct a better-informed allocation of humanitarian resources.

Few researchers have investigated the effects of policy decisions, such as camp capacity changes, camp and border closures and forced redirection, on forced population movements. To make such a study accurate and feasible in terms of human effort, we automate our generalised SDA by introducing and applying the FabFlee automation toolkit. We use our automated SDA to analyse the South Sudan crisis by incorporating two capacity changes to Adjumani camp, a border closure between South Sudan and Uganda, and forced redirection between Ethiopian camps. We find that a reduction in camp capacity induces up to 16% fewer forced population arrivals while an increase in camp capacity results in a limited increase in forced population arrivals ($< 4\%$) at the destination camps. In addition, border closure results in 40% fewer force population arrivals and an increasingly long travel journey to other camps. There is also a lingering effect in prolonged force population journey times once a border is again reopened and a clear boost in forced population arrivals when forced population are redirected to a reduced number of camps with larger capacities. To the best of our knowledge, we are the first to conduct such an investigation for forced displacement conflict situations.

Acknowledgements

It is my pleasure to acknowledge the role of several people who supported and guided me in completing my PhD course, which has been a truly life-changing experience for me.

First and foremost, I would like to express my sincerest gratitude to my principal supervisor, Dr Derek Groen, for introducing and encouraging to pursue the research area of modelling and simulation. His continuous support, motivation and guidance helped me to develop myself as a researcher and do research in the best possible way. I am thankful for his valuable suggestions, extended discussions and constant feedback to attain both scientific and programming expertise. I appreciate all the opportunities and challenges he has offered me to widen this research from various perspectives. It is an honour to be his first PhD student, and I could not have imagined having a better mentor.

Secondly, I would like to show my deepest appreciation to Dr David Bell for his insightful advice and positive criticism during this research. I want to extend my appreciation to the members of the progression review panel Professor Martin Sheppard, Dr Simon J. E. Taylor, Dr Alan Serrano-Rico and Dr Anastasia Anagnostou for their practical suggestions and helpful contributions.

I also have great pleasure in acknowledging my gratitude to my friends, PhD colleagues and fellow researchers in the Department of Computer Science for their unwavering support and insightful discussions throughout these three years.

Lastly, I am extremely grateful to my dad, mum and brother for their unconditional love and profound belief in my abilities to pursue a doctoral degree. I am deeply indebted to Roy Sudasinghage Don for his endless encouragement and motivation, who also played a decisive role in keeping me focused and inspired every day. This life opportunity would not have been possible without their support.

Declaration

I hereby declare that this thesis is the result of my own work and has been composed solely by myself except where stated otherwise. This research has not been submitted for any other professional qualification or degree.

The following papers are parts of this research that have been published or submitted for publication:

Suleimenova, D., Bell, D. and Groen, D. (2017), "A generalized simulation development approach for predicting refugee destinations". *Scientific Reports*, 7 (13377).

Suleimenova, D., Bell, D. and Groen, D. (2017), "Towards an automated framework for agent-based simulation of refugee movements". In Proceedings of the 50th Winter Simulation Conference (WSC17), edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page. IEEE, pp. 1240-1251, Las Vegas, Nevada, December 3-6.

Chan, N.T., Suleimenova, D., Bell, D. and Groen, D. (2018), "Modelling refugees escaping violent events: A feasibility study from an input data perspective". In Proceedings of the Operational Research Society 9th Simulation Workshop (SW18), edited by A. Anagnostou, R. Meskarian, and D. Robertson, pp. 156-163, Worcestershire, England, March 19-21.

Campos, C. V., Suleimenova, D. and Groen, D. (2019), "A Coupled Food Security and Refugee Movement Model for the South Sudan Conflict". In the 16th Multiscale Modelling and Simulation Workshop, International Conference on Computational Science 2019, Lecture Notes in Computer Science, edited by Rodrigues J. et al. (eds), 11540, pp. 725-732, Faro, Portugal, June 11-14.

Groen, D., Knap, J., Neumann, P., Suleimenova, D., Veen, L. and Leiter, K. (2019), "Mastering the scales: A survey on the benefits of multiscale computing software". *Philosophical Transactions of the Royal Society*, A: Mathematical, Physical and Engineering Sciences, 377, pp. 1-16.

Groen, D., Richardson, R. A., Wright, D. W., Jancauskas, V., Sinclair, R., Karlshoefer, P., Vassaux, M., Arabnejad, H., Piontek, T., Kopta, P., Bosak, B., Lakhlili, J., Hoe-

nen, O., Suleimenova, D., Edeling, W., Crommelin, D., Nikishova, A. and Coveney, P. V. (2019), “Introducing VECMAtk - verification, validation and uncertainty quantification for multiscale and HPC simulations”. International Conference on Computational Science 2019, Lecture Notes in Computer Science, edited by Rodrigues J. et al. (eds), 11539, pp. 479-492, Faro, Portugal, June 11-14.

Groen, D., Bell, D., Arabnejad, H., Suleimenova, D., Taylor, S. E. J. and Anagnostou, A. (2019), “Towards modelling the effect of evolving violence on forced migration”. In Proceedings of the 2019 Winter Simulation Conference (WSC19), edited by N. Mustafee, K. H. G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y. J. Son, National Harbor, Maryland, December 8-11.

Suleimenova, D. and Groen, D. (2020), “How Policy Decisions Affect Refugee Journeys in South Sudan: A Study Using Automated Ensemble Simulations”. *Journal of Artificial Societies and Social Simulation*, 23(1)2.

Suleimenova, D., Chopard, B. and Groen, D. (in preparation), “A formalized simulation development approach”.

Abbreviations

ABM	Agent-based Modelling
ABS	Agent-based Simulation
ACLED	Armed Conflict Location and Event Data Project
API	Application Programming Interface
CAR	Central African Republic
CNDD-FDD	National Council for the Defense of Democracy-Forces for the Defense of Democracy
CSS	Cascading Style Sheets
CSV	Comma-separated Values
DRC	Democratic Republic of Congo
DSL	Domain-specific Language
FabFlee	Fabric for Flee Simulation
FabSim	Fabric for Simulation
GeoJSON	Geographical JavaScript Object Notation
HiDALGO	High-performance Computing and Big Data Technologies for Global Systems
HLA	High Level Architecture Agent
HPC	High-performance Computing
IDPs	Internally Displaced Persons
JADE	Java Agent DEvelopment Framework
JSON	JavaScript Object Notation
MASE	Mean Absolute Scaled Error
MASON	Multi-Agent Simulator of Neighbourhoods (or Networks)

Modgen	Model Generator
MSF	Medecins Sans Frontieres
NGO	Non-governmental Organisation
QCG	Quality in Cloud and Grid
RC	Republic of Congo
Repast	Recursive Porous Agent Simulation Toolkit
REST API	Representational State Transfer Application Programming Interface
SAROBMED	Search and Rescue Observatory for the Mediterranean
SDA	Simulation Development Approach
SPLA	Sudan People's Liberation Army
SPLM	Sudan Peoples' Liberation Movement
SPLM/A	South Sudan People Liberation Movement in Opposition
SSH	Secure Shell
UNHCR	United Nations High Commissioner for Refugees
USCR	United States Committee for Refugees
VECMA	Verified Exascale Computing for Multiscale Applications
VVUQ	Verification, Validation and Uncertainty Quantification
XLS	eXceL Spreadsheet
YaML	YaML Ain't Markup Language

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Background and research problem statement

Human migration is a global phenomenon with a long history. According to Fleagle et al. (2010), earliest human movements began almost two million years ago with Hominins from Africa occupying Eurasian continent, and migration of African people to the Mediterranean and the Arabian lands occurred 120,000 to 90,000 years ago. Climate change, particularly the Ice Ages, caused humans to disperse to other mainlands (deMenocal and Stringer, 2016). The centuries after migration was proceeded by the pre-modern movement of people in the years of the Renaissance, and strengthened by colonisation, revolutionary transitions, and globalisation (Wickramasinghe and Wimalaratana, 2016).

To understand human migration today, we categorise it in terms of size, time, lawful status, territory, type and cause (see Figure 1.1). A *size* of the movement is a category comprising individual, group or mass migration. The *time* category recognises migration in terms of temporary or permanent movement. The *lawful status* of migration, which is defined by government authorities, indicates whether a migrant is legal or illegal in the destination country. Under the category of *territory*, migration splits into internal, as in dispersal of people inside the country of origin, and international, as in dispersal outside the country. These territory divisions are essential in discerning *types* of migrants. For example, people who migrate voluntarily are considered to be migrants. People who migrate involuntarily are identified as forced migrants. Researchers use the term 'involuntary' interchangeably with irregular or forced displacement. However, Edwards (2016) stress to distinct migration from forced population displacement. Throughout this thesis, we use the term 'forced displacement' when referring to involuntary or forced population movements.

There are three types of forced migrants, namely oustees, internally displaced persons and refugees. Voutira (1997) defined each type as follows: oustees refers to people who are

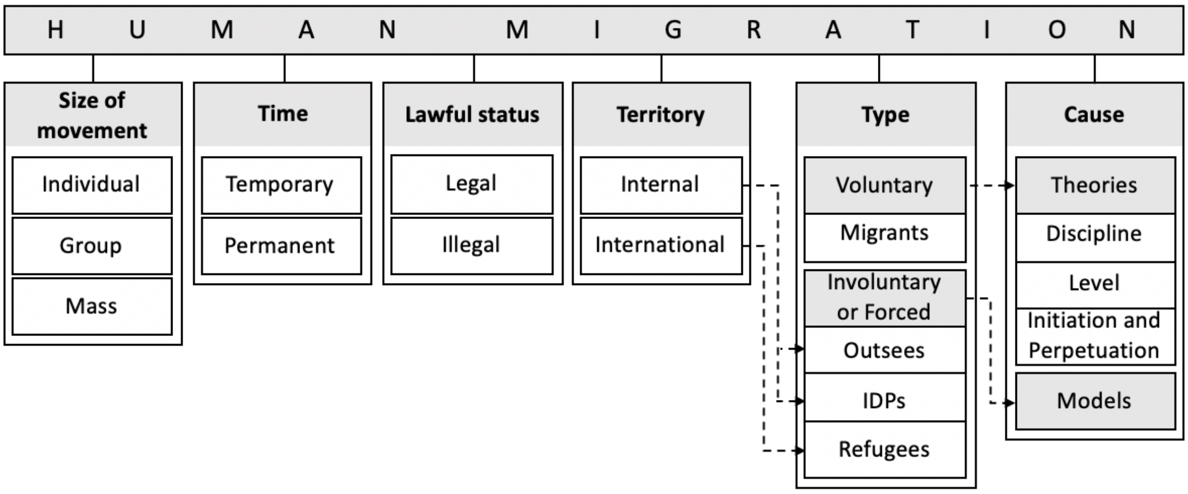

Figure 1.1: Overview of human migration categorisation by size of movement, time, lawful status, territory and cause derived from migration types (Faist, 2000).

permanently and internally forcibly displaced due to natural preservation and governmental actions to improve the standard of living or capital-intensive projects; the internally displaced people (IDPs) are internally dispersed within a country of residence due to famine, violence, armed conflict, war or other extreme situations; refugees migrate for the same reasons as IDPs, but do so internationally meaning that they are displaced across borders of their home country. According to the 1951 Refugee Convention, a refugee is "someone who is unable or unwilling to return to their country of origin owing to a well-founded fear of being persecuted for reasons of race, religion, nationality, membership of a particular social group, or political opinion" (UNHCR, 2010, p. 3).

The most examined human migration category in literature is *causes* that define migration theories and forced displacement models from voluntary and involuntary conditions driving people to migrate or displace from their origin country (Faist, 2000). Migrants choose to move voluntarily, for example, for better economic opportunities or living standards. While forced displacement makes people vulnerable and leads them to displace, in search for a secure and stable location. Hence, people who are forced to flee, namely forced migrants or forced population, have different and challenging experiences compare to migrants.

In 2018, 68.5 million people were forcibly displaced worldwide, a number which includes 25.4 million refugees and 40 million IDPs (UNHCR, 2018). An alarming increase in forced displacement counts is mostly due to violence, armed conflict and war. There are also sub-national causal factors that explain forced population displacement, including ethnic or religious differences and existential obstacles such as severe economic or ecological decline (Wood, 1994;

Kalipeni and Oppong, 1998).

Researchers establish three concerns faced by forcibly displaced people, namely, the choice to stay or flee, the choice to flee internally or across borders, and the choice of destination (Salehyan, 2014). Their decisions are often based on economic and political push and pull factors in origin and destination countries. Especially, Schmeidl (1997) states that economic and political instabilities, poverty, violence and insecurity in the origin countries push people to flee. In contrary, economically and politically stable and safe countries pull forced population to their destination areas. Thus, we can consider the economic and political conditions, security, the challenges and expenses of moving internally or across borders as causes of forced displacement.

Unfortunately, forced displacement has enduring consequences on population, as well as on both origin and destination countries. For instance, civil war and violence within the origin countries may spread across borders. Similarly, destination nations may interfere in internal conflicts and wars occurring in origin countries to prevent further increasing forced population arrivals (Gleditsch et al., 2008). In addition, destination countries may face external costs by hosting forcibly displaced people as residents and have to share available resources. Some countries may not have enough support and required humanitarian aid in terms of shelter, food and safety. Hence, hosting forced population can have positive or negative consequences (Martin, 2005).

According to Jacobsen (1996, p. 674), some governments base their decision whether to host or refuse forced population arrivals on "the costs and benefits of accepting international assistance, relations with the origin country, political calculations about the local community's absorption capacity, and national security considerations". Policy decisions on the basis of human rights, economic, political and humanitarian factors for both origin and destination governments, and non-governmental organisations (NGOs) can manage, resolve and overcome the consequences of forced displacement. They can also facilitate an efficient allocation of human resources required for a forced population in camps. However, the literature lacks in identifying effective policies to assist and overcome forced displacement. Moreover, it is seldom clear how policy decisions affect displaced people' journeys and camp arrival rates, particularly those in other countries.

1.2 Research motivation

Researchers have mostly investigated why human migration occurs and what effects it has on economies using migration theories and econometric models. Today, however, these theories are not extensive enough for practical applications, statistical methods are outdated, and models are inappropriate for forecasting forced population counts (Edwards, 2008; Disney et al., 2015). For instance, many early warning models ignore predictions of forced population movements. They also lack the accuracy and flexibility to accommodate the context changes that lead to large-scale forced displacement (Lopez-Lucia, 2015). In turn, there is not an appropriate method or model to predict forced population movements, which is a decisive gap in the research area.

Forecasting forced displacement is crucial since global displacement has reached record levels. It is also challenging as many forced population data sets are small and incomplete, and data sources have too little information. Yet, forced population predictions are essential to save forced migrants lives, to investigate the consequences of a nation closing its border for forced population, and to help complete incomplete data collections on forced population movements. Improvements in data collection may be a possible solution to overcome data issues, but we require an enhanced logical framework to capture forced displacement thoroughly.

The use of computational models can contribute to a better understanding of forced displacement patterns. Particularly, they have potential due to their reduced ethical burden, which generally impedes empirical analysis, to help governments and NGOs to conduct a better-informed allocation of humanitarian resources. There is also the possibility to derive causal relations between forced displacement and policy decisions, such as camp capacity changes, camp and border closures, and forced redirection. Importantly, forced displacement simulation can assist governments and NGOs in estimating where and when forced migrants are likely to arrive, and which camps are most likely to become full in the short term. Hence, forced displacement simulation can be vital for informing, predicting and fulfilling gaps within forced displacement predictions.

There is also a prediction urgency of displacement crises when we simulate multiple conflict scenarios or conflicts that occur in a short time period. To address this urgency, stakeholders require a generalised simulation development approach, which involves the selection of data sources, the extraction and conversion of data, construction of an initial model and execution of simulations, as well as the validation of simulation predictions against empirical data.

In addition, an automated simulation approach can constitute as an essential step for creating rapid, consistent and efficient forced population arrival predictions within days of a new conflict eruption. We can systematically investigate the effect of policy decisions and other counterfactual outcomes using an automated simulation development approach. Since many scenarios need to be constructed and analysed, and manual simulation development is simply too labour-intensive.

1.3 Research aim and objectives

The aim of this research is:

"To develop a generalised and automated simulation development approach predicting forced population movements in conflict regions, which will enable the 'virtual implementation' of policy decisions allowing governments and NGOs to conduct a better-informed allocation of humanitarian resources".

The objectives of this research are to:

1. Review human migration theories and forced displacement models to obtain an understanding of the state-of-the-art prediction methods.
2. Develop a computational simulation technique for forced population displacement.
3. Implement the developed approach into a simulation tool that predicts forced population displacement.
4. Evaluate the tool and test the validity of results by comparing simulation output against empirical data.
5. Automate the simulation tool to reduce inefficiency.
6. Run the automated tool, validate its accuracy on real conflict situation and its ability to incorporate policy decisions.

1.4 Research methodology

We develop a generalised simulation development approach (SDA) based on an agent-based model (ABM) to simulate the distribution of incoming forced population across destination camps forced to flee because of war, armed conflict and/or political instability. ABM is a very popular computational approach in social sciences (Castle and Crooks, 2006; Crooks et al., 2008). Its popularity is in part due to the decentralised nature of the approach, which allows a heterogeneous mix of many agents to act and interact autonomously, in turn leading to emergent behaviours in the system at higher levels. ABM consists of agents interacting within an environment, as illustrated in Figure 1.2. Especially, ABM is suitable for modelling active objects, such as individuals, animals or products, in relation to time, event or behaviour (Borshchev and Filippov, 2004). It has been applied to model problems ranging from small-scale behavioural dynamics to large scale migration simulations (Macal and North, 2010).

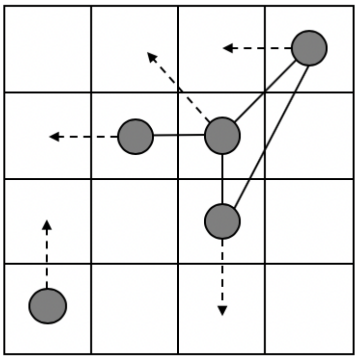

Figure 1.2: Schematic representation of ABM elements on a grid, including agents and the environment, where straight lines are agent-to-agent interactions and dotted lines are agents' interactions within their environment.

We propose a generalised SDA for situation-specific scenarios to fulfil the demand for generic simulation development using the existing simulation processes and technical expertise. Our generalised SDA involves two main step-by-step processes, namely a generic model, which has one-time construction phases, and simulation development and validation that applies to individual situation specific scenarios. The emergence of publicly and thoroughly curated data repositories of the last decade, such as the United Nations High Commissioner for Refugees (UNHCR, data2.unhcr.org), the Armed Conflict Location and Event Data Project (ACLED, acleddata.com) and the Bing Maps (bing.com/maps), enables us to reconstruct conflict situations with unprecedented accuracy. They also provide us with the empirical data to validate and visualise our simulation results against the UNHCR data. We exclude IDPs from the model, as there is a lack of systematic data providing their exact destinations in conflict scenarios.

1.5 Scientific importance

In this thesis, we present generalised forced displacement simulations that are designed to (a) enhance conflict situation awareness using multiple data sources, (b) provide predictions on forced displacement patterns to aid policy decisions, and (c) enable 'what-if' scenario analysis for governments and NGOs looking to study the impact of forced displacement. We are also the first to predict the distribution of forced population arrivals to potential destinations across four conflicts, so governments and NGOs can efficiently allocate humanitarian resources and provide protection to vulnerable people.

We develop an SDA to construct and execute forced displacement simulations that can be applied to other migration situations. Through the use of computational modelling, we are able to systematically explore and predict the possible impact of conflict scenarios forcing people to flee. Moreover, we automate our SDA using Fabric for FLEE Simulation (FabFlee) toolkit, which is a combination of Fabric for Simulation (FabSim) toolkit and the FLEE simulation code. It is rapid, consistent, efficient, and saves efforts in developing forced displacement simulations. Hence, using a highly transparent and customised approach, we can automate key tasks, including the creation, execution and analysis of models. Our implementation also provides a platform to run ensemble simulations for parameter explorations, predict alternative conflict scenarios, assess the effects of different camp allocations of forced population, and account for the sensitivity to several of the individual parameters and assumptions in the model.

1.6 Thesis structure

This thesis has eight chapters, where each chapter focuses on six outlined objectives, respectively (see Figure 1.3). To achieve the aim of this research, we organise this thesis as follows: **Chapter 2** reviews existing literature on migration theories, forced displacement models and prediction techniques to formulate the research problem and justify the necessity for developing a prediction approach for forced population displacement. This chapter also introduces the computational modelling techniques, such as the gravity model, system dynamics and Markov chain, as well as ABM, which is the main modelling tool of this thesis. We provide a thorough understanding of ABM and examine its application to migration and forced displacement studies.

Chapter 3 discusses existing simulation development processes and identifies the main requirements for agent-based simulation (ABS) development. In this chapter, we propose our methodology for developing situation-specific simulations including forecasting approach for forced population displacement.

Chapter 4 presents forced displacement simulation development approach to predict the distribution of incoming forced population across destination camps. This chapter extensively describes each phase of our SDA, namely problem selection, data collection, model construction, model refinement, simulation execution and analysis.

Chapter 5 focuses on three forced displacement crises in African countries. Here, we apply our generalised SDA to model forced population movements in Burundi, Central African Republic and Northern Mali. This chapter also presents simulation results for each conflict, analyses and validates the outcome against the UNHCR data, and reproduces 75% of the forced population movement destinations.

Chapter 6 assesses existing automation tools and techniques in application to ABS development. In this chapter, we develop an automated SDA for forced displacement modelling to facilitate the rapid, consistent and efficient development process. Notably, we automate data collection, model construction, refinement, simulation execution and analysis simulating forced population movements quickly, and in a short time notice.

Chapter 7 investigates policy decisions, such as camp capacity changes, camp and border closures, and forced redirection, affecting forced population movements. Here, we apply a generalised and automated SDA to model an on-going forced displacement conflict of South Sudan and investigate how each of the policy decisions impacts the distribution of incoming forced population across neighbouring camps.

Chapter 8 summarises each chapter, presents the significance of this thesis through theoretical and practical contributions, discusses the research limitations and provides future research directions.

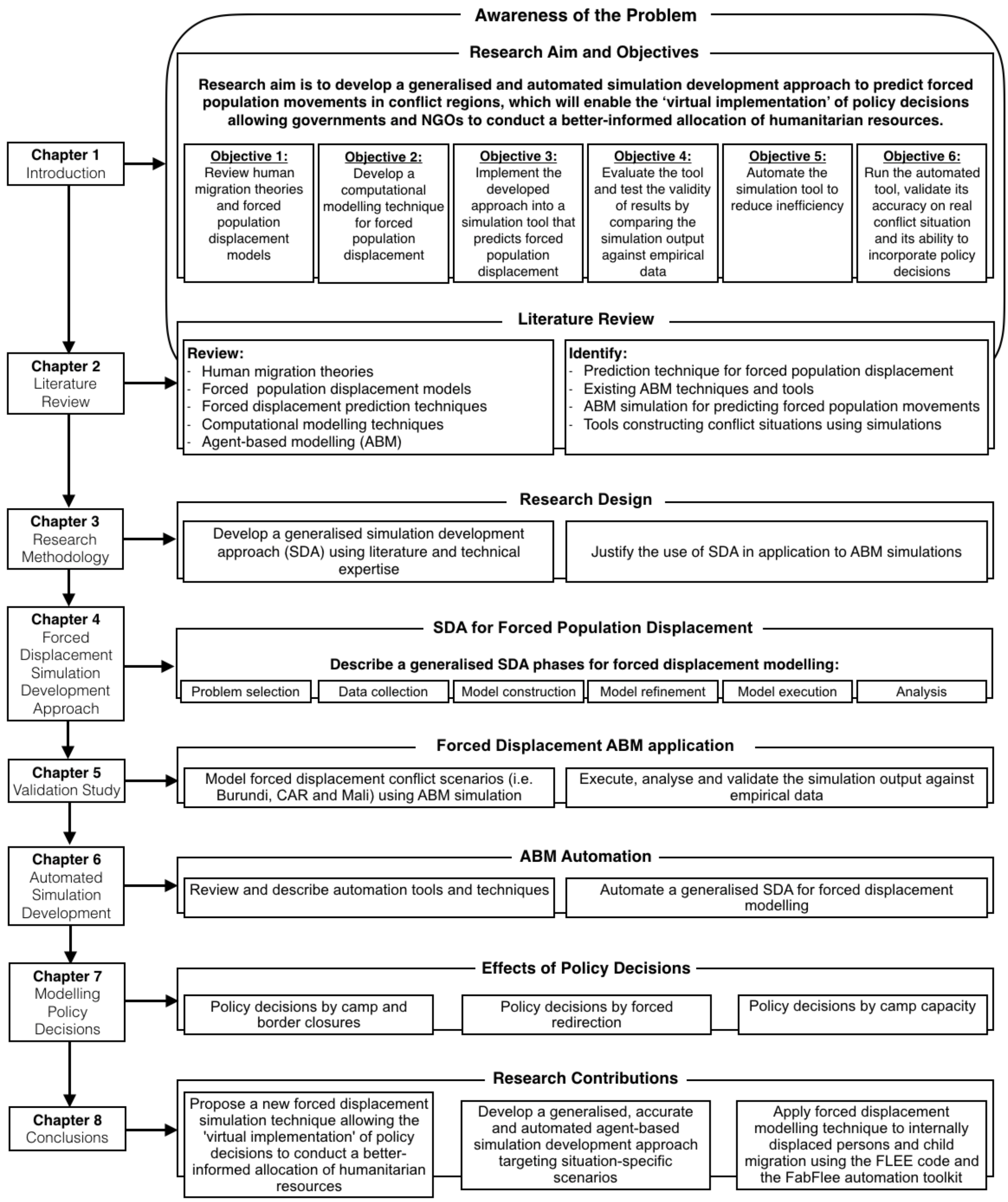

Figure 1.3: Thesis outline.

Chapter 2. Literature Review

2.1 Introduction

This chapter investigates state-of-the-art of human migration. First, we explore migration literature from various disciplines and establish a theoretical background to understand the core concepts of migration. Second, we determine existing migration theories and forced displacement models to explain why migration and forced displacement occur, as well as what effect they have on economies. We also define that existing econometric methods and early warning models have a decisive gap in the research area of forced displacement predictions. To fulfil it, we examine computational modelling techniques, in particular, agent-based modelling in application to human behaviour and movement.

2.2 Migration theories

Since the 1950s, literature on migration has mainly explored the voluntary movement of people (Hagen-Zanker, 2008). The main focus was to understand why human migration occurs and what effects it has on economies. Consequently, there is a vast number of migration theories. To determine their roles in explaining migration, firstly, we introduce theories distinguished across various disciplines. Secondly, we explore other categorisations of theories from perspectives of causation, application and analysis of migration. Finally, we follow the classifications of migration theories by the inclusion of forced population displacement.

2.2.1 Disciplinary migration theories

To start with, Bijak (2006) surveys migration theories developed by researchers of distinct disciplines, such as sociology, economics, geography and combined fields (or unifying). He offers an overview of theories across disciplines of science demonstrated in Figure 2.1.

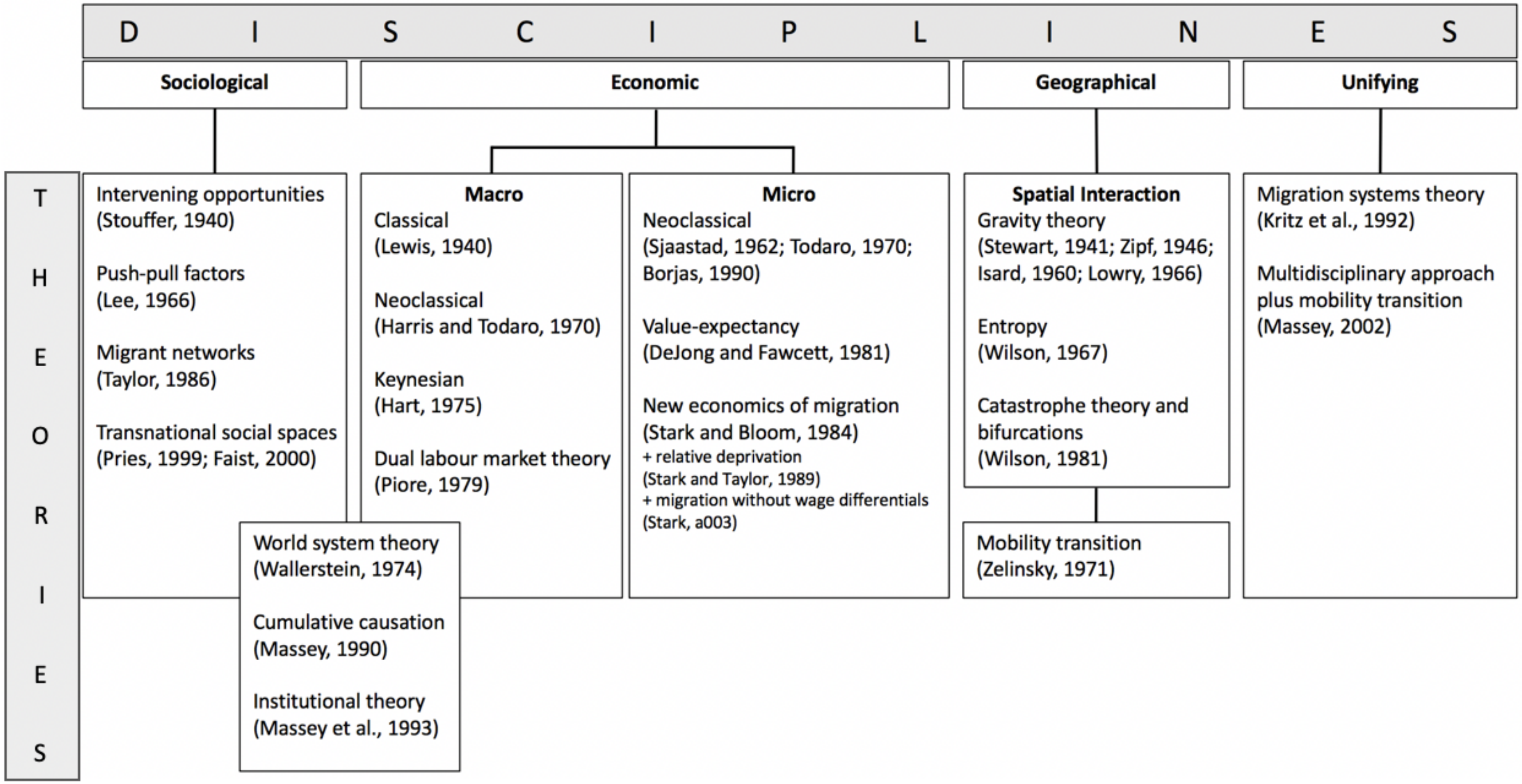

Figure 2.1: Migration theories across several disciplines (Bijak, 2006).

Each of these disciplines includes migration theories established over the years. First, sociology features the concept of intervening opportunities as the initiator for migration theories. For instance, Stouffer (1940) found that attracting opportunities, such as jobs, increase migration at the destination location with a condition of an opportunity being further from the home country. Lee (1966) proposed the theory of opportunities and suggested that push-pull factors determine migration. The push factors from the origin country forcing migrants to migrate, and the pull factors attracting to the destination country are given in Figure 2.2. The intervening obstacles between the origin and destination indicate the travel of the relocation. Major push-pull factors include poverty, unemployment, environmental disasters, armed conflicts and violence. As a result, Lee's push-pull factors can be used to represent social, political, and economic conditions of migrants' origin and potential destination (Moore and Shellman, 2007). Huzdik (2014) argues that globalisation and technological advancements have changed how people migrate or displace in the 21st century, but push-pull factors still explain and rationalise population mobility.

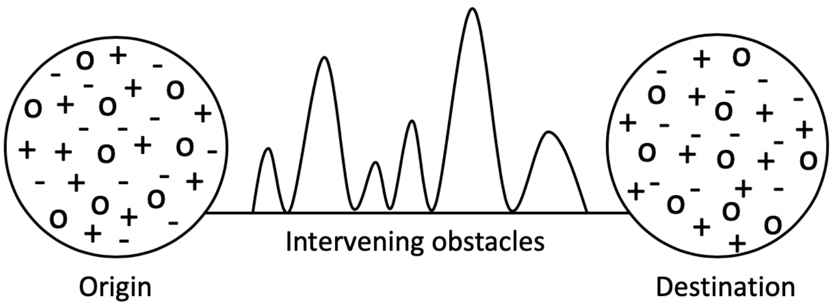

Figure 2.2: Origin and destination factors with intervening obstacles (Lee, 1966).

Boano et al. (2003) provides various push-pull factors in relation to forced displacement, shown in Table 2.1, repurposing the concepts from Wood (1994) in an international context. Moore and Shellman (2004) find that the violent actions of governments are also push factors forcing people to migrate. Besides, pull factors may influence migrants and refugees to flee as social, political and economic circumstances of neighbouring countries may be superior (Moore, 2006).

Migration networks theory focuses on the reasons behind pull factors. For example, Taylor (1984) suggested that social relationships, such as family, friends or acquaintance, at the destination country may increase migration rates, due to features of networks that lessen psychological, financial and other risks related to migration. The concept of networks is further expanded in the transnational social spaces theory of migration, which stresses the importance

Push factors (Origin country)	**Pull factors (Destination country)**
War	Personal capital:
Persecution	Social networks
Genocide	Presence of the family
Abuse of human rights	Required profession
Bad economic conditions	
Poverty	External capital:
Famine	Less violence
Natural hazards	Good asylum policy
Development projects	Positive attitudes towards refugees
	Existence of refugee settlements
	Better economic situations
	Political stabilities

Table 2.1: Push-pull factors of forced displacement (Boano et al., 2003).

of social and symbolic cross-border interconnections amongst individuals and groups of people migrating internationally (Faist, 2000). This theory emphasises that neighbouring countries with different social and symbolic norms observe lower migration rates, as these norms might not be transferable across borders.

We can distinguish the economic migration theories, the second discipline in Figure 2.1, into two perspectives: macro and micro. The macroeconomic migration theories are organised into classical, neoclassical and Keynesian schools of thought. In the 1940s, economist Arthur Lewis had the initial contribution to classical macroeconomic theory emphasising the structural change of economies focusing on growth and development that prompted urban-rural migration. Subsequently, in the 1970s, neoclassical macroeconomic approach from Harris and Todaro explained the flow of people between countries created by wage differentials in labour market. Later, Keynesian economics argued that migration also occurs due to unemployment and economic imbalances. Another macroeconomic theory that includes migration is the dual labour market theory, which describes demand characteristics of labour markets, targeted by migrants at their potential destination. It suggests that people move internally to find more attractive jobs, whereas people displaced from other countries take jobs that are difficult and dangerous (Bijak, 2006).

Several other migration theories in sociological discipline also link with macroeconomics (see Figure 2.1). The world system theory proposed by Wallerstein in 1974 considers world structural changes and economic emergence as drivers of migration. Notably, it finds that capital mobility influences migration decisions due to the expansion of agricultural and manufacturing exports amongst economies (Kurekova, 2011). Massey et al. (1993, p. 451) define a cumulative causation theory, where "...each act of migration alters the social context within which subsequent migration decisions are made, typically in ways that make additional movement more likely". Indeed, it proposes that migration generates socio-economic changes, such as an aggregation of social capital in origin and destination countries, which in turn creates a new movement of people over time (Massey, 1990). They also suggest that institutions and organisation ease migration processes through legal and illegal practices of migrants recruitment, or international human trafficking.

The microeconomic migration theories focus on the individual decisions of migrants. There are three main theories of neoclassical, value-expectancy and the new economics of migration. The neoclassical microeconomics theory by Sjaastad (1962) implied that the movement of people is a contribution to human capital and a consequence of cost-benefit analysis. Other economists further expanded the neoclassical theory and formalised a framework in which migrants' expected future income after various costs of migration maximises their choice of a destination. Moreover, the value-expectancy proposed by De Jong and Fawcett (1981) considers the individual motivation to migrate dependent on the favoured outcomes and expectations of migration. However, the new economics of migration challenged previous microeconomics theories and related migration decisions to households, such as family patterns, but not to the individual level of decision-making (Bijak, 2006). Overall, within microeconomic migration theories, the decision to migrate determined by individuals and households, as well as intentions of migrants to maximising income or minimising cost while moving to destinations.

The third discipline explaining migration in Figure 2.1 is geographical theories. Within this discipline, the aspect of distance that people traverse is important as it defines spatial movements between an origin and potential destination. Henceforth, it can be stated that spatial interactions explain geographical migration theories. There are several theories, namely gravity, entropy, and catastrophe and bifurcations. Each of these has specific equations or formulas determining population movements that are described in detail in Bijak (2006). Despite their different representations, distance travelled, and mass of flow are two common factors. They can be measured using time, transport routes and transport prices for distance and economic factors of employment, or income for a mass of migration.

The mobility transition is an additional geographical theory which focuses on human mobility from the perspective of demographic transition. It examines how changes, such as industrialisation, modernisation and recent communication advancements, enhanced the movement of people worldwide. Forecasting human migration based on this theory is difficult and limited due to the lack of data on mobility types and any means of communication used for migration purposes.

Unifying migration theory attempts to combine migration system theory with mobility transition. The system theory involves historical, economic, political and cultural factors that force people to migrate, while mobility transition unifies sociological, economic, political and psychological reasons for migration, as well as the continuity of stay of migrants. According to Bijak (2006), these unified theories are generally too complex to apply in practice and to explain migration.

2.2.2 Migration theories from other perspectives

A considerable amount of literature has focused on classifying migration theories in terms of level analysis. Three main levels are compatible with each other, namely micro, meso and macro. Micro-level refers to individual values, desires and expectations of migrants to migrate. Collectives and social networks define meso-level with related migration theories, while macro-level demonstrates migration theories that caused by economic structures (Hagen-Zanker, 2008). In Table 2.2, we illustrate how migration theories can be evaluated in terms of these levels. There are conventional theory allocations between economic discipline and level-based classifications, particularly for micro and macro levels. Some theories from other disciplines are also featured in this level-based analysis.

	Migration cause	**Migration theories**	**Sources**
Micro-level	Individual desires, values, expectancies e.g. improving survival, wealth etc.	Push-pull factors	Lee (1966)
		Neoclassical micro-migration	Sjaastad (1962); Fisher et al. (1997)
		Behavioural models	Wolpert (1965); Ritchey (1976); deJong and Fawcett (1981)
		Theory of social systems	Noffmann-Novotny (1981)
Meso-level	Collectives, social networks e.g. social ties	Social capital theory	Massey et al. (1998); Massey (1990)
		Institutional theory	Goss and Lindquist (1995); Guilomoto and Sandron (2001)
		Cumulative causation	Massey (1990)
		New Economics of Labour migration	Stark (1980); Taylor (1986); Stark and Lucas (1988)
Macro-level	Opportunity structure e.g. economic structure (income and employment opportunity differentials	Neoclassical macro-migration	Lewis (1954); Ranis and Fei (1961); Todaro (1969); Todaro and Harris (1970)
		Migration as a system	Mabogunje (1970); Kritz and Zlotnik (1992)
		Dual labour market theory	Piore (1979)
		World system theory	Wallerstein (1974)
		Mobility transition	Zelinsky (1971)

Table 2.2: Migration theories by level analysis (Hagen-Zanker, 2008).

Boswell (2002) analyses micro, meso and macro levels for forced population displacement. He states that at the micro level, forcibly displaced people consider costs, which involve the psychological and financial contributions for fleeing to another country, and benefits, such as personal and family safely. At the meso level, collective and network theories can assist in understanding the destination choice of a refugee as social ties or relationships may allow them to gather more information about safer location, get protection or have some interest to a particular destination. The macro level factors include government repressions, violence or civil war terrors that people experience in their home country.

Another perspective divides theories into categories of initiation and perpetuation of migration, which is presented in Table 2.3. The former combines migration theories that determine causes, while the latter comprises theories providing explanations on the continuity of migration, which is a new domain. This division allows us to understand migration rule sets by applying theories, respectively.

Initiation of migration	**Perpetuation of migration**
Neoclassical macro-migration theory	Migration as system
Neoclassical micro-migration theory	World system theory
Migration as system	Social capital theory
World system theory	Institutional theory
Dual labour market theory	Network theory
Push-pull factors	Cumulative causation theory
Behavioural models	
Theory of social systems	
New economics of labour migration	

Table 2.3: Classification of migration theories based on initiation, which explains migration causes, and perpetuation, which defines different factors from the start to continuation of population dispersal (Hagen-Zanker, 2008).

Overall, we find that researchers use disciplinary theories to investigate human decisions to migrate and to cover specific aspects of the migration process. However, there is no single theory comprising all the factors and reasons for migration. The use of various theories at once could be a new perspective for studies. In the next section, we examine alternative models that specifically focus on forced population displacement.

2.3 Forced population displacement models

Currently, there are more than 68 million forcibly displaced people worldwide, of which 40 million are IDPs, and over 25 million are refugees (UNHCR, 2018). A comprehensive understanding of human migration provides background knowledge for exploring and understanding the recent increase in forced displacement. There are various and often complex reasons behind the decision of people to move, but motivation, desire and pressure are key in most cases (Boano et al., 2003).

Compared to other sub-fields, we found relatively few research works on forced displacement. Kunz (1973) and Richmond (1988) were among the first to explore and publish studies on forced displacement. Kunz (1973) identified that movement could be by force, by flight or by absence, while Richmond (1988) gave an initial motivation to explore the scenario of population displacement in appliance with migration theories. Wood (1994) identified the classification of internal national and external forces, as well as sub-national causal factors that lead to forced displacement, which we show in Figure 2.3. The sub-national causal factors have three sets including war, persecution and political instability in the first set; ethnic, religious and tribal conflicts in the second set; and ecological crises and life-threatening economic decline in the last set.

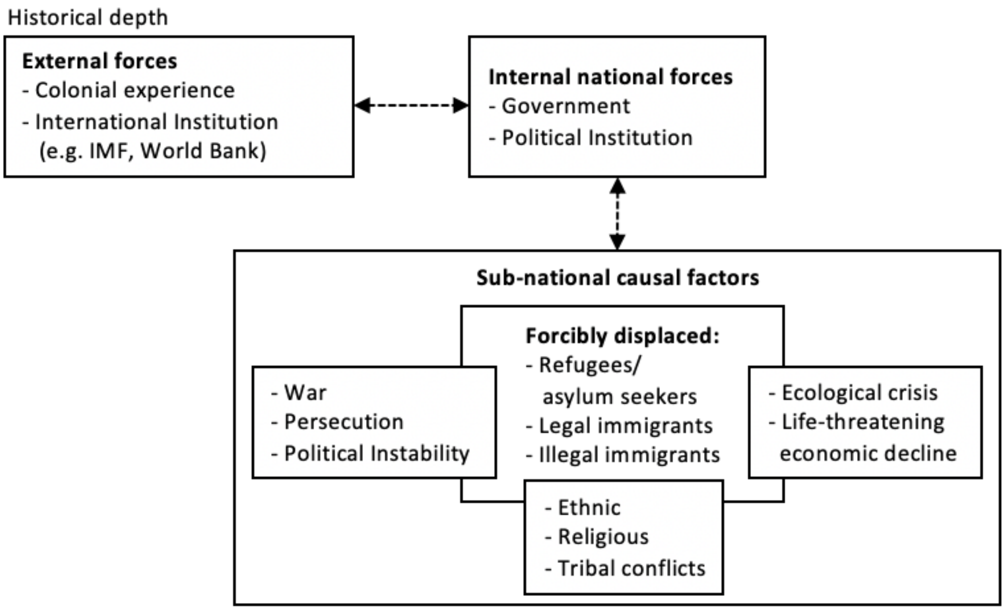

Figure 2.3: Model of forced displacement (Wood, 1994).

Similarly, Davenport et al. (2003) conclude that forced displacement occurs due to internal or external conditions that trigger the movement of people, which are listed in Table 2.4. In this division, internal conditions include domestic threats of violence, famine and natural disasters, while external comprises colonialism, inequality and deterioration of the environment. These factors overlap in forcing people to displace internally and internationally. Shami (1996) argued that mostly war, disaster or development factors often force people to flee from their home country. Although the most recognised one in the literature is violent conflicts (Schmeidl, 1997; Neumayer, 2005), there are not any studies focusing on the type of conflict that has the most impact on displacement (Brück et al., 2018).

Internal conditions	External conditions
Violence	Colonialism
Worsening of socio-economic conditions	Unfair trade regulations
Famine	Global inequality
Natural disasters	Impact of corporations on local economies
	Deterioration of the environment

Table 2.4: Internal and external conditions explaining forced displacement (Davenport et al., 2003).

2.4 Forced displacement prediction techniques

The previous section has shown that there are various migration theories and forced displacement models, which are generally not extensive enough for practical applications. However, researchers do use these theories and models to identify the determinants of migration, to explore the migration consequences, to understand the experiences of forced migrants at the destination country and to examine changes in policy decisions concerning population movements (Radu and Straubhaar, 2012).

Researchers also attempt to forecast the migration patterns and predict the population counts. The terms ‘forecast’ and ‘predict’ are used interchangeably in research regardless of a slight difference in their meaning. It is crucial to predict forced displacement, as accurate predictions can help save lives by allowing governments and NGOs to conduct a better-informed allocation of humanitarian resources. Predicting forced population counts can also be critical for policymakers to regulate migration policies and prepare for future challenges with appropriate schemes. Here, we provide an overview of existing migration forecasting methods and the recent forced displacement prediction techniques.

2.4.1 Existing forecasting methods

Bijak (2006) presents an overview of existing migration forecasting methods (see Figure 2.4). He views these methods from two perspectives: judgemental and statistical. Within the judgement-based methods, the most applied methods are unaided judgement and quantitative analogies, which adopt similar scenarios from the past with available information. For instance, Schmeidl (1997) investigated forced displacement using a time series analysis for the period of 1971-1990. Here, the adopted technique was not used to predict refugee counts, but to define and explain the causes of forced displacement. She applied statistical regression to suggest root causes of forced displacement, such as violence, foreign intervention in civil wars, and ethnic dissent. The judgemental perspective also includes the intention or expectations method that examines the potential of migration using survey-based studies. There is also the Delphi method, which refers to interactive discussions of specialists in the field of migration, and decomposition of migration, where a similar approach is applied, but with more statistical analysis.

On the side of statistical methods of migration, we find that extrapolation methods and causal models are mostly used for research purposes. The former consists of the deterministic and stochastic approaches based on probabilistic statistics, accounting for uncertainty. The latter includes econometric analysis to examine migration and forced displacement. For instance, Moore and Shellman (2007) explored forced population displacement for the period between 1964-1995 for every country that has available data in the UNHCR database. They found that refugees flee first to neighbouring countries, and after they might travel in small numbers to countries with which they have colonial relations or diasporas. There are other methods, such as data mining, which use various available sources and neural networks, and rule-based forecasting, which focuses on identifying the uncertainty of migration forecast.

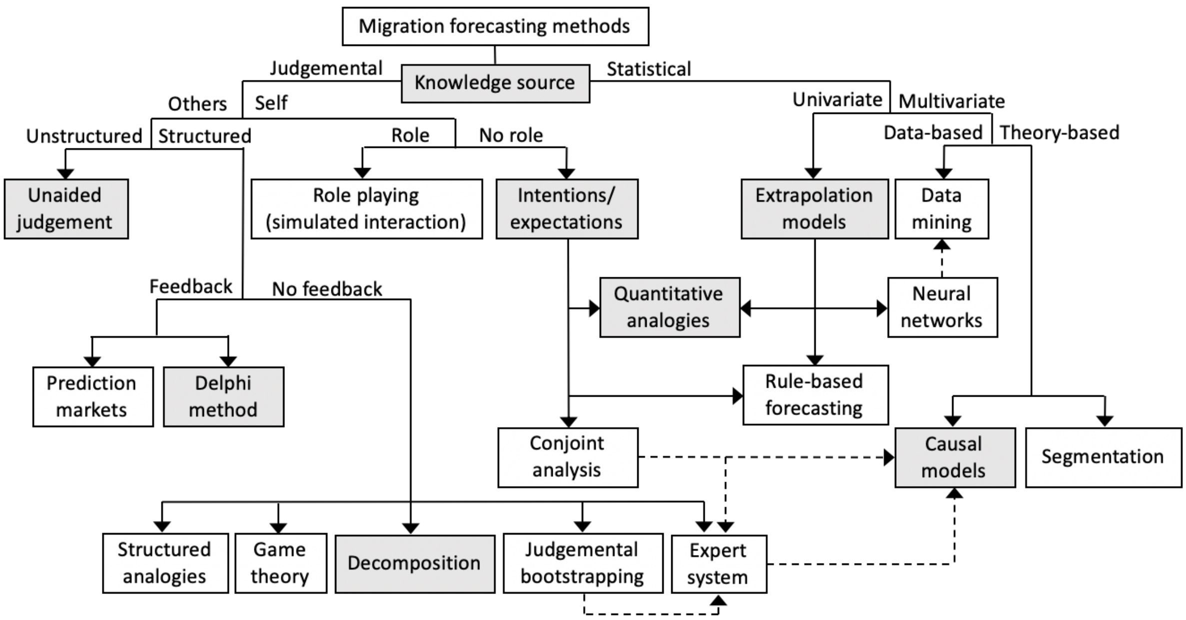

Figure 2.4: Migration forecasting methods proposed by Bijak (2006). Grey boxes represent types of forecasting methods, where straight lines show definite relationships and dotted lines show possible relationships between various methods.

Although we demonstrate a comprehensive overview of existing methods for forecasting migration in Figure 2.4, there are still several methods that have not been applied to predict forced displacement. Moreover, researchers and policymakers still face incomplete data, which mainly concentrates on voluntary migration, and uncertainty in predicting forced displacement, mostly due to problems related to data collection. Specifically, prediction techniques are less effective when current data is heterogeneously reported and documented among different countries (Lopez-Lucia, 2015). Besides, collection methods and data assessment techniques vary across organisations and economies, which in turn makes data comparison more difficult (Bijak and Wisniowski, 2011).

Hilderink et al. (2002) conduct a study predicting the forced displacement of non-Europeans in 1991 using an estimation technique. Primarily, forced displacement predictions are based on immigrants who reside illegally within Europe, regardless of refused applications for work and asylum. Nowadays, several European Union (EU) countries use this technique to predict forced displacement, while other countries rely on records of refused entry or discharge orders against migrants. Vogel et al. (2011) argue that the estimates mentioned above are outdated practices of predicting forced displacement. The reason is the irrelevance of the underlying methods used for these predictions. As an alternative, they introduce comparative estimates of maximum and minimum figures of aggregated EU estimates for the years 2002, 2005 and 2008. They note a decrease in forced displacement from 3.8 million to 1.9 million, primarily due to changes in legislation and regulations in the EU.

The United States Committee for Refugees (USCR) and UNHCR provide data for forced displacement studies. Researchers now use the UNHCR database to compare their prediction results. To illustrate, Alhanaee and Csala (2015) run a regression analysis on 1 million Syrian refugees to understand the reasons and motivation behind their choice of destination. They also use social media data from Facebook to identify their origin and current destinations and compare results to the UNHCR refugee database. Their main findings suggest that the choice of destination for Syrian refugees strongly depends on the distance travelled and factors forcing them to flee, such as protection and economic condition of destination locations.

Similarly, the European Commission launched the CLANDESTINO project, which aims to collect and predict forced displacement data for 12 countries in Europe (Jandl, 2011). The initial goal was to fill gaps in predicting movements, but it eventually became an on-line database for forced displacement. Jandl (2011) suggests that prediction, in general, is a problematic effort for policymakers and causes issues because forced displacement is undocumented.

2.4.2 Early warning models

In the late 1970s, researchers introduced an early warning model for predicting forced displacement. It aimed to detect and protect forced population but did not prevent their displacement.

Schmeidl (1997) suggested one of the unique extensions of the early warning model for forced displacement, namely the Clark model, indicating causes and factors which can be used for simulation. Schmeidl's model outlines root and proximate causes, and intervening factors of forced displacement that can be used as variables in running simulation. However, a limitation of this model is its inability to determine the importance or impact of specific parameter (Sokolowski and Banks, 2014). Indeed, it is mostly unknown what effects different parameters have on predicting forced population movements, and which variables impact prediction in general.

Shellman and Stewart (2007) investigated Haitian dispersal to the United States using an early warning model of forced displacement and predicted risk factors, such as civil violence, economic conditions and external interventions, that forced people to migrate. Similarly, Martineau (2010) used an early warning model to predict which countries have the potential to create refugees. However, existing early warning models of forced displacement focus on understanding the causes (Bunoiu and Udroiu, 2016) and are not as successful in predicting forced population movements as in predicting natural disasters (Schmeidl and Jenkins, 1998; Schmeidl, 2003; Birkmann et al., 2013). Moreover, they lack the accuracy and flexibility to accommodate the context changes that lead to large-scale forced population movements (Lopez-Lucia, 2015).

According to Edwards (2008) and Disney et al. (2015), there are relatively few appropriate models for predicting forced population displacement. The methods described above are also outdated and do not forecast forced population counts. Hence, there is a decisive gap in the research area of forced displacement. The use of computational approach and improvements in data sources may be a possible solution in forecasting the number of refugees or IDPs, which we examine in the next section.

2.5 Computational modelling techniques

Computational models have the potential to contribute to a better understanding of movement patterns, and to inform, predict and fulfil gaps within forced displacement predictions (Groen, 2016). Hence, they have been widely applied to study displacement processes (Willekens, 2016) using gravity models, system dynamics, Markov chains and agent-based models.

Park et al. (2018, p. 1) define the gravity model as "a certain type of flow between two regions is proportional to the product of 'mass' of each region and inversely proportional to a certain power of distance between the regions", where 'mass' refers to the population counts of a particular region. It allows to model population displacement using multiscale networks. To illustrate, Echevarria and Gardeazabal (2016) explore the determinants of forced displacement using a gravity model. They find that refugees bring a positive effect on the level of civil liberties in the destination country. Similarly, Abel et al. (2019) investigate the determinants of refugee dispersal for 157 countries for the years between 2006-2015 using a gravity model. Their primary focus is to identify the causal relationship between climate, armed conflict and forced displacement, which have impact specific to the time and context of a country.

System dynamics was first introduced in the 1950s to describe characteristics of information and feedback loops. Now, it uses differential equations to define the behaviour of systems using computer simulations (Amblard et al., 2010). For instance, Sato and Stansen (2007) analysed the violence, death and forced displacement in the city of Darfur, Sudan, from 2003 through 2007 using system dynamics. They identify the effects of violence that pose a risk to the population and addressed the importance of system dynamics to find genocide patterns and aid policy decisions.

Markov chain is a stochastic modelling technique that has been widely applied to different domains (e.g. biology, physics, game theory, etc.), as well as to model large-scale problems of long-distance migration. To illustrate, Huang and Unwin (2019) use the Markov chains to study the Burundian refugee crisis for the period between May 2015 - June 2016 and compare their results to an agent-based refugee model developed by Suleimenova et al. (2017). They also emphasise that their Markov chains model offers efficiency, simplicity and conciseness by omitting refugee movements across the home country and intermediate probabilities as they are incorporated into the transition probability of the Markov chains model.

2.5.1 Agent-based modelling

Agent-based modelling (ABM) is a computational approach that provides an opportunity to model complex systems. It can explicitly model social interactions and networks emerging from it. Its popularity is in part due to the decentralised nature of the approach, which allows a heterogeneous mix of many agents to act and interact autonomously, in turn leading to emergent behaviours in the system at higher levels. According to Bonabeau (2002), ABM originated from modelling individuals' decision-making and human behaviour with individual heterogeneity.

ABM is suitable for modelling active objects in relation to time, event or behaviour (Borshchev and Filippov, 2004). Macal and North (2014) provide a detailed explanation of ABM elements, which includes agents, agents' relationships and their environment (see Figure 2.5). Particularly, ABM consists of agents that represent animals, humans, organisations or any other types of entities interacting with each other and within their environment. There is no consensus on the exact definition of the term agent. However, agents are autonomous and often unique, meaning that each agent is distinct in terms of attributes, behaviour, size and location. The use of ABM allows to model how agents and their environment vary across time and space.

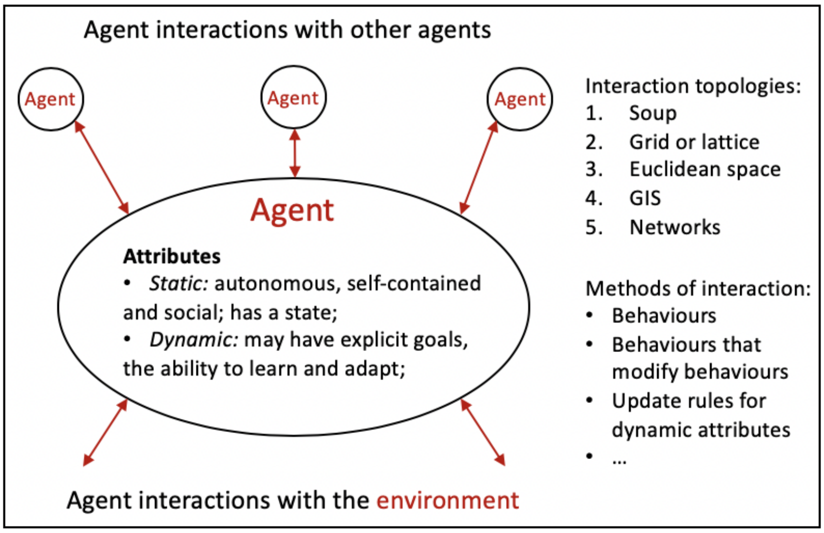

Figure 2.5: Three elements of agent-based models, namely agents, agents' relationships and their environment (Macal and North, 2014).

The initial concept of ABM was introduced in the late 1940s. However, it has become popular in the 1990s as the use of ABM required computational advancements (Arora et al., 2017). Today, ABM is widely applied to various research disciplines, such as biology, business, economics, social sciences, and technology, and practical areas including infrastructure, civilisation, terrorism, military and crowd modelling. There are also newly emerging application domains, such as cyber-security and the social factors of climate change (Allan, 2010).

In this thesis, we concentrate on the domain of crowds involving human movement patterns and evacuation modelling. Schelling (1971) was the first to represent people as agents and their social behaviour as agent interactions. Only after two decades, Epstein and Axtell (1996) broadened the idea of modelling human behaviour and their movement patterns in society and geography. More recently, human movement modelling has expanded using various ABM software tools. To illustrate, we present seven ABM software tools in six application areas mutually exclusive in considering patterns of human behaviour (see Table 2.5). These ABM tools are diverse in terms of source code language, model development, and their level of scalability (Abar et al., 2017). However, they do not explicitly model migration or forced population displacements.

ABM application	**Software tool**	**Availability**	**Source code**	**Model development**	**Scalability level**
Mapping passenger flow	HLA_Agent* HLA_Repast	Open source Open source	C++ Java	Complex, hard Complex, hard	Large scale Large scale
Microsimulations for demographic development	Modgen**	Closed source	Microsoft Visual Studio	Moderate	Medium-scale
Pedestrian crowd mapping	PedSim	Open source	C++	Simple, easy	Small-scale
Tourist flow	Simio	Closed source	C#	Moderate	Medium-scale, large-scale
Military scenarios, human behaviour, actions modelling	SimJr	Open source	Java	Simple, easy	Small-scale
Urban development modelling	UrbanSim	Open source	Opus with Python and Numpy	Moderate	Large-scale

* High Level Architecture Agent
** Model generator

Table 2.5: Comparison of ABM software tools within the scope of human movement (Abar et al., 2017).

There are also ABM libraries or programmes, namely Swarm, MASON, Repast, NetLogo, AgentSheets and AnyLogic, widely used to build ABM simulations. In Table 2.6, we compare these libraries on several characteristics. It is apparent that NetLogo and AgentSheets are quick to execute and require basic programming experience to use. These libraries may not be the best fit for large and complex system problems, while MASON and Repast are fast at execution and provide a platform to model. However, the latter two libraries demand strong programming skills to learn, install and execute. Yet, none of these ABM libraries are ideal for modelling and predicting forced displacements counts.

ABM library	**Date of inception**	**License**	**Modelling language**	**Programming experience**	**Speed of execution**
Swarm	1996	Open source	Objective-C, Java	Strong	Moderate
MASON*	2003	Open source	Java	Strong	Fast
Repast**	2000	Open source	Java, Python, Microsoft.Net	Strong	Fast
NetLogo	1999	Shareware	Proprietary scripting	Basic	Moderate
AgentSheets	1991	Proprietary	Proprietary scripting	None - Basic	Moderate
AnyLogic	Unknown	Proprietary	Proprietary scripting	Moderate	Fast

* Multi-Agent Simulator of Neighbourhoods (or Networks)

** Recursive Porous Agent Simulation Toolkit

Table 2.6: Comparison of existing ABM libraries (Castle and Crooks, 2006; Gilbert, 2008).

Although ABM models problems ranging from small-scale behavioural dynamics to large scale migration simulations (Macal and North, 2010), it is becoming particularly prominent for population movement studies (Castle and Crooks, 2006; Crooks et al., 2008). It is already used in a wide range of refugee-related settings, such as disaster-driven migration, which incorporate changes in climate and demographics (Entwisle et al., 2016). For example, Hassani-Mahmooei and Parris (2012) analysed the influence of climate change on migration in Bangladesh while Kniveton et al. (2011, 2012) developed an ABM to simulate climate migration in Burkina Faso between 1970–2000 and to predict future migration flows to 2060. Additionally, Anderson et al. (2006, 2007) suggested an ABM for refugee communities to inform policy decisions for governments and other organisations. The German armed forces developed an ABM to understand the interactions and behaviour of refugees with military groups in refugee camp environments (Johnson et al., 2009).

Raymer and Smith (2010) stress that there are four aspects to consider when modelling forced displacement, namely the type of migrants, methods structuring available data, modelling approach and measures of uncertainty associated with data. It is also crucial to consider the course of movement of refugees and IDPs, including when they decide to leave, where they choose to flee and whether to stay or flee further from the first destination choice (Hébert et al., 2018). Thus, ABMs could be applied interactively to assist governments, organisations and NGOs in estimating when and where the forced population are likely to arrive (Estrada et al., 2017), and which camps are most likely to become full in the short term.

Similarly, Latek et al. (2013) build a multi-agent model that predicts the Syrian conflict characteristics and investigated potential conditions and outcomes of the conflict. Hattle et al. (2016) examined the Syrian refugee flows to European countries using ABM and discussed possible policy recommendations on distributing humanitarian resources amongst potential refugee-hosting countries. Several groups also applied ABM to capture social aspects, such as networks, group formation and travel distance in the forced displacement crisis and stress the importance of computational modelling for migration predictions (Collins and Frydenlund, 2016; Lin et al., 2016).

Sokolowski and Banks (2014) propose an ABM Environment Matrix methodology for constructing forced displacement simulations. Precisely, they establish their simulation using Schmeidl's early warning model of forced displacement, match the factors with UNHCR and develop an assessment template to record model outcomes. They use their ABM Environment Matrix for a specific situation, namely forced displacement in the Syrian city of Aleppo, and attain preliminary results using one hundred replications of Monte Carlo simulations (Sokolowski et al., 2014). However, they do not verify their model with any other conflicts or validate simulation output against reliable data. Hence, further investigation and analysis are required to justify their methodology, as well as to determine the principal causes and intervening factors in these ABM simulations.

There are other significant challenges within the ABM community. For instance, there is an ongoing debate on whether prediction should be a major purpose for ABMs (Elsenbroich, 2012), or whether explaining and illuminating problems should be a priority (Epstein, 2008). Moreover, Klabunde and Willekens (2016) identify major challenges in the selection of empirical evidence for model validation within migration studies, where model validation refers to "the process of determining whether a simulation model is an accurate representation of the system, for the particular objectives of the study" (Law, 2005, p. 24). Furthermore, there is an issue of obtaining necessary data for all output, which may be insufficient and unavailable in great detail.

Most of the human displacement studies perform validation and sensitivity analysis at the basic level. Sensitivity analysis is a vital element of prediction and validation that determines parameters to which simulation outcome may be sensitive (Cirillo and Gallegati, 2012). It is plausible that small change in model parameters may induce immense effects on the output derived from a simulation, but it is not viable to analyse every single parameter for its sensitivity on the output. Nevertheless, an automated simulation can create the whole environment for researchers and organisations to investigate parameters for sensitivity, to validate the obtained results and execute an ensemble of runs by simplifying and accelerating key phases of simulation.

Constructing and executing any simulation also oppose the issue of unstructured parameters, excessive details and assumptions in design, implementation and documentation. In turn, modellers are not able to produce reliable and reproducible simulations. Therefore, simulations must be complete with description and specification to replicate and reiterate them independently by researchers, developers and others. Similarly, it is essential to address the importance of model verification as it ensures the correctness of a model and error-free simulations for execution. There are various model verification techniques, such as elegance of code scripts, assertions and comments describing the purpose of code blocks, and unit testing, that minimise and identify bugs in ABM simulations (Gilbert, 2008).

2.6 Conclusion

In this chapter, we examined migration theories and forced displacement models to understand human migration today comprehensively. Since researchers mostly investigate why human migration occurs and what effects it has on economies, it was evident that these theories and models are generally not extensive enough for practical applications. The existing judgemental and statistical methods are also outdated and do not predict forced population counts.

Hence, there is a decisive gap in the research area of forced displacement. The use of computational approaches and improvements in data sources may be a possible solution in forecasting the number of refugees or IDPs. Particularly, widely adopted ABMs have the potential to contribute to a better understanding of population dispersal patterns, and to inform, predict and fulfil gaps within forced displacement predictions. They could also be applied to assist governments, organisations and NGOs in estimating forced population arrivals escaping violence and armed conflicts. In the next chapter, we explore the ABM development techniques and processes to aid the development of ABM simulations, both for predicting forced population displacement patterns and for assessing the implications of policy decisions.

Chapter 3. Research Methodology

3.1 Introduction

In this chapter, we present our research methodology for developing a computational simulation technique. We examine existing literature on simulation processes and propose a generic simulation development. Notably, we develop a generalised SDA to build and validate agent-based simulations (ABS) for situation-specific scenarios.

3.2 Requirements for our methodology

The notion of a development process has been referred to in the literature by different headings, such as model development process, model cycle or life cycle of simulation, methodology process, methodological framework or steps for a successful simulation. Despite these variations, they comprise, reflect and share the concept of cyclic phases using theoretical and empirical analyses. In our view, the term simulation development approach (SDA), which we use throughout this thesis, defines the idea of a development process for computational simulation.

The development processes of ABS vary, but they still consist of similar fundamental building phases. Researchers use a systematic set of phases to build ABS. These phases include the formulation of the real world problem, transfer it into a model and then convert the model into a computerised simulation, execute experimental runs, analyse the outcome and present documentation for re-use. In turn, ABS provides results to a formulated problem, which represents a real system, using accurate and appropriate simulation development process (Balci, 1994; Law, 2008; Sargent, 2011). To illustrate, Heath et al. (2009) provide a clear overview of basic development blocks for a computational simulation (see Figure 3.1).

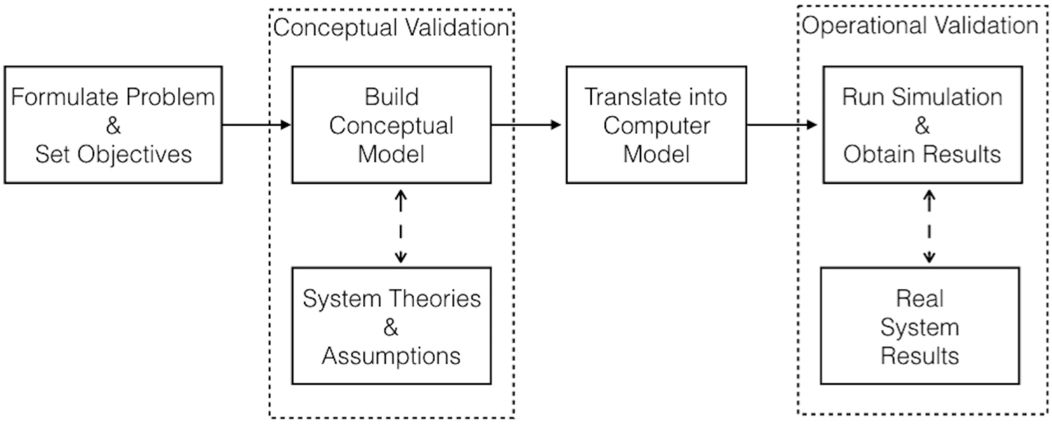

Figure 3.1: A basic development process of simulation (Heath et al., 2009).

Moreover, Davidsson et al. (2007) survey the application of ABM and present an evaluation framework including four ABM development phases, namely problem description, modelling approach, implementation approach and results (see Table 3.1). They provide an extensive analysis that describes aspects and categories related to each phase of ABS development, where the use of ABM spread comprehensively across various domains.

	Aspects	**Categories**
Problem description	Domain	animal societies, physiological systems, social systems, organisations, economic systems, ecological systems, physical systems, robotic systems, transport/traffic systems
	End-user	scientists, policy makers, managers, other professionals
	Purpose	prediction, verification, analysis, training
Modelling approach	Simulated entity	living, physical artefact, software process, organisation
	Agent types	1 - 1 000
	Communication	yes/no
	Spatial explicitness	yes/no
	Mobility	yes/no
	Adaptivity	yes/no
	Dynamic	yes/no
Implementation Approach	Platform used	NetLogo, Repast, Swarm, JADE, C++, etc.
	Simulation size	1 - 10 000 000
	Scale	limited/partial or full-scale
	Input data	artificial data or real data
	Distributed	yes/no
	Mobile agents	yes/no
Results	Maturity	conceptual proposal, laboratory experiment or deployed
	Evaluation	qualitative, quantitative or none
	Validation	qualitative, quantitative or none

Table 3.1: Summary of an evaluation framework for four development phases of ABM (Davidsson et al., 2007).

Researchers derive these simulation development phases from requirements that justify simulation techniques. These requirements address the validated, verified and reproducible solution of computational problems, which is necessary for the rapid construction of models and execution of simulations. We distinguish between 'model' and 'simulation'. According to Robinson (2008, p. 283), a model is "a non-software specific description of the computer simulation ... describing the objectives, inputs, outputs, content, assumptions and simplifications of the model". Hence, the difference is a formulated problem (model) prior to translation and deployment into a computational or computerised version (simulation).

In our methodology, there are several requirements for ABS development. First, it is important to establish a *testable* simulation to validate obtained results against the real data for evaluation purposes. Second, there is a necessity for a *unified* approach for situations given a specific problem to facilitate accurate and rapid development of ABS. The third requirement is an *end-to-end* simulation development that can provide complete and direct application of a model by stakeholders, such as researchers, governments and NGOs, to save time and gain efficiency. Finally, we require a *simple* simulation process, which provides a basis for gradual development and clarity to each change, and flatten the learning curve in model development. In sum, these methodology requirements, namely testability, unification, completeness and simplicity, are key drivers for the definition of the simulation development phases.

3.3 Generic model and simulation development for validation

Problem formulation is a meaningful basis of modelling defined as "the process by which the initially communicated problem is translated into a formulated problem sufficiently well defined to enable specific research action" (Balci, 1994, p. 126). This phase has the purpose of providing a clear and complete definition, description and purpose of the problem to undertake simulation (Birta and Arbez, 2013).

In this thesis, our main aim is to develop a generalised and accurate SDA, predicting forced population displacement in conflict regions. With our aim and methodology requirements in mind, we can retrace what kind of activities may be required for the simulation development. First, we select a situation with a testable evaluation of simulation results. To achieve this, we validate our simulation results against real-world data. To obtain those simulation results, we, of course, need to execute an actual simulation. In addition, we need to perform uncertainty quantification analysis and extract real-world data (see Figure 3.2).

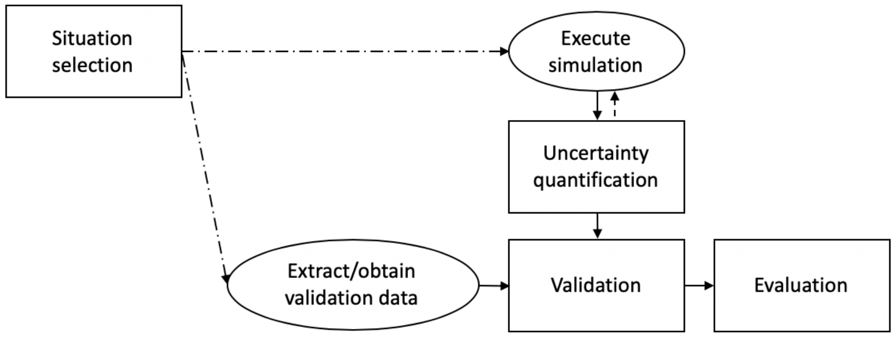

Figure 3.2: SDA phases for a testable evaluation of a selected situation.

Second, to execute any simulation, we require a refined simulation that is a modified or improved form of a simulation. Any 'simulation' is an implemented equivalent of a model, so we also require a refined model. Taking this into account, we then obtain the SDA shown in Figure 3.3.

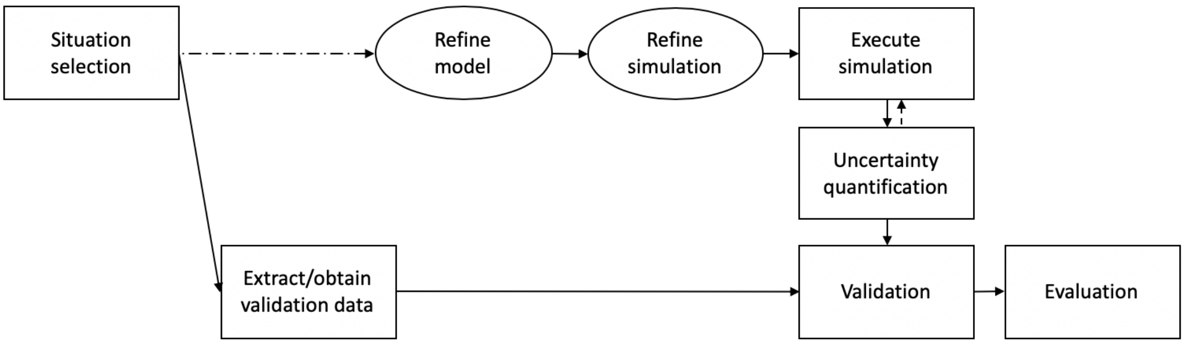

Figure 3.3: SDA phases presented in Figure 3.2 with addition of refined model and simulation.

Third, any 'refined' simulation is ideally a refined version of situation-agnostic simulation rather than an ad-hoc one. Similarly, any model ideally is derived from a situation-agnostic model and modified with situation-specific circumstantial evidence. To use a model, we normally need to extract and obtain input data, which requires to be clearly identified and specified by the selected situation (see Figure 3.4).

Fourth, to create a situation-agnostic model, we need to specify our input data identify our assumptions and determine whether they are unidentified assumptions (or free parameters), heuristics, which refers to common sense, or evidence-driven assumptions that are identified from data or supporting information. Moreover, we need to redefine and rescope our validation metrics to match the type of situation-agnostic model we defined.

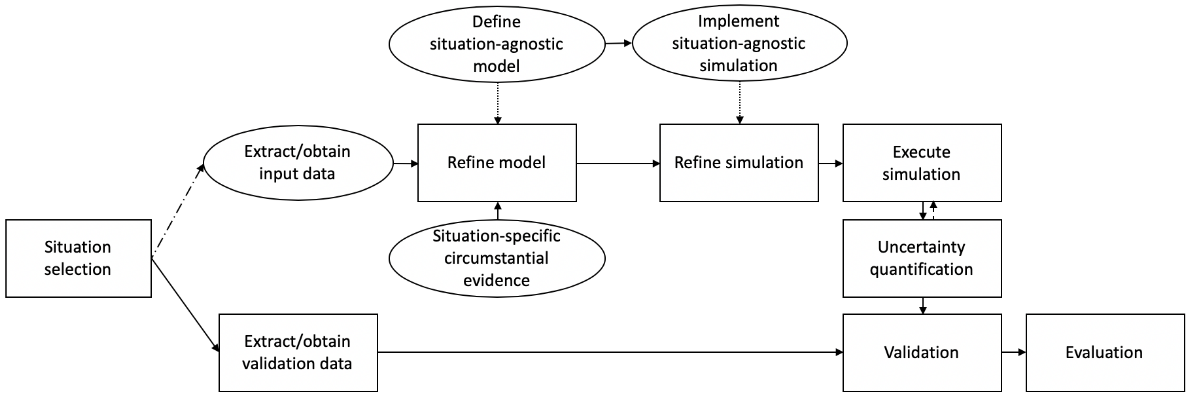

Figure 3.4: SDA presented in Figure 3.3 with additional four phases required for model and simulation refinement.

In Figure 3.5, we present our generalised SDA captured by two main step-by-step processes, namely the generic model, which has one-time construction sequence, and *simulation development for validation* that applies to individual situation-specific scenarios. Importantly, the situation selection phase is constrained by the scope problem definition of a generic model. Together, these phases clarify the interconnected and cyclical tasks required to achieve an efficient simulation development for validation process.

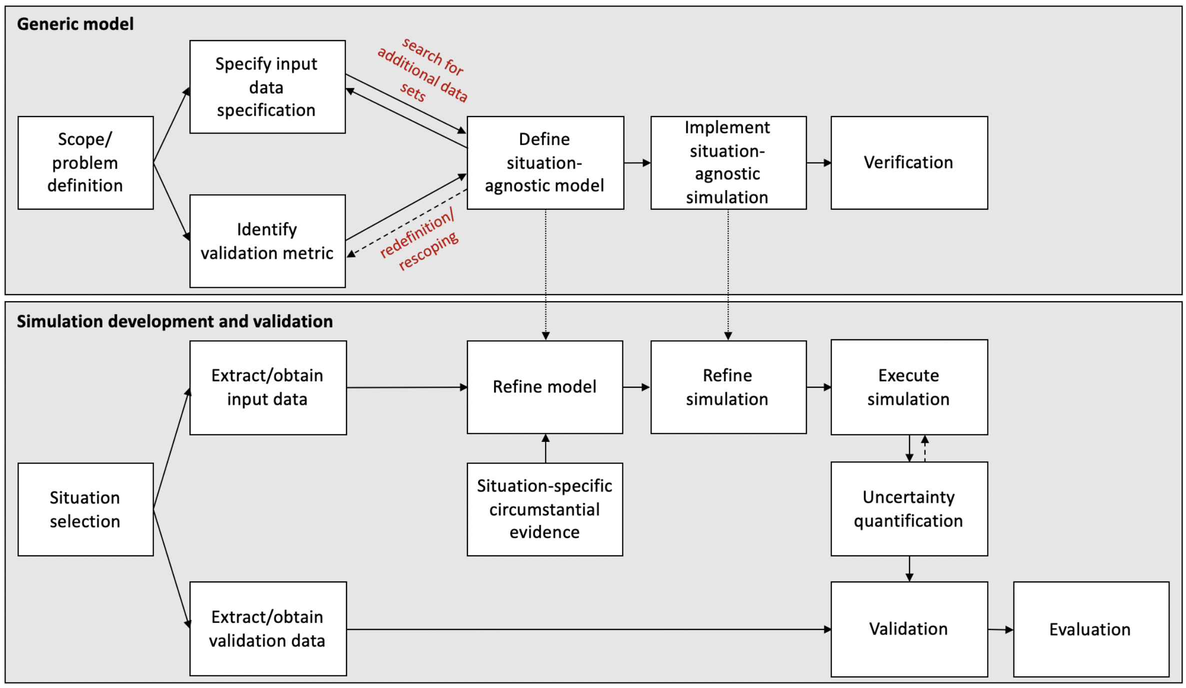

Figure 3.5: Overview of a generalised simulation development approach including generic model and simulation development for validation.

Validation is a crucial part of model evaluation, but model context assessment and goodness of fit to a research problem have importance too. Bearing this in mind, we evaluate our SDA by exploring its applicability, usability, consistency and generality. Specifically, we can apply our SDA to a situation-specific context and self-evaluate the process through user experiment analysis or request other researchers to use SDA and get feedback by conducting a usability study or interviews. Moreover, we can search for a researcher(s) who applied SDA to a selected problem in other disciplines, explore literature for similar scenarios or track citations of SDA publications, which are all future work.

3.4 Simulation development approach for forecasting

The simulation development for forecasting is different from ones for validation presented in the previous section of this chapter. It is because validation simulations are the prediction of past and compared to historical data while forecasting simulations attempt to predict future trends and patterns, and are evaluated using post-forecast validation from real-world observations. Moreover, several data sets may be incomplete, unreliable or even non-existent when using simulations for forecasting context. Therefore, forecasting tools can replace empirical evidence. For instance, several databases are providing historical data on armed conflict locations, but we require a model to generate these data for forecasting purposes.

Forecasts also rely on different assumptions compared to validation, as it is challenging to forecast future events and patterns. Hence, there is high uncertainty within forecasting assumptions, including social, political and economic changes and policies, forced displacement patterns, and conflict propagation of crisis.

We present our SDA for forecasting in Figure 3.6, which has an additional step of defining forecasting scope and metrics to tune output metrics and meet forecast requirements, which is linked and concurrently carried out to refine model and simulation.

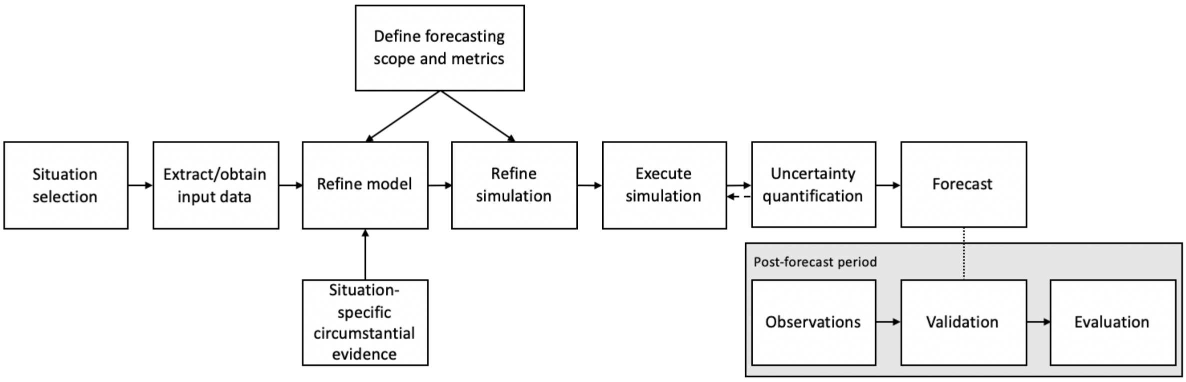

Figure 3.6: Overview of SDA for forecasting simulation.

3.5 Conclusion

In this chapter, we discussed existing simulation development processes and identified issues within any simulation development. To tackle these issues, we proposed a generalised SDA for situation-specific scenarios, such as forced population displacement. Our approach consists of two parts, namely a generic model, simulation development and validation. The former is one-time construction initiated by problem definition of a real system and translated into situation-agnostic model and simulation. The latter focuses primarily on a specific situation constrained by the main problem and selected by a modeller, whether it is a researcher, government or NGO. Moreover, we develop a generalised SDA specific to the situation, which has an additional phase of defining forecasting scope and metrics. We emphasise that our generalised SDA for validation and forecasting applies to other simulation models.

In the next chapters, we adopt our generalised SDA methodology to predict the distribution of incoming forced population across destination camps (Chapter 4). We then apply our approach to model forced displacement crisis in three African countries as a validation study (Chapter 5). Finally, we automated our purposed SDA to improve the simulation process and provide better integration of application (Chapter 6) and explore effects of policy decisions on forced displacement conflict of South Sudan (Chapter 7).

Chapter 4. Forced Displacement Simulation Development Approach

4.1 Introduction

In this chapter, we use our proposed SDA methodology, as introduced in Chapter 3, to predict the distribution of incoming forced population across destination camps forced to flee because of war, persecution and/or political instability. Our SDA for forced displacement has six main phases: situation selection, data collection, model construction, model refinement, simulation execution and analysis (see Figure 4.1).

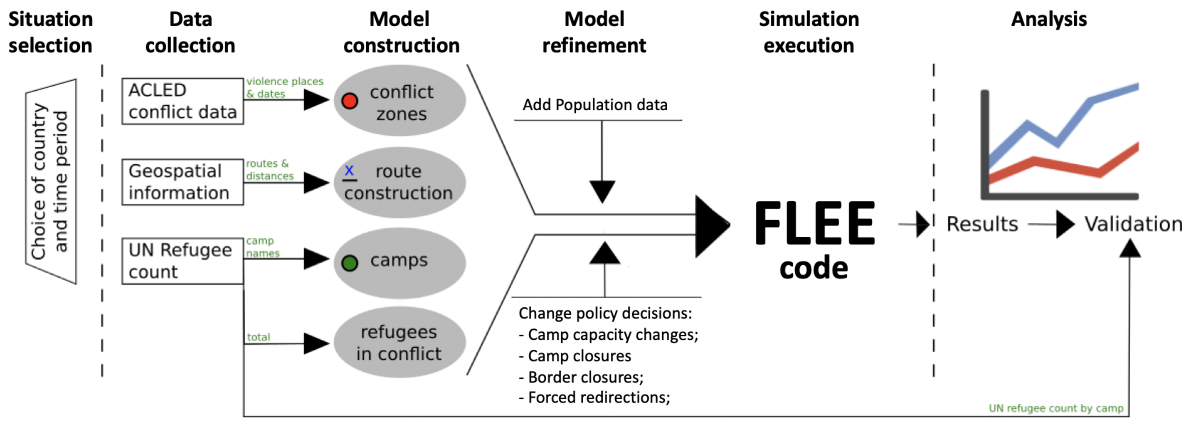

Figure 4.1: A simulation development approach to predict the distribution of incoming forced population across destination camps.

In our approach, we first select a country and time period of a specific conflict, which resulted in large scale forced displacement. Second, we obtain relevant data to the conflict from three data sources: the United Nations High Commissioner for Refugees (UNHCR, data2.unhcr.org), the Armed Conflict Location and Event Data Project (ACLED, acled-data.com), and the Bing Maps platform (bing.com/maps). Third, we construct our initial model using these data sets and create, among other things, a network-based ABM model.

Once we have built the initial model, we refine it as part of the fourth phase. Here, we manually extract population data to help determine where forcibly displaced people flee from, as well as information on border closures and forced redirection. The fifth phase involves the main simulation, which we run to predict, given a total number of forced population in the conflict, the distribution of displaced people across the individual camps. We run our simulations using the FLEE simulation code. FLEE is optimised for simplicity and flexibility and provides a range of scripts to handle and convert forced population data from the UNHCR database. Once the simulations have completed, we analyse and validate the results against the full UNHCR forced displacement numbers as part of the sixth phase. In the next sections of this chapter, we extensively elaborate on each phase to provide more insights on constructing and executing forced displacement simulations (see Appendix A and B for detailed tutorial).

4.2 Situation selection

The first phase of forced displacement simulation is the selection of a conflict country with fleeing migrants and a conflict duration to indicate the simulation time period. We select a conflict country from the UNHCR operational portal (`data2.unhcr.org/en/situations`), which covers 96 countries and provides a thorough overview of 23 conflict situations. The portal also has numerous reports and background information to gather initial narrative and set a clear modelling scenario. We choose one of the conflicts and specify the simulation time period to initiate a forced displacement prediction.

4.3 Data collection

In the second phase, we collect data for the chosen conflict using three data sources. The first data source is the UNHCR database, which presents data for the number of forcibly displaced people in the conflict, the camp locations in neighbouring countries and their population capacities. We also derive the total number of registered forced population from the public UNHCR, which is the number of new forced population arrivals in our simulation. We refine the data by interpolating linearly between data points, and calculate the total forced population count by aggregating the (interpolated) registrations for each of the camps.

Moreover, we manually obtain the forced displacement registration data for each camp from the website in comma-separated values (CSV) format, including the name and locations of the camps, as well as their estimated capacities. These data also include level 1 registrations and, after certain dates, level 2 registration. As level 1 registrations are known to result in overestimations of forced population count, we scale down these values so that the last data point using level 1 registrations matches the first data point using level 2 registrations.

The second source is the ACLED data that provides the locations and dates of battles that have taken place in the conflict. The UNHCR data at the destination camps populate conflict zones or the source locations. Therefore, we determine the conflict areas or zones from locations where battles occur. We note the start date of any event as 'battle' during the simulation period. All conflict locations are assigned a population based on the latest census data. We omit settlements with less than 10,000 inhabitants to emphasise on large conflicts.

The third data source is the Bing Maps platform to identify locations of major settlements and to route information between the various camps, conflict zones and other settlements. We select locations by combining our UNHCR camp locations and ACLED conflict locations with other major settlements that reside en-route between these locations by using the Bing Maps platform. Locations are interconnected with links in cases where we notice the presence of roads in Bing Maps. The length of the link (in km) is then estimated using the Bing route planner for cars. In cases where obvious shorter routes are visible, we drag the Bing marker to force the software to calculate this shorter route. To retain the simplicity of our model, we directly connect forced displacement camps to the nearest location in the country of conflict. Identified conflict zones are locations that are connected with routes or paths demonstrating forced population movements towards safer places or camps in neighbouring countries.

4.4 Model construction

In the third phase, we build our initial model for forced displacement simulation. Here, we construct a geographic network model using `carto.com` and OpenStreetMap data, which is a route representation linking conflict zones, camps and intermediate towns. We also create a script and specify the collected information from UNHCR, ACLED and Bing Maps for the selected conflict scenario. In Figure 4.2, we demonstrate an example code that includes conflict period, location names and distances between these locations.

```python
if __name__ == "__main__":
  print("Simulating Country X.")

  end_time = 10
  e = flee.Ecosystem()

  l1 = e.addLocation("A", movechance=0.3)

  l2 = e.addLocation("B", movechance=0.0)
  l3 = e.addLocation("C", movechance=0.0)
  l4 = e.addLocation("D", movechance=0.0)

  e.linkUp("A","B","834.0")
  e.linkUp("A","C","1368.0")
  e.linkUp("A","D","536.0")

  d = handle_refugee_data.DataTable("source-data-unhcr.txt", csvformat="countryx-pdf")
```

Figure 4.2: Example code for constructing an initial model.

We present in Figures 4.3 and 4.4 flowchart descriptions of the algorithm assumptions for forced displacement simulations. Each step of the simulation represents one day. During each step, we insert forced population counts into the simulation based on the daily increase in the total registration count from the UNHCR data. These displaced people are inserted in their location of origin, which is one of the conflict locations (as obtained from the ACLED database). The exact location is picked among all conflict zones, where the likelihood of each conflict zone being selected is proportional to its population. The population of a location is decremented by one each time a displaced person is created. We insert forced population in conflict zones on the day of camp registration and forced population travel, which is non-instantaneous. To correct for this, we multiply the forced population in each of the camps by $N_{data,all}/N_{sim,all}$, where $N_{data,all}$ is the total forced population count for the conflict on a given day according to the UNHCR data. In our setting, this is a known quantity, as we are predicting the distribution of forced population across camps, given this total forced population count. $N_{sim,all}$ is the total number of forced population in camps according to the simulation on that same day. Decreases in UNHCR forced displacement registrations increment a “forced population debt” variable, which first needs to be compensated by subsequent registration increases before additional agents are again inserted in the simulation (i.e., we do not delete agents).

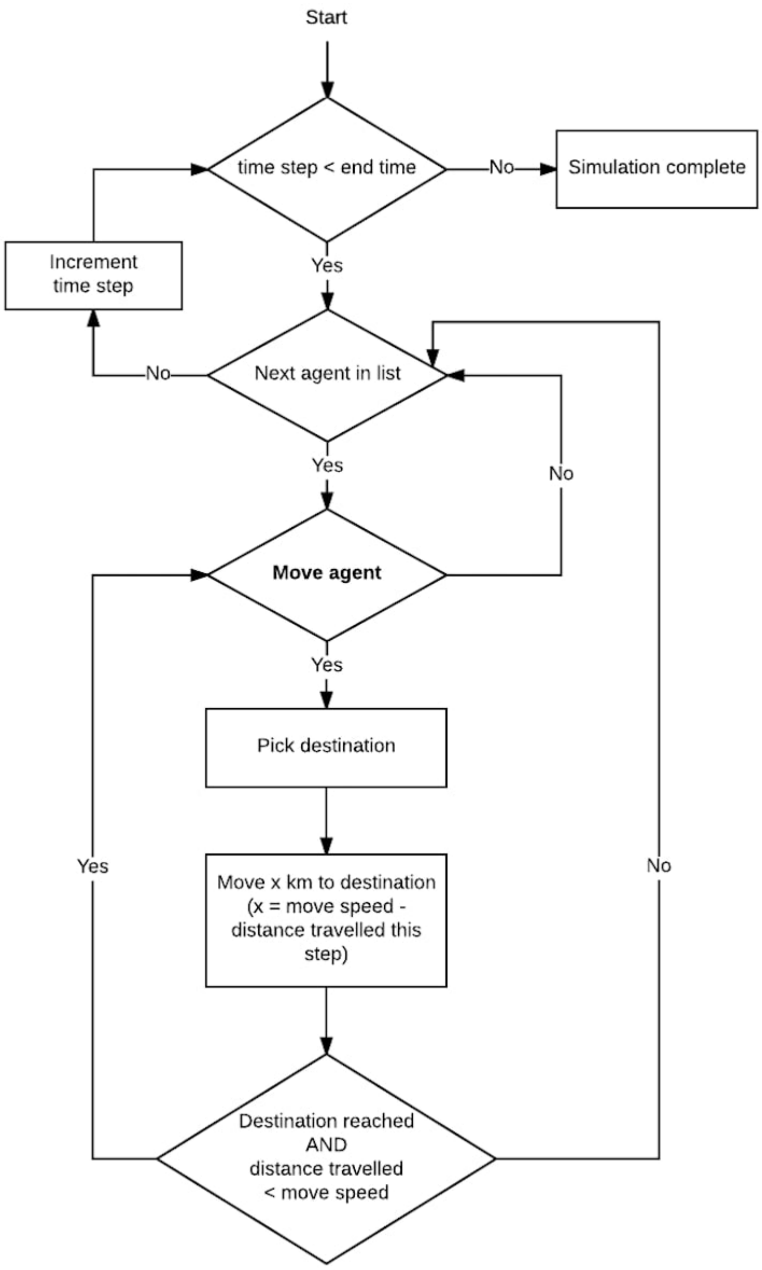

Figure 4.3: A basic flowchart of algorithm assumptions used by the FLEE code to predict forced population destinations. Move agent factor has a detailed sub-path demonstrated in Figure 4.4.

During each step, a displaced person traverses either zero or one link. The probability of traversing a link is determined by the move chance, which is location dependent. When an agent traverses a link, it needs to choose one of the available paths. Path selection is made using a weighted probability function, the weight of each link is equal to the attractiveness value of the destination divided by the length of the link in kilometres. The attractiveness value of the destinations equals 0.25 for conflict zones, 1.0 for other locations in the country of conflict, and 2.0 for locations abroad. We assume that all links require all day travel to traverse, due to the unavailability of data for travel times and departure dates. In addition, forcibly displaced people take major roads, which are the shortest journey paths identified using bing.com/maps.

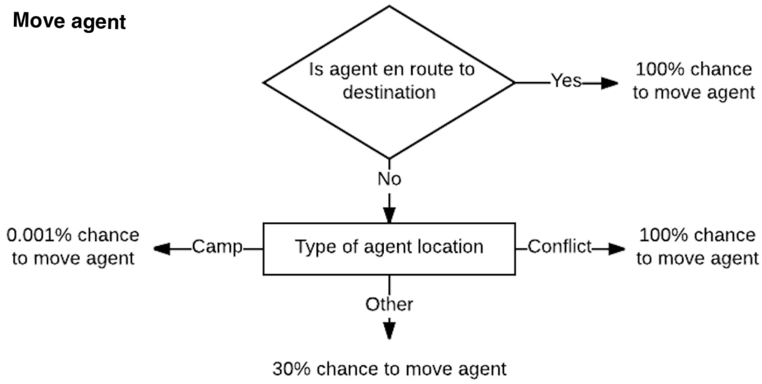

Figure 4.4: A detailed description of move agent where agents (forced population) have three variations of location determined by the move chances.

4.5 Model refinement

The fourth phase focuses on refining the initial model with population data to improve simulation and incorporate the actual population counts. In some cases, we add 'forwarding' locations, where the forced population are automatically rerouted to a camp, following descriptions in the UNHCR reports. We also remove links when there is a camp or border closure and add links back when UNHCR reports camp and border openings. We illustrate code fragments for these instances in Figures 4.5 and 4.6. Alternatively, some reports might declare that displaced people arrived at camps on foot, due to the lack of roads. To accommodate this fact, we incorporate specific 'off-road links' from conflict zones to camps. Moreover, off-road routes are likely to result in slower travel speeds. Thus, we multiply the coordinate point-by-point distances by 2 for all walking routes.

```
def close_location(self, location_name, twoway=True):
  """
  Close in- and outgoing links for a location.
  """
  if twoway:
    return self._change_location_1way(location_name, mode="close", direction="both")
  else:
    return self._change_location_1way(location_name, mode="close", direction="in")

def reopen_location(self, location_name, twoway=True):
  """
  Reopen in- and outgoing links for a location.
  """
  if twoway:
    self._change_location_1way(location_name, mode="reopen", direction="both")
  else:
    self._change_location_1way(location_name, mode="reopen", direction="in")
```

Figure 4.5: Code fragment for a camp or location closure or reopening.

```
def close_border(self, source_country, dest_country, twoway=True):
  """
  Close all links between two countries.
  If twoway is set to false, the only links from source to destination will be closed.
  """
  self._change_border_1way(source_country, dest_country, mode="close")
  if twoway:
    self._change_border_1way(dest_country, source_country, mode="close")

def reopen_border(self, source_country, dest_country, twoway=True):
  """
  Re-open all links between two countries.
  If twoway is set to false, the only links from source to destination will be closed.
  """
  self._change_border_1way(source_country, dest_country, mode="reopen")
  if twoway:
    self._change_border_1way(dest_country, source_country, mode="reopen")
```

Figure 4.6: Code fragment for a border closure or reopening.

4.6 Simulation execution

We use the FLEE code (https://github.com/djgroen/flee-release) to execute our simulations, which is an agent-based modelling code written in Python with a limited feature set and optimised for simplicity and flexibility. It is able to support simulations with 100,000s of agents on a single desktop, and provides users with the ability to define and use their models through a relatively straightforward application programming interface (API). We provide a range of functional tests to allow users to verify the consistency of the code results. FLEE also features a range of scripts to handle and convert forced population displacement data from the UNHCR database, as well as an automated plotting tool for output generated by the simulation (see Figure 4.7). To use the code, one requires a Python 3 interpreter, as well as the numpy, scipy, matplotlib and pandas Python modules. For detailed instruction on how to install and use the FLEE code, see Appendix B.

```python
def plotme(out_dir, data, name, offset=0, legend_loc=4, naieve_model=True):
  """
  Advanced plotting function for validation of refugee registration numbers in camps.
  """
  plt.clf()

  # data.loc[:,["%s sim" % name,"%s data" % name]]).as_matrix()
  y1 = data["%s sim" % name].as_matrix()

  y2 = data["%s data" % name].as_matrix()
  days = np.arange(len(y1))

  #print(name, offset, len(y1), len(y2))
  plt.xlabel("Days elapsed")

  matplotlib.rcParams.update({'font.size': 20})

  #Plotting lines representing simulation results.
  if offset == 0:
      labelsim, = plt.plot(days,y1, linewidth=8, label="%s simulation" % (name.title()))
  if offset > 0:
      labelsim, = plt.plot(days[:-offset],y1[offset:], linewidth=8, label="%s simulation" % (name.title()))

  # Plotting line representing UNHCR data.
  labeldata, = plt.plot(days,y2, 'o-', linewidth=8, label="%s UNHCR data" % (name.title()))

  fig = matplotlib.pyplot.gcf()
  fig.set_size_inches(12, 8)
  #adjust margins
  set_margins()

  if offset == 0:
      fig.savefig("%s/%s-%s.png" % (out_dir, name, legend_loc))
  else:
      fig.savefig("%s/%s-offset-%s.png" % (out_dir, name, offset))
```

Figure 4.7: Code fragment for an automated plotting of simulation results again the UNHCR data.

4.7 Analysis

In the final phase of our SDA for forced displacement, we analyse our simulation results by calculating an averaged absolute error using the following equation:

$$E(t) = \frac{\sum_{x \in S}(|n_{sim,x,t} - n_{data,x,t}|)}{N_{data,all}} \tag{4.1}$$

Thus, the number of forced population found in each camp x of the set of all camps S at time t is given by $n_{sim,x,t}$ based on the simulation predictions, and by $n_{data,x,t}$ based on the UNHCR data. The total number of displaced people reported in the UNHCR data is given by $N_{data,all}$.

We also present comparisons to naive models using the Mean Absolute Scaled Error (MASE). It was first proposed by Hyndman and Koehler (2006), and is particularly well suited to quantify simulation accuracy due to its scale-invariant nature and it symmetrically penalises overestimations and underestimations. We calculate the MASE score using the aforementioned averaged relative difference at each time step, as follows:

$$MASE = \frac{1}{T} \frac{\sum_{t=0}^{T} E(t)}{\frac{1}{T-w} \sum_{t=w}^{T} E_{naive}(t)} \tag{4.2}$$

Here, T is the full duration over which the naive prediction can be made, w is the warmup period required for the naive model to make its predictions (depending on the model type). The averaged relative difference using the naive model compared to the validation data is given by $E_{naive}(t)$. In addition, the MASE is straightforward to interpret. In our case, its value is less than one if our prediction approach has a smaller error, while its value is more than one if the selected naive technique has a lower error.

4.8 Conclusion

In this chapter, we introduced our SDA for forced population displacement consisting of six main phases of situation selection, data collection, model construction, model refinement, simulation execution and analysis of model outcome. Each phase is described in detail to provide a comprehensive understanding and an opportunity to construct and execute an ABS predicting forced population distributions. The emergence of publicly and thoroughly curated data repositories of the last decade, such as UNHCR, ACLED and Bing Maps platform, enable us to reconstruct conflict situations with unprecedented accuracy, and provides us with the empirical data needed to build our simulations. In the next chapter, we apply our SDA to three African crises following a clear process from construction to execution and analysis of simulation results.

Chapter 5. Validation Study

Based on:

Suleimenova, D., Bell, D. and Groen, D. (2017), "A generalized simulation development approach for predicting refugee destinations". *Scientific Reports*, 7 (13377).

5.1 Introduction

In this chapter, we apply our generalised SDA to model three forced displacement crises in African countries, including two crises that have never been simulated before. These conflicts are the 2015 civil war in Burundi, the conflict in the Central African Republic (CAR) between 2013 and 2016, and the Northern Mali conflict in 2012. We construct, run and analyse forced population movement simulations using an ABM estimating the distribution of incoming forced population across destination camps, given the expected total number of forced population in the conflict. We are also able to reproduce key trends in forced population arrival rates found in the UNHCR data across three African countries.

It is important to be able to predict where forcibly displaced people go to save their lives, as it helps governments and NGOs to allocate humanitarian resources correctly; to help complete incomplete data collections on forced population movements; and to investigate the consequences of a nation closing its border for forced displacement. We examine these motives by applying a generalised SDA to three African countries of Burundi, CAR and Mali.

5.2 Overview of conflict situations

Burundi has a lengthy history of civil war resolved with the 2000 Arusha Peace and Reconciliation Agreement. The post-war position of the country supported and focused on power-sharing constitution leading to peace and progress. Pierre Nkurunziza, who was the Chairman of the National Council for the Defence of Democracy-Forces for the Defence of Democracy (CNDD-

FDD) party, was elected as the first president. Despite an increasing concern of corruption and autocratic governing, Nkurunziza continued for the second presidential term. The Arusha Agreement encloses a two-term limit, however, in April 2015 the CNDD-FDD a new elected the first president for the third term (Boyce and Vigaud-Walsh, 2015; IRRI, 2016). In turn, it has triggered protests, coups and forced displacement crisis, forcing people to flee to the neighbouring countries of Rwanda, Uganda, Tanzania and the Democratic Republic of Congo (DRC). Each neighbouring country has several registration locations, and camps were forced population allocated (Raleigh et al., 2016). Hence, we adopted the current situation of Burundi for the simulation starting from the 1st May 2015 until 31st May 2016. Burundi simulation scenario has eight major conflict locations, starting in capital Bujumbura from the start of the simulation and including other province capitals, namely, Cibitoke, Bubanza, Kayanza and other cities.

In March 2013, the central government of CAR were overthrown by the Seleka group, which represents mainly Muslim population located in the north of the country. Not long after anti-Balaka (Christian militia groups) took over the power from the Seleka group, which has enhanced the conflicts and violent attacks between communities of Muslim and Christians. Subsequently, other communities, such as agriculturalists associated with anti-Balaka and Muslim herders linked with ex-Seleka, increased clashes in CAR. With the escalation of fighting, the situation in the country became unstable and dangerous resulting in forced population displacement internally and to neighbouring countries, namely Chad, Cameroon, DRC and the Republic of Congo (RC). The simulation period for the situation of CAR begins from the 1st December 2013 and ends on the 29th February 2016.

In Mali, we focus on the Northern Mali Conflict that took place in 2012 when insurgent groups began a campaign to fight for the independence of the Azawad region in Northern Mali. This situation is described in detail by Groen (2016). The Northern Mali conflict started on January 16th 2012, when Touareg rebels began conquering places in Northern Mali, and the simulations start on February 29th, the date that the first forced population registrations were done. This scenario has three major conflict locations, starting in Kidal from the beginning of the simulation, and adding Gao and Timbuktu on the 31st of March (day 31). The total number of the forced population registered in these camps reaches a peak of $\sim$145,000 around day 170.

We derive our conflict events from the ACLED, selecting each 'Battle' event from the database. Whenever such an event initially occurs, we mark the selected location as a conflict zone. The occurrence of subsequent events does not alter the status of the location, as locations remain a conflict zone for the duration of the simulation once they have been marked as such. A list of the initial conflict events can be found in Table 5.1 for Burundi, in Table 5.2 for CAR, and Table 5.3 for Mali.

Date	**Day in simulation**	**Conflict location**
1 May 2015	1	Bujumbura
10 July 2015	70	Kabarore
11 July 2015	71	Bukinanyana
15 July 2015	75	Cibitoke
26 October 2015	178	Mwaro
23 November 2015	206	Gisuru
8 December 2015	221	Burambi

Table 5.1: Burundi: Occurrence of conflicts.

Date	**Day in simulation**	**Conflict location**
10 December 2012	*	Ndele
15 December 2012	*	Bamingui
28 December 2012	*	Bambari
18 January 2013	*	Obo
11 March 2013	*	Bangassou
24 March 2013	*	Bangui
17 April 2013	*	Mbres
3 May 2013	*	Bohong
17 May 2013	*	Bouca
7 September	*	Bossangoa
14 September	*	Bossembele
10 October	*	Bogangolo
26 October	*	Bouar
10 November	*	Rafai
28 November	*	Damara
6 December 2013	5	Bozoum
1 January 2014	31	Bimbo
28 January 2014	58	Boda
6 February 2014	67	Kaga-Bandoro
11 February 2014	72	Berberati
11 March 2014	100	Nola
8 April 2014	128	Dekoa
10 April 2014	130	Bria
14 April 2014	134	Grimari
26 April 2014	146	Paoua
23 May 2014	173	Carnot
30 July 2014	241	Batangafo
10 October 2014	313	Sibut

*Conflict zones that occurred before simulation period

Table 5.2: Central African Republic: Occurrence of conflicts.

Date	Day in simulation	Conflict location
3 February 2012	*	Kidal, Timbuktu and Niafounka
29 February 2012	1	Menaka
2 March 2012	2	Tenenkou
13 March 2012	13	Dire
23 March 2012	23	Gao
30 March 2012	30	Bourem and Ansongo
10 August 2012	163	Bamako
1 September 2012	185	Douentza
28 November 2012	273	Lere

*Conflict zones that occurred before simulation period

Table 5.3: Mali: Occurrence of conflicts.

5.3 Overview of forced population displacement models

We present the names of camps located in neighbouring countries of simulation countries in Table 5.4 for the simulation period identified from UNHCR.

Country	Neighbouring country	Camps
Burundi	Tanzania	Nyarugusu and Nduta
	Rwanda	Mahama
	Uganda	Nakivale
	DRC	Lusenda
CAR	Cameroon	East and Adamaoua
	Chad	Belom, Dosseye, Amboko, Gondje and Moyo
	DRC	Inke, Mole, Bili, Mboti and Boyabu
	RC	Betou and Brazaville
Mali	Mauritania	Mbera
	Burkina Faso	Mentao and Bobo-Dioulasso
	Niger	Abala, Mangaize, Niamey and Tabareybarey

Table 5.4: List of existing camps used in simulations.

In addition, several camps open after each conflict has commenced, as the UNHCR data indicate no influx of forced population there prior to a certain date. These include:

- Lusenda (Burundi) on July 30th, 2015;
- Nduta (Burundi) on August 10th, 2015;
- Bili (CAR) on April 1st, 2015.

There are conflict zones and camps connected with intermediate towns for all three countries. We also calculated all links within conflict areas and provided detailed network graphs in Figure 5.1.

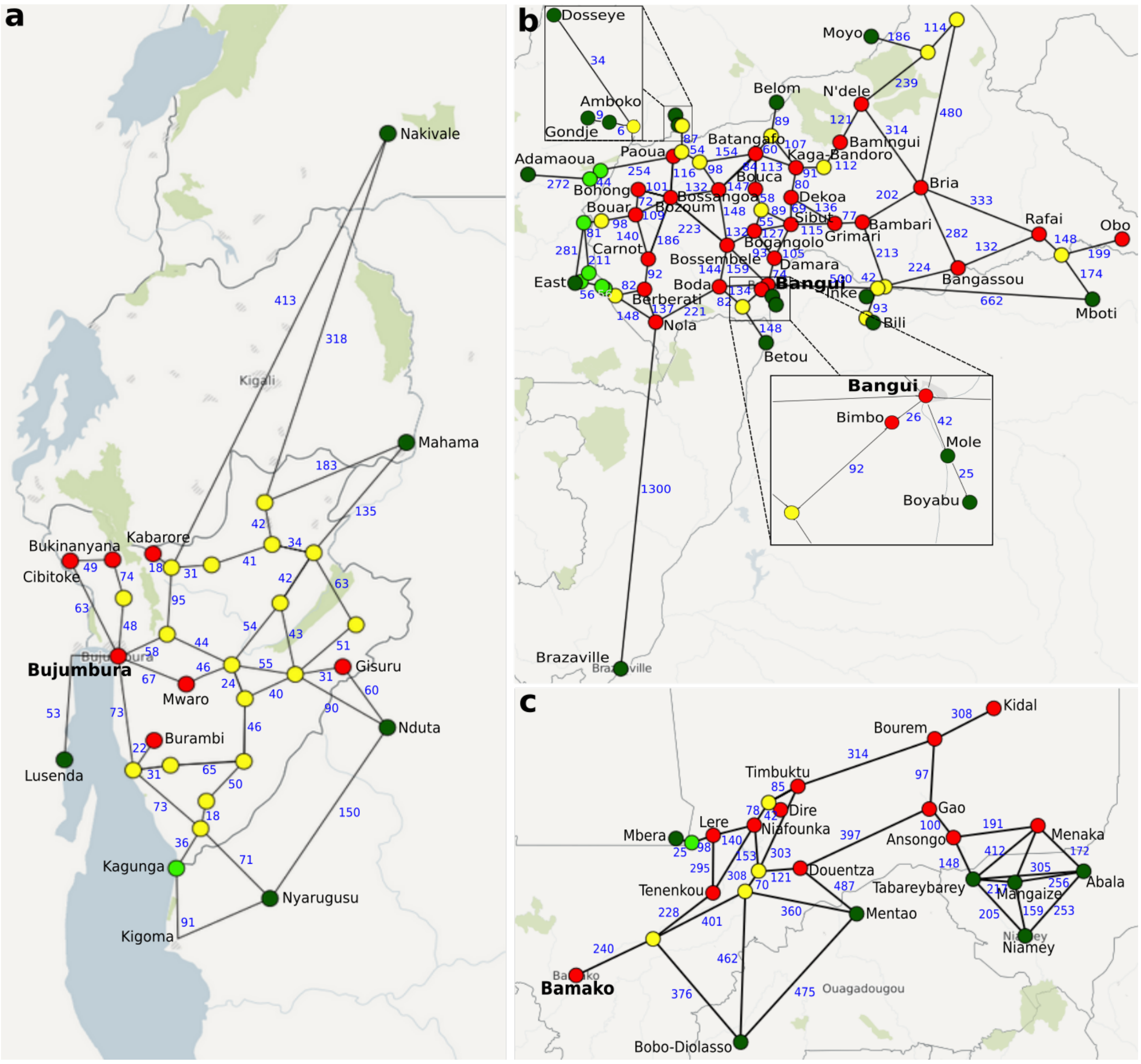

Figure 5.1: Overview of geographic network models for (A) Burundi, (B) Central African Republic and (C) Mali. Models contain conflict zones (red circles), camps (dark green circles), forwarding hubs (light green circles) and other major settlements (yellow circles). Interconnecting roads are given in a simplified straight-line representation, with adjacent blue numbers used to indicate their length in kilometres. Background maps are courtesy of carto.com created using OpenStreetMap data.

These three simulations were also subject to a range of reported border closures. These include:

- CAR - DRC Congo (from the 5th of December 2013 until the 30th of June 2014) (UNHCR, 2014b);
- CAR - Chad (from the 12th of May 2014 onwards) (Reuters, 2014);
- Mali - Burkina Faso (until the 21st of March 2013) (UNHCR, 2012a);
- Mali - Niger (until the 1st of April 2013) (UNHCR, 2012b).

Forced population were redirected from forwarding hubs or camps to other camps in the same country, according to a range of UNHCR reports. These locations include (Table 5.5):

Country	From location	Forward to
Burundi	Nduta	Nyarugusu (prior to August 10th, 2015)
	Kagunga	Nyarugusu
CAR	Gado-Badzere	East
	Lolo	East
	Mbile	East
	Timangolo	East
	Borgop	Ademaoua
	Ngam	Ademaoua
Mali	Fassala	Mbera

Table 5.5: Redirected forced population from forwarding hubs or camps to other destination camps.

The above locations are forced redirection points, which mean that all forced population arriving in this location immediately continue their journey towards the destination. Most of these forwarding points reside within Cameroon in the CAR simulation. Here, forced displacement tends to be scattered across the region, and the forced migrant count is aggregated for the East and Ademaoua regions. To reflect this, we place the smaller camps within the simulation graph but forward any forced population arriving at these smaller camps directly to the main locations.

We summarise parameters used for these three situations in Table 5.6.

Parameters	Burundi	CAR	Mali
Initial number of conflict zones	1 (Bujumbura)	1 (Bangui)	1 (Menaka)
Maximum number of conflict zones	7	27	12
Total number of intermediary towns	17	15	4
Total number of cities in simulation	24	42	16
Maximum number of camps	5	14	7
Camps with later opening	2 (Nduta, Lusenda)	1 (Bili)	-
Forwarding hubs	1 (Kagunga)	7	1 (Fassala)
Number of links	39	80	36
Av.number of new forced migrants per day	688	636	472

Table 5.6: Overview of each country case.

5.4 Error measures

We present several error measures in Figure 5.2, including an overview of the number of forced population in camps according to the simulation and the UNHCR data in (Figure 5.2 a, c and e) and the averaged relative difference in Figure 5.2b, d and f. The averaged relative difference explained in Section 4.7 is less than 0.5 after the first few days, indicating that our simulations accurately predict more than 75% of the forced population movements in absolute terms. In all our runs, the averaged relative difference is lower at later stages of the simulations, with relative differences of 0.1-0.3 or towards the end of all runs.

For Burundi (Figure 5.2 a), our simulations contain substantially fewer displaced people in camps than the UNHCR measurements for the same day. This difference is larger than in other cases and affects the averaged relative difference (Figure 5.2 b), primarily because Burundi is a densely populated country with a large number of settlements in the network graph. However, the difference decreases after Day 5 once substantial numbers of forced population arrive in camps in the simulation, and only increases to a peak around 0.48 on Day 151, due to a coincidence of peak mismatches in both Nyarugusu and Nduta. In the CAR situation (Figure 5.2 c and d) the mismatch in the number of forced population remains relatively small, while the averaged relative difference fluctuates around 0.3. The jump in error around Day 300 is largely due to a sudden large increase in forced displacement in East Cameroon at that time, according to the UNHCR data. In the Mali situation (Figure 5.2 e and f) we see a large but decreasing mismatch at the very start of the simulation. It is because the Fassala camp is technically not defined as a camp within our simulation, as from the start of the simulation period forcibly displaced people were already redirected from Fassala to Mbera. However, Fassala is considered to be a camp according to the data. After Day 30, the number of forced migrants in camps in the simulation is relatively close to the reported number, and the averaged relative difference remains relatively constant.

To our knowledge, there are no other prediction techniques that have been previously applied in this setting. However, it is possible to perform naive predictions, extrapolating future behaviour from historical data, after a conflict has started. To measure the added value of our prediction approach, we here present a comparison of our method against a set of naive prediction models. We compare the accuracy of our method by obtaining MASE relative to the six other techniques.

For comparison purposes, we have created three different types of naive models, namely, 0th order (flat) extrapolation, 1st order (sloped) extrapolation and extrapolation by a ratio (fraction). We define these three types in the next paragraphs and note that all rely on some section of historical data to extrapolate values in the future. While our simulation approach can be used from Day 1 to provide a prediction of camp forced populations, we can only apply naive models after a number of days have elapsed. It is because naive models extrapolate from past data, and such data can only be acquired after the conflict has started and forced population have departed.

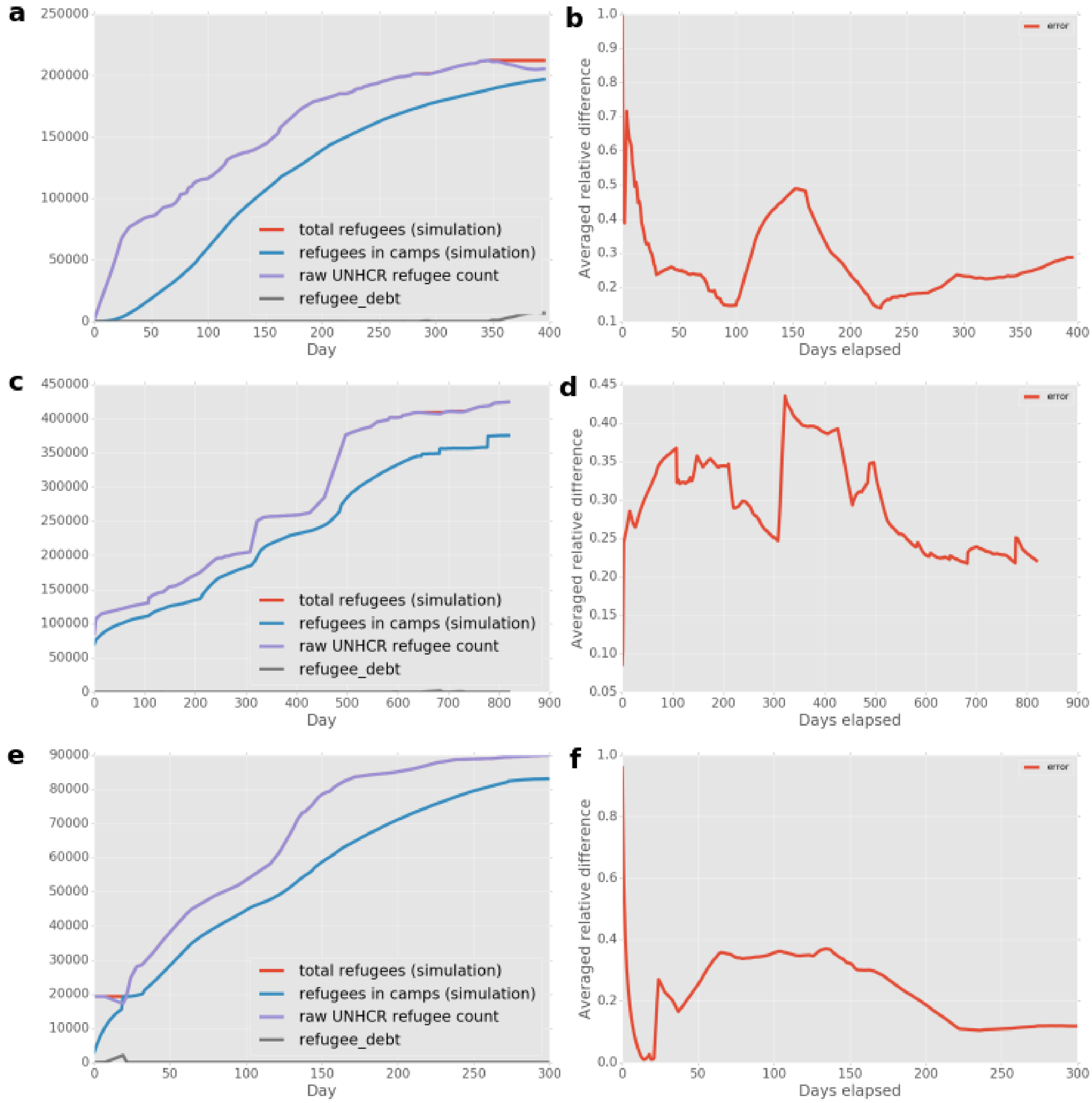

Figure 5.2: Comparison of number of forced population in camps between the simulation and the data (left column), and overview of the averaged relative difference between simulation and data (right column). The averaged relative difference across camps between simulation and data is given by the red line. These comparisons are respectively for (**a, b**) the Burundi simulations (top row), (**c, d**) the CAR simulations (middle row) and (**e, f**) the Mali simulations (bottom row).

We compare our approach, as described in the Analysis section of Chapter 4, against naive model predictions that take place respectively seven days, and 30 days after the starting date of the respective simulation periods. We argue that a week is required to obtain sufficient data to apply any meaningful extrapolation. However, naive models that require more than a month before they can be applied are arguably of little use, as many of the initially forced population movements have already taken place by then (particularly in the case of the Burundi conflict). It should be noted that the collection of forced population registrations is by no means an instantaneous process, and any time overhead in obtaining such would further delay the application of these naive models.

For each camp location in each conflict, we have applied the following three types of naive prediction:

1. 0th order (flat) extrapolation: Here, we take the forced migrant count on either day 7 or day 30 in each camp, and assume that this number does not change over time.

2. 1st order (sloped) extrapolation: Here, we take the forced population count on either day 7 or day 30, as well as the registration count on day 0. We then linearly extrapolate future values in time from these two registration counts.

3. Extrapolation by a ratio (fraction): Here, we take the forced migrant fraction in a given camp, which we calculate by dividing the forced population count in a given camp, on either day 7 or day 30, by the total number of forced population across all camps on that same day. We then forecast forced population counts in each camp by assuming that this forced migrant fraction remains constant over time, and predict future value by taking that fixed fraction of the total forced population (which is a known quantity in our setting) over time.

We present the results from our comparison in Table 5.7. In all cases, our prediction approach results in a lower averaged relative difference than the naive prediction models. We obtained MASE scores of 0.0639-0.942 (Burundi), 0.0367-0.705 (Central African Republic) and 0.116-0.513 (Mali).

Run name	MASE (7-day naive model)			MASE (30-day naive model)		
	flat	slope	fraction	flat	slope	fraction
Burundi	0.279	0.144	0.942	0.443	0.0639	0.791
CAR	0.585	0.705	0.452	0.639	0.0367	0.341
Mali	0.23	0.127	0.503	0.314	0.116	0.513
Weighted average	0.453	0.473	0.598	0.542	0.0544	0.491

Table 5.7: Comparison of our prediction approach against six naive models for each of our three conflict simulations. Simulations were run with the same settings. We report on MASE for each of the naive models in columns 2 to 7. Here, values below 1 indicate that forecasts using our prediction approach resulted in a smaller averaged relative difference than those relying on that specific naive model. In the bottom row we provide a weighted average of the MASE score across the three conflict simulations, with the weightings based on the maximum number of forcibly displaced people in each conflict (205445 for Burundi, 424496 for CAR, and 89991 for Mali).

5.5 Results and discussion

We present a generalised prediction approach to estimate the distribution of incoming forced population across destination camps. Accurate predictions can help save migrants' lives, as it helps governments and NGOs to allocate humanitarian resources correctly to camps before the (often malnourished or injured) forced population themselves have arrived. To our knowledge, we are the first to attempt such predictions across multiple major conflicts using a single simulation approach.

Although our approach has evident limitations, in part due to the simplicity of our model and gaps in the empirical data, it does enable us to reproduce the key forced population movement patterns in each of the conflicts and correctly predicts at least 75% of the forced displacement destination distribution in almost all cases.

We illustrate the number of forced migrants in each camp listed in Table 5.4 across three African countries discussed above. We compare our simulation predictions to forced migration data from the UNHCR database for the simulation period, both according to our prediction approach and according to the empirical data from UNHCR for all three conflict situations.

The first country to discuss is Burundi. We present our simulation predictions and the UNHCR forced population counts for the Burundi conflict in Figure 5.3. Within the camps in Nyarugusu, Mahama and Nakivale, our simulation results accurately capture the key growth trends in forced displacement. Our approach does underpredict the forced population growth in Mahama, as there is a delay in forced population arrival due to the many non-conflict settlements between Mahama and the conflict zones.

Both the Nduta and Lusenda camps opened only after the start of the period of simulation.

Nduta was only established as a camp on the 10th of August 2015 (day 101), after Nyarugusu became overpopulated. In the case of Nduta, our simulation shows a small population travelling at the start (when the location was not yet a camp), and a steep population increase to 30,000 during the 90 days after the camp is opened. Lusenda, which opened on day 90, quickly fills to capacity in the simulation, whereas a more gradual increase can be observed in the data. Here, the mismatch could be due to delays in the UNHCR registration process, as virtually no forced population were registered correctly in the whole of DRC prior to the 30th of October 2015 (day 182).

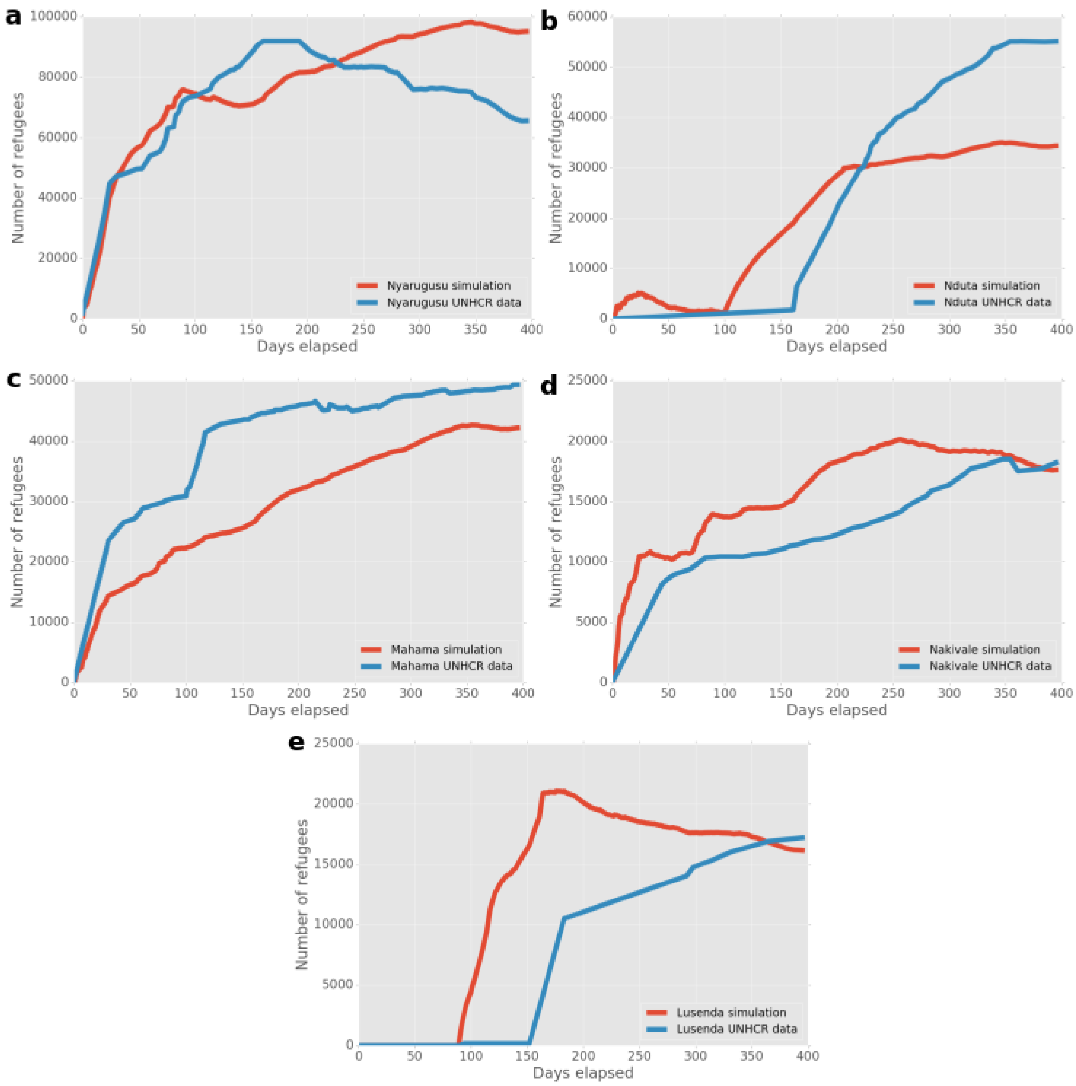

Figure 5.3: Number of forced migrants as predicted by forced displacement simulation and obtained from the UNHCR data for the Burundi conflict. (**a-e**) Graphs are ordered by camp population size, in descending order.

In Figures 5.4 and 5.5, we present the number of forced population in camps for the CAR conflict simulation. Our simulation predictions closely follow the trends observed in the data for the two largest camps, East Congo and Adamaoua. Here our simulation underpredicts the total forced population in East Congo by about 35,000 ($\sim$20%), and overpredicts the population in Adamaoua by about 23,000 ($\sim$30%).

The camps in DRC (Inke, Mole, Boyabu and Mboti) were subject to border closures between CAR and DRC from the 5th of December 2013 (simulation day 4), until the 30th of June 2014 (day 211). It is reflected by a period of relatively stable forced populations both in the simulation and in the data. Bili also is located within DRC, but was established only after the border was reopened.

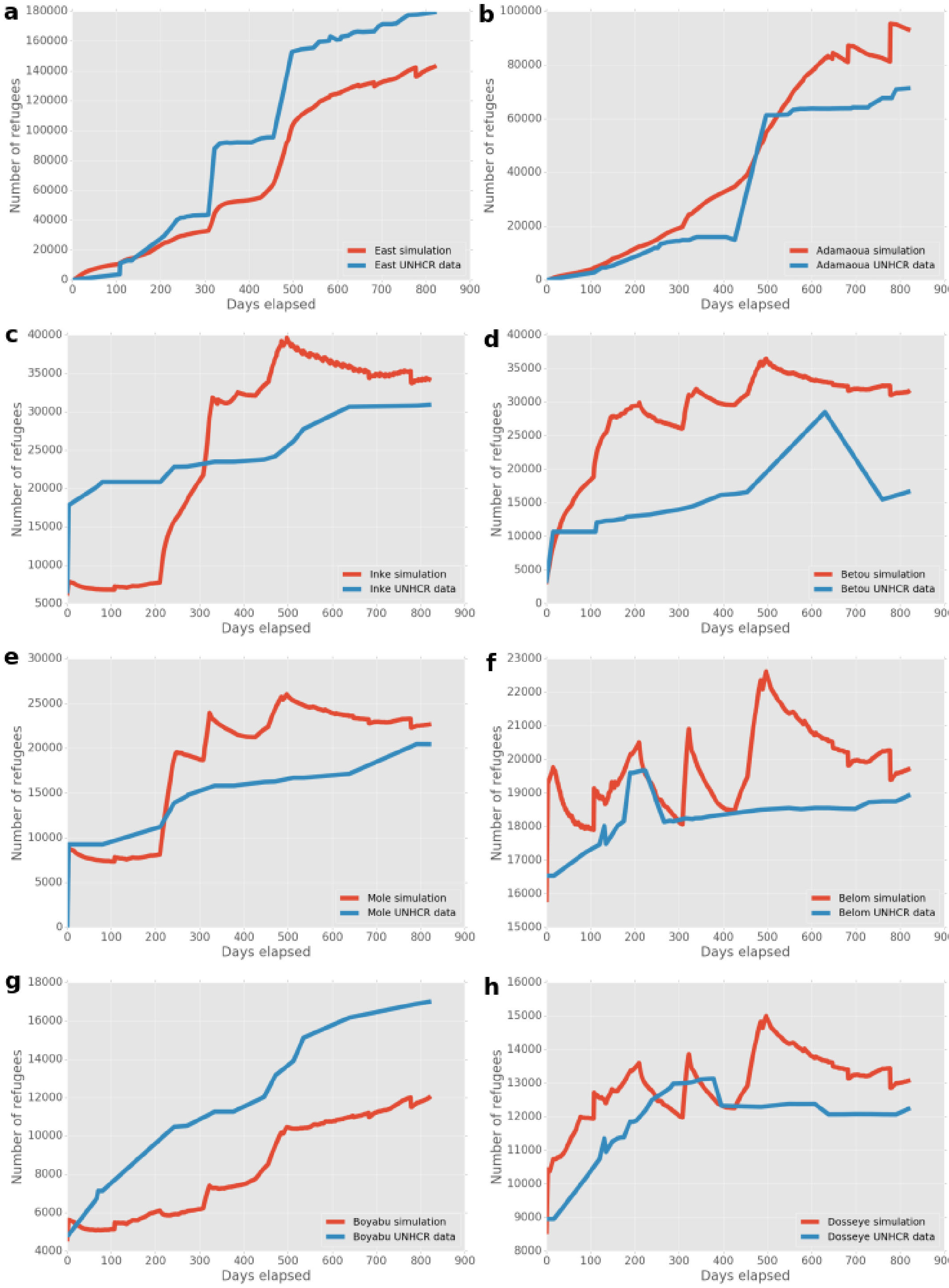

Figure 5.4: Number of forced migrants as predicted by forced displacement simulation and obtained from the UNHCR data for the CAR conflict. (**a-h**) Graphs are ordered by camp population size, in descending order.

The predicted forced population counts in the Chad camps (Amboko, Belom, Dosseye and Gondje) are in close agreement with the data, except that large fluctuations occur during the simulation after the border closure on the 12th of May 2014 (day 163). At this time all the camps are close to full occupancy, which results in forced population moving from between the camps and the city of Gore, a city in Chad which lies in close proximity to the camps.

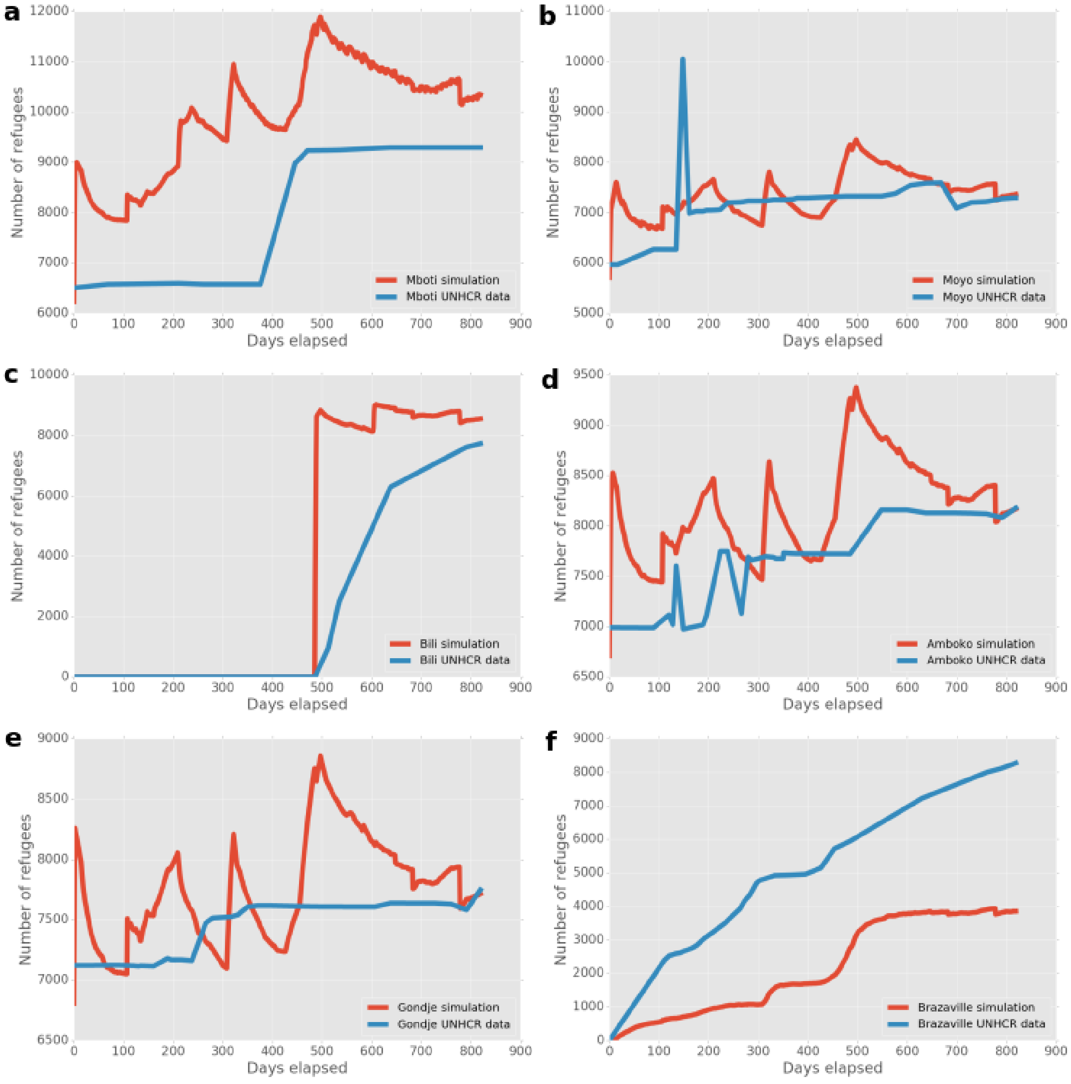

Figure 5.5: Number of forced migrants as predicted by forced displacement simulation and obtained from the UNHCR data for the CAR conflict. (**a-f**) Graphs are ordered by camp population size, in descending order.

The Betou camp in Congo is another example of a camp close to the conflict areas, and it also fills up quickly in the simulation. The Brazaville location is far removed from the conflict zone, and here, our simulation underpredicts the forced population. It could be that the size of the city of Brazaville may increase its attractiveness as a forced displacement destination. We did not incorporate this factor in the runs presented here, but we do wish to examine it in future simulation studies.

In the case of Mali, Figure 5.6 presents the number of forced migrants in camps around

Mali over the 300 day simulation period. Our simulation results are in close agreement with the data for the two largest camps. The maximum differences here are an underprediction of

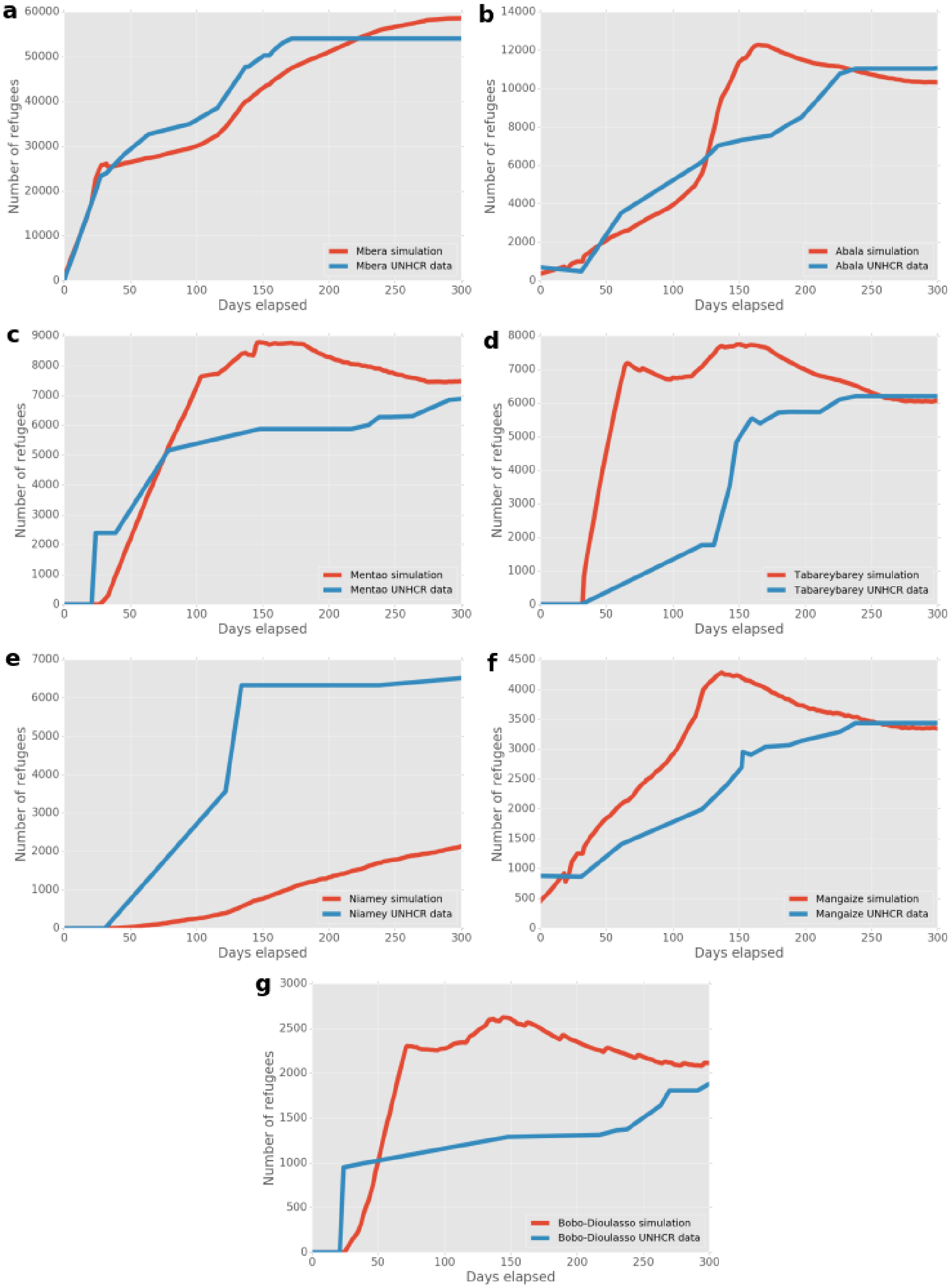

Figure 5.6: Number of forced migrants as predicted by forced displacement simulation and obtained from the UNHCR data for the Mali conflict. (**a-g**) Graphs are ordered by camp population size, in descending order.

7,000 (∼18%) for Mbera around day 135, and an overprediction of about 4,500 (∼60%) for

Abala around day 160. Tabareybarey, Niamey, Mentao and Bobo-Dioulasso were established once the conflict was already underway. Tabareybarey and Niamey camps have forced population for simulation and data from day 30, whereas the camps in Burkina Faso, Mentao and Bobo-Dioulasso reopened their previously closed borders on the 1st of April 2012 (day 32).

The simulation predicts a fast-paced growth of forced population for both Mentao and Bobo-Dioulasso, while the data features a sudden spike in forced population arrivals around day 30 in these camps. The simulation predictions for Mangaize in Niger are in line with the data, though slightly higher. The large inflow early in the simulation is primarily due to the close proximity of Mangaize to one of the early conflict zones (Menaka). Our simulation results do not accurately match the data for Tabareybarey and Niamey. Niamey is not directly connected to regions in Mali, due to two other camps being located along the way. However, Niamey is a large capital city (like Brazaville in the CAR simulation), which may be the reason why more forced population choose that destination than our simulation predicts. In general, our predictions overestimate the forced population inflow into the three border camps in Niger. A significant cause here may be the presence of partial restrictions for crossing the Niger border during the conflict (Groen, 2016).

5.6 Conclusion

Using our approach, we have reproduced the key forced population movement patterns in each of the three conflicts and correctly predicted at least 75% of the forced population movement destinations in all these conflicts after the first 12 days. In the Burundi conflict, our approach correctly predicts the largest inflows in Nyarugusu, Mahama and Nakivale during the early stages of the conflict. In CAR, our prediction approach correctly reproduces the growth pattern in East Congo, as well as the stagnation of forced population influx in the Chad camps. In the case of Mali, our predictions accurately capture the trends in the data for both Mbera and Abala, which together already account for $\sim$75% of the forced population.

As a result of conducting this study, we discovered several important issues and limitations. For example, our model omits a range of factors which are considered important according to the empirical literature, but for which we could not find accurate and tractable means to convert empirical conclusions to simulation parameters. In some cases, such as GDP and presence of existing conflicts, the significance of these factors has been confirmed on a country-by-country level, but not on a city-by-city level (Moore, 2006; Moore and Shellman, 2007).

In other cases, such as religion and ethnicity, we did not find reliable statistical information on a local level for these conflicts. Some parameters, such as the level of knowledge of agents about the surrounding region, were found to have little effect on the simulation results beyond being aware of adjacent locations. The obtained averaged relative difference changes little when we adjust maximum movement speed of forced population to values less or more than 200 kilometres per day. In general, empirical data collection during these conflicts is very challenging, in part due to the nature of the environment and in part due to the severe and structural funding shortages of UNHCR emergency response missions. Both CAR and Burundi are among the most underfunded UNHCR forced population response operations, with funding shortages of respectively 76% and 62% (UNHCR, 2015a). More funding for these operations is bound to save human lives, has the side benefit of providing more comprehensive empirical data, and thereby enables the validation of more detailed prediction models.

Yet, important steps have been made in recent years, as the combination of a conflicts database (Raleigh et al., 2010), a public UNHCR forced displacement data repository, and a sophisticated mapping platform enabled us to do this work. Given the increasing effort in collecting forced population data, and increasing recognition for data science, we are confident that ongoing advances in data collection will accelerate future research efforts on simulating forced population movements.

Chapter 6. Automated Simulation Construction

Based on:

Suleimenova, D., Bell, D. and Groen, D. (2017), "Towards an automated framework for agent-based simulation of refugee movements". In Proceedings of the 50th Winter Simulation Conference (WSC17), edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, pp. 1240-1251, Las Vegas, Nevada, December 3-6.

6.1 Introduction

In this chapter, we propose an automated SDA assisting researchers and organisations to more easily construct and execute forced displacement simulations. We require an automated SDA since conflicts forcing people to migrate can erupt at any time and create urgent situations. Particularly, we automate SDA phases comprising data collection, model construction, simulation execution and analysis of simulation results. Our approach incorporates a diverse range of data sources, and uses the FabSim3 toolkit in conjunction with the FLEE simulation code to quickly generate simulation workflows. It is rapid, consistent, efficient, and saves efforts in developing conflict situation models. The next sections present existing automation tools and their application, as well as an automated SDA for forced population simulations.

6.2 Automation tools

A valid simulation development is time-consuming and challenging. Hence, we automate our SDA to improve the simulation process and provide better integration of applications. To facilitate the automation process, we explore existing computational advancements that aim to solve complex problems.

There is a wide range of languages, open-source software and automation tools, which assist scientists and programmers in developing computational research. To demonstrate, domain-specific languages (DSLs) are small and specialised for a specific aspect of a software application. The use of DSLs, such as cascading style sheets (CSS) or application program interfaces (APIs), helps developers to build programs written in general-purpose languages. Mernik et al. (2005) provide detailed guidance for design, analysis, and implementation processes of DSL development for developers. Alternatively, there is a package management software named Homebrew, which helps to instantly install the required software and developed for Macintosh operating systems. It is integrated with the command-line and favoured for its ease of use.

Another example of existing computational advancement is the swift parallel, which scripts engine is reducing complexities of structuring file systems (Wilde et al., 2011). It provides a basis for writing and executing codes or programs across dispersed computing resources from computer clusters to supercomputers. There is also Longbow tool to run on high-performance computing (HPC) resources, which automatically stages input data sets and executes large simulation results. According to Gebbie-Rayet et al. (2016), Longbow provides an opportunity for users to simply and instantly run tasks on HPC resources using their own desktop environment.

In terms of workflow-specific automation engines, Kepler provides a platform to operate with various data formats, merge software components and execute locally or online to generate results. In addition, Kepler and other workflow engines, such as Taverna, simplify data access and flow of control, as well as ensemble distribution of databases with remote machines. Curcin and Ghanem (2008) discuss several workflow systems and distinguish them in terms of their flow of control and data flow characteristics.

Importantly, Groen et al. (2016) propose a highly modifiable automation toolkit named FabSim, which has a purpose of time management by automating and simplifying a range of computational activities. For example, researchers save their time as FabSim helps them construct, manage and organise input and output files, user and machine configurations, as well as application executions using remote resources, all using one-line commands. It also consists of a software toolkit that is easy to use, navigate, explain and adjust. Hence, the main strength of this automation toolkit is its ease of customisation in structuring models, helping to maintain computational tasks.

To provide easy remote access, FabSim relies on low-level secure shell (SSH) software that is used on most Unix-based machines and supercomputer, and is written in Python 2. It does not require any administrative or additional heavyweight installations, and therefore can be easily applied to perform activities on remote resources. FabSim also uses the Fabric library to access and manage remote machines conveniently while using the YaML library as a platform to provide a dense, instinctive and human-readable data structure. The overall construction of FabSim automation toolkit is illustrated in Figure 6.1.

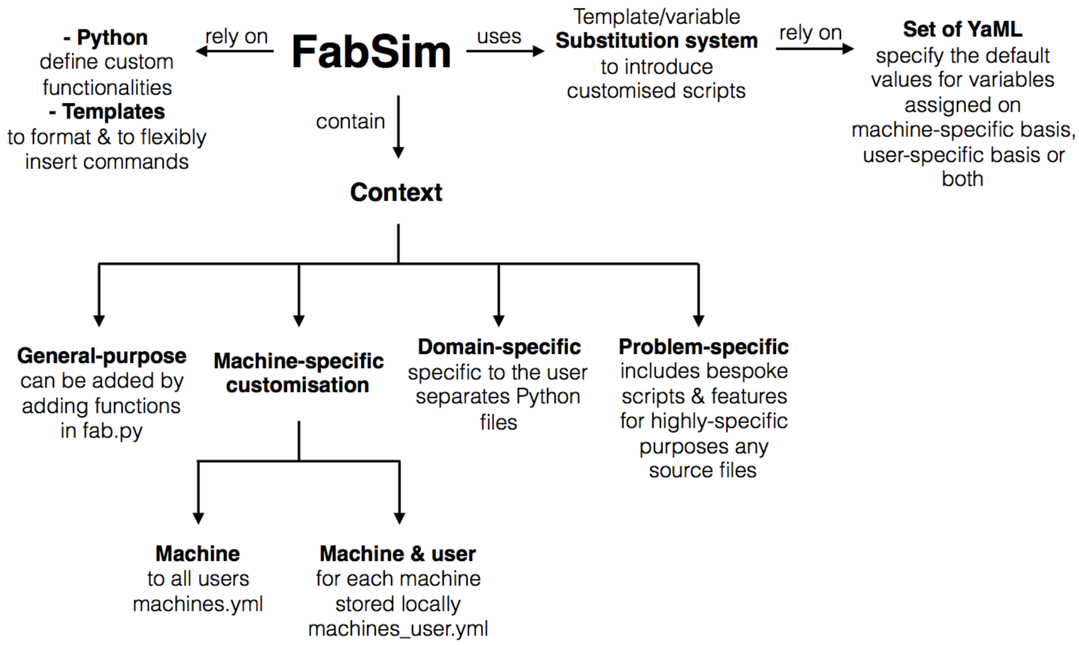

Figure 6.1: FabSim construction.

It has already been applied successfully across three different disciplines, including computational fluid dynamics, materials science problems, and biomolecular interactions. It is unique in its primary focus on facilitating *workflow modification*, and extending this focus beyond composition activities into operations that change toolkit itself. Indeed, when applied to specific scientific problems, FabSim is frequently renamed (e.g., to FabHemeLB (Groen et al., 2013) or FabMD (Suter et al., 2015)) to reflect the depth of the adaptations made by the researchers.

There is an improved version of FabSim written using Python3, namely FabSim3 (`github.com/djgroen/FabSim3`). It currently contains an integrated test infrastructure, more flexible customisation options using a plugin system, and a range of additional in-code documentation and examples to improve usability (see Appendix B for installation instructions).

FabSim3 helps to create multiscale models from small to large scales (see Figure 6.2). To begin with, there is an evacuation modelling, which focuses on forced displacement within a designated area and fleeing conflicted town. Specifically, we try to detect where forced population movements happen. Next, there is a forced displacement modelling constructed using the FLEE code that predicts forced population movements from a home country towards camps in destination countries. Finally, migration modelling focuses on the movements of people across countries. Although these three model types are quite different fields of study, there is the discrepancy in time scales between them.

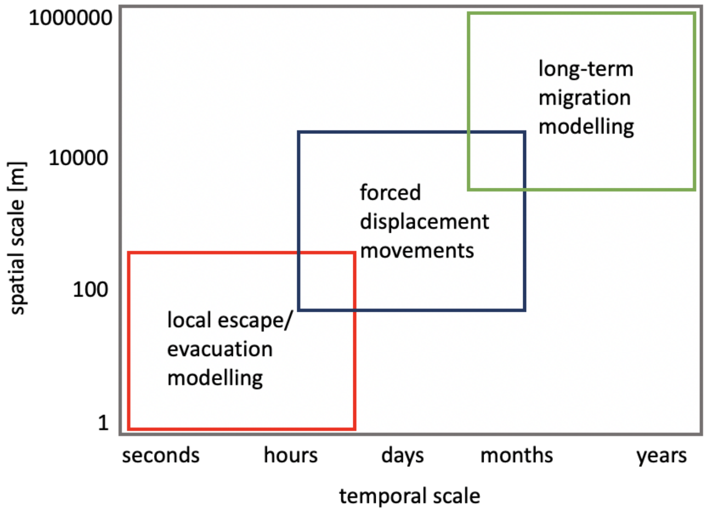

Figure 6.2: Multiscale forced displacement modelling (Groen, 2016).

6.3 Automated forced displacement simulation toolkit

We propose a FabFlee toolkit (https://github.com/djgroen/FabFlee.git), which is a combination of FabSim3 and the FLEE simulation code, as well as a plugin for automated implementation of our SDA for forced displacement simulations. In Figure 6.3, we present the SDA phases with automated functionality from model construction to analysis. Since manual extraction of data sources, construction of network maps, exploration of different parameters, and execution of multiple runs are time-consuming and error-prone procedures and require careful scrutiny. Simulations are also limited in reusability, and thus, it is essential to build a simulation with reusable components (Swaminathan et al., 1998).

FabFlee is a unique and useful simulation addition as automation toolkit can create the whole environment for researchers and organisations to curate data, add data processing components, construct models and modify simulations, instantiate and execute multiple runs, validate and visualise the obtained results against the existing data, but mainly to predict the distribution of incoming forced population across destination camps. In the next sections, we provide a detailed description of automation applied to each phase of the SDA. The FabFlee plugin installation and execution instructions are in Appendix B.

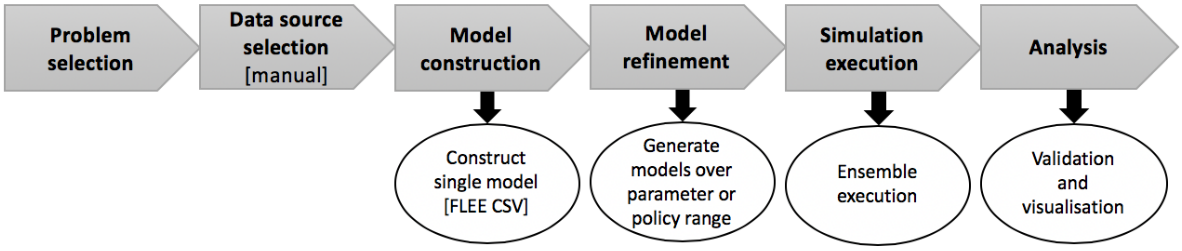

Figure 6.3: Phases of our simulation development approach,given in arrow boxes, and automation implemented in FabFlee for each phase, described in the ovals.

6.3.1 Data collection

We extract data from three main sources, namely UNHCR, ACLED, Bing Maps and population databases. We model forced displacement in a conflict crisis, and each of these sources provides essential information for simulation construction. In particular, UNHCR database identifies countries with previous or current forced displacement crisis, lists camps located in neighbouring countries and keeps a record of forced population numbers for each camp in JavaScript Object Notation (JSON) formats. In turn, ACLED is a database providing detailed information on conflicts and protests for African and Asian countries that can be downloaded in eXceL Spreadsheet (XLS) formats. Moreover, City Population database establishes population distributions of major cities and intermediate towns within conflict areas. WorldPop is also a population distribution database that allows mapping population distribution with our simulation.

As raw data sources explain data diversity, a data converter examines efficient and effective ways of gathering input files from all databases. To achieve this, firstly, we investigate UNHCR and ACLED APIs with some complications. Specifically, we found that the current UNHCR operational portal for forced displacement situations has API documentations but they appear to refer to an older version of UNHCR platform. Similarly, some of the conflict situations still under old UNHCR API, whereas other situations use new API codes. Henceforth, we present in Table 6.1 the sequence to follow for both versions of UNHCR APIs.

	Web API addresses	Description
Old UNHCR API	1. http://data.unhcr.org/api/instances/list.json 2. http://data.unhcr.org/api/regions/show.json?id=countryname E.g. http://data.unhcr.org/api/regions/show.json?id=mali	1. *instance_id* provides country code (or countryname) 2. this provides list of camps for each country e.g. *mali* is a countryname
New UNHCR API	1. https://data2.unhcr.org/en/search 2. https://data2.unhcr.org/api/population/get/timeseries?... ...geo_id=(number)&frequency=day&population_collection=10 E.g. ...geo_id=933&frequency=day&population_collection=10	1. view source page for *geo_id* that shows number codes for all cities 2. change *geo_id* to get camps counts e.g. *geo_id=933* is Lusenda camp

Table 6.1: An old and new UNHCR APIs for camp names and forced population counts.

As for the ACLED data, there are XLS files to download for each conflict situation. However, there is information within these files that do not have value for our simulation construction. Hence, ACLED API documentation provides descriptive guidelines that return the data used for forced displacement simulation. For instance, `https://api.acleddata.com/acled/read.csv?gwno=432&event_type=battle&fatalities_where=%3E&fatalities=0` illustrates an API code for Mali crisis. Notably, *gwno=432* is the country code corresponding Mali, *event_type=battle* filters the data and determines the occurrence of conflicts with event type being battle and *fatalities_where=%3E&fatalities=0* modifies the nature of the query providing fatality rate greater than 0.

In Figure 6.4, we demonstrate our data collection phase that follows from raw data sources to a unified input data files. We simplify the data collection phase by creating reader modules for CSV formats for input data. Three formats of CSV files, namely *locations.csv*, *routes.csv* and *closures.csv*, are integrated with FLEE's input interface.

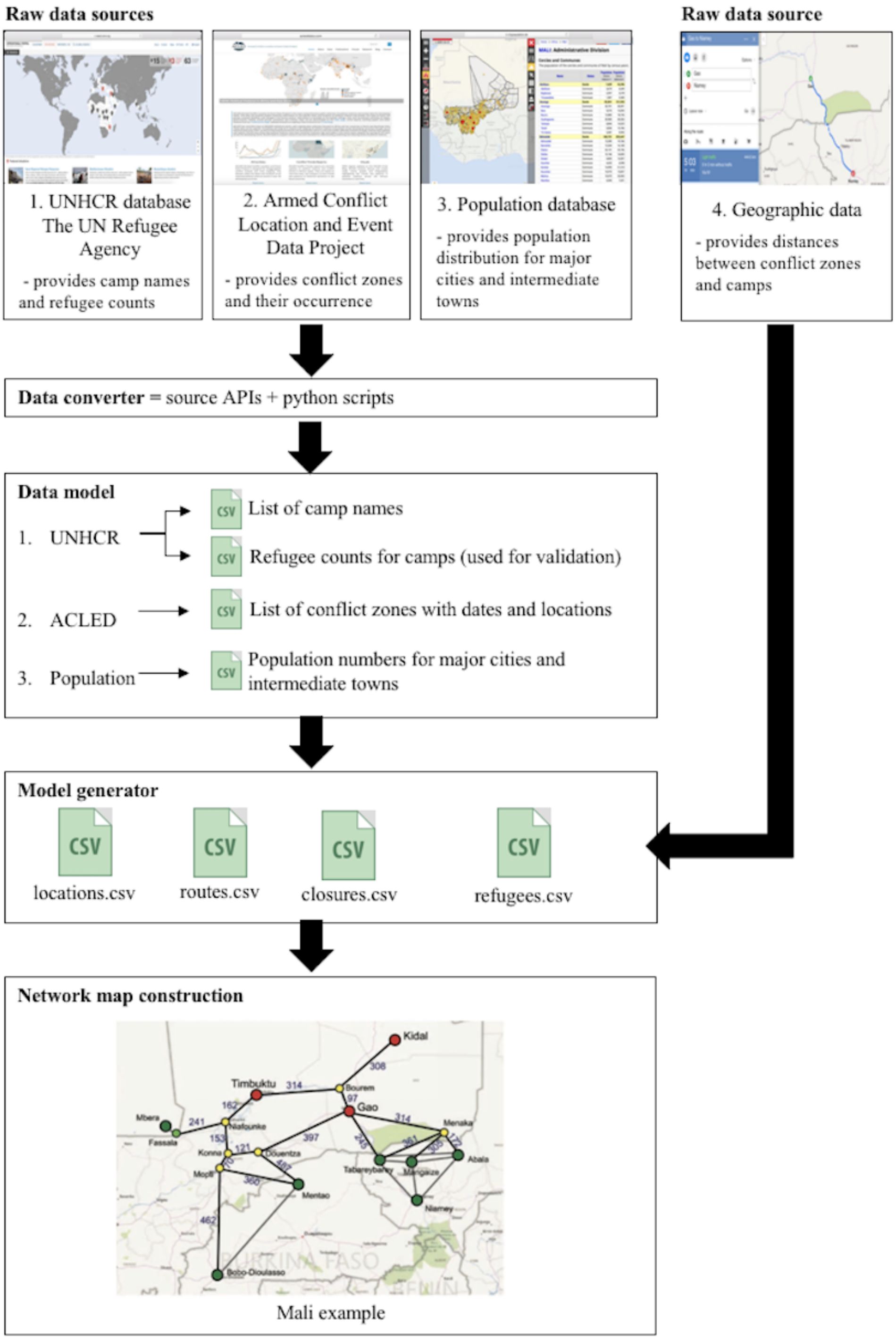

Figure 6.4: The model building approach for forced displacement, conflict and population data.

We create these CSV files manually according to the formats demonstrated in Tables 6.2, 6.3 and 6.4 for the conflict scenario and store them under the base conflict data. Importantly, we create the *routes.csv* file using Bing Maps geographical data source to provide distances between conflict zones and camps. These CSV files reduce our data collection time and are easy to use in the model construction phase.

name	county	country	latitude	longitude	location_type	conflict_date*	population/capacity
					conflict		population of location
					town	-	-
					camp	-	camp capacity
					forwarding_hub	-	-

Table 6.2: locations.csv contains all locations with the properties required for simulation construction, such as name and geographical information of locations, and populations (for non-camp locations) or capacities (for camp locations). *Note:* conflict_data is given as an integer, counting the number of days after the simulation start. The value of 0 indicates the start, while -1 indicates the end date of the simulation.

location1	location2	distance [km]	forced_redirection*
			0
			1
			2

Table 6.3: routes.csv specifies distances between two locations. *Note:* forced_redirection refers to redirection from source location (can be town, camp or forwarding_hub) to destination location (mainly camp). The value of 0 indicates no redirection, 1 indicates redirection from location2 to location1 and 2 corresponds to redirection from location1 to location2.

closure_type*	name1	name2	closure_start*	closure_end*
location				
country				

Table 6.4: closures.csv provides camp closure event specifying locations names or border closure event requiring country names to name1 and name2 respectively. *Note:* closure_type can be two types: location corresponding camp closure and country referring to border closure. closure_start and closure_end are given as integers, counting the number of days after the simulation start. The value of 0 indicates the start, while -1 indicates the end of the simulation.

We construct network maps with extracted data from sources that provide conflict, camp and intermediate locations. We use these CSV files with specific information employing the CARTO map construction platform. At present, we can upload CSV files with location names and geographic coordinates to CARTO, which instantly identifies and marks required locations. Although it reduces the search time for conflicts, camps and intermediate town allocations, we still manually link and determine distances between these locations. We provide a step-by-step model construction tutorial for forced displacement in Appendix A.

6.3.2 Model construction

After the generation of *locations.csv, routes.csv* and *closures.csv* files, we follow Figure 6.5 to construct the initial model using CSV formats, refine the model with a new set of parameters or policy decisions, execute an automated ensemble of runs and analyse the obtained results with the use of automated plotting tools.

We load a base conflict data which includes CSV files and the source data of a conflict scenario using *load_conflict* command as presented in Figure 6.5. In turn, it duplicates all existing files from a base conflict directory to a working directory, namely active conflict data. The load command also generates a text file (i.e. *commands.log.txt*) that records command logs of commencing activities.

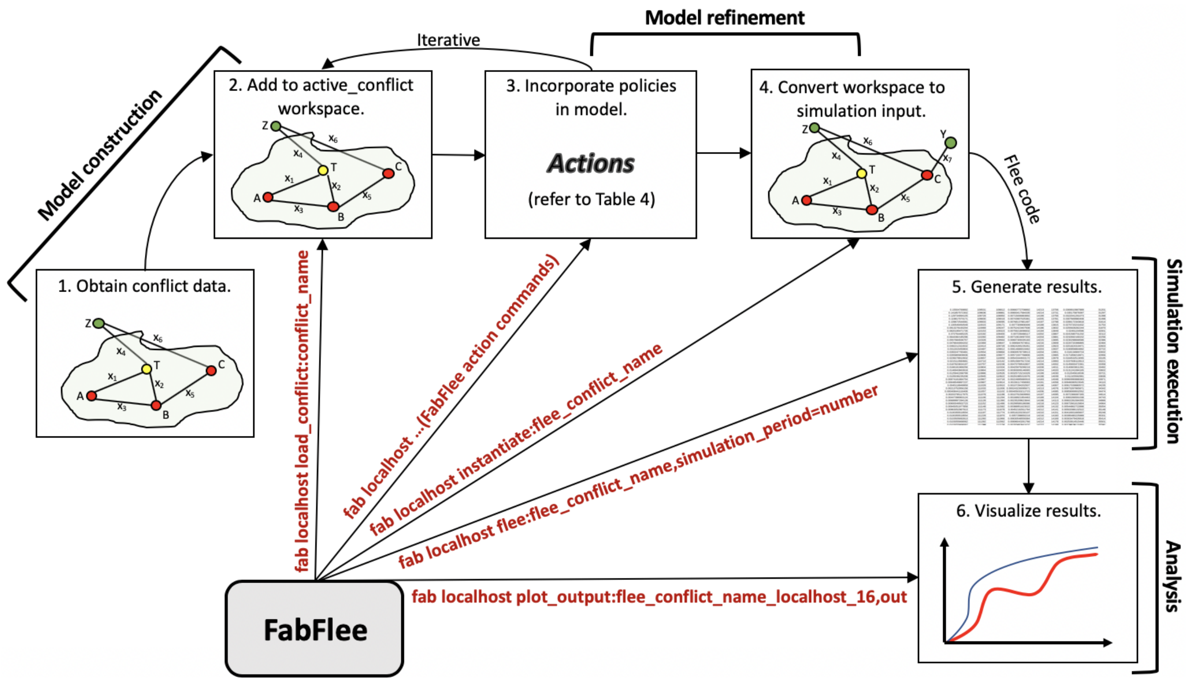

Figure 6.5: FabFlee workflow diagram demonstrating automated simulation steps from construction to analysis.

6.3.3 Model refinement

To refine the initial model, we examine policy implications through parameter explorations related to a forced displacement emergency. We have developed several parameter exploration commands to modify a range of parameters illustrated in Table 6.5. We use the FabFlee toolkit that provides more systematic, quick, simple and time efficient refinement of our model.

Actions	**FabFlee command**
change camp capacity	change_capacities:camp_name=capacity
add a new location	add_camp:camp_name,region,country,lat,lon
delete an existing location	delete_location:location_name
camp closure	close_camp:camp_name,country,closure_start,closure_end
border closure	close_border:country1,country2,closure_start,closure_end
forced redirection	redirect:source,destination,redirect_start,redirect_end

Table 6.5: FabFlee functions for policy decision exploration.

Following the refinement phase, we duplicate parameter changes of the model by running the instantiate command. The instance is then saved in a new directory, which can include run name, version and date of instantiation on users insert choice. To create a clean slate for other refinements, we can clear the active conflict directory using

fabsim localhost clear_active_conflict,

upon which we can reload the conflict and change other parameters (and instantiate and run a new simulation).

6.3.4 Simulation execution

To run instantiated forced displacement simulations (briefly FLEE jobs), we execute the following command triggering the FLEE code and producing simulation results:

fabsim localhost flee:<conflict_name>,simulation_period=<number>,

In turn, it copies the job input (in config_files directory), to the remote location specified in FabFlee deployment files and duplicates the input to the remote results directory, as well as executes a simulation run and generates results. In this execution command, *<conflict_name>* refers to a conflict instance generated after refinement phase and *<number>* is a simulation period of a conflict scenario.

We can also run an ensemble of FLEE jobs on local machine as it offers an efficient way to execute instantly a vast number of forced displacement runs using the following command:

fabsim localhost flee_ensemble:<conflict_name>,simulation_period=<number>.

Alternatively, ensemble simulations can be executed using large supercomputers. Specifically, the Quality in Cloud and Grid (QCG) Pilot Job mechanism `github.com/vecma-project/QCG-PilotJob` provides two interfaces that may be used interchangeably. The first one allows to specify a file with the description of sub-jobs and execute the scenario in a batch-like mode, conveniently supporting static scenarios. The second interface is offered with the REST API, and it can be accessed remotely in a more dynamic way. It supports scenarios where a number of replicas and their complexity dynamically changes at application runtime. A Pilot Job is a container for many sub-jobs that can be started and managed without having to wait for individual resources to become available. A Pilot Job may serve several defined sub-tasks and execute ensemble runs, and can be submitted using:

fabsim qcg flee_ensemble:<conflict_name>,simulation_period=<number>,Pilot=true.

6.3.5 Analysis

We perform simulation analysis to interpret, validate and visualise obtained simulation results. For evaluation purposes, we compare simulation results against the UNHCR data using *plot-flee-output.py* Python script. It constructs and generates a graphical visualisation for each camp in a destination country, as we presented in Chapter 5 for three African conflicts.

The initial graphical results provide a basis for further analysis, such as sensitivity analysis varying agent awareness levels, speed limits of forced population movements and other policy exploration parameters. Hence, we analyse each of these agent-specific parameters, such as move speed and awareness level, as well as the simulation duration. Groen (2016) executes multiple simulations runs with various travel speeds, forced population spectrum of locations awareness and simulation periods of conflict crisis manually, which is very time-consuming. To automate this, we present *fabsim localhost test_sensitivity* FabFlee function (see Table 6.6 for more details) with a range of forced population speed limits and awareness levels of inter-connecting links. These sensitivity tests are executed manually and one at a time.

Sensitivity test	**FabFlee command**
forced population move speed	fabsim localhost test_sensitivity:flee_conflict_name,simulation_period=number,name=MaxMoveSpeed,values=50-100-200
forced population awareness level	fabsim localhost test_sensitivity:flee_conflict_name,simulation_period=number,name=AwarenessLevel,values=0-1-2

Table 6.6: FabFlee functions for sensitivity test analysis.

EasyVVUQ

To improve the automation of sensitivity analysis further, we combine the FabFlee toolkit and EasyVVUQ, which facilitates verification, validation and uncertainty quantification (VVUQ) for simulation analysis (see Appendix B for installation details). It allows us to automate parameter exploration analysis and explore essential one-at-a-time input uncertainty quantification. Importantly, uncertainty quantification and sensitivity analysis are required in multiscale migration application to understand in what regime and scenario our simulation approach performs well.

FabSim3, EasyVVUQ, QCG Pilot Job and other QCG components can be combined in a variety of ways, enabling users to combine their added values while retaining a limited deployment footprint. EasyVVUQ can use FabSim3 to facilitate automated execution. Users can convert their EasyVVUQ campaigns to FabSim3 ensembles using a one-liner (*campaign2ensemble*), and the FabSim3 output is ordered such that it can be directly moved to EasyVVUQ for further decoding and analysis.

VisualFlee

Visualising results using the network map can assist researchers in constructing simulations for forced population movements. To reveal potential mistakes, bugs in the network graph and to picture the distribution of forced population for simulation periods, we established VisualFlee. Its ultimate aim is to produce a tightly coupled application of the type that would allow modelling forced displacement in real time. Python code combines the CSV data, describing geographical locations and how their population changed through time, and produces data in the standard GeoJSON format. The visualisation itself is created in HTML and JavaScript using the popular Leafletjs library to display maps and the Leaflet.timeline plugin to animate them. We set out to visualise the movement of people during a conflict on a map, with circles which grow and shrink as the population in each location changes. The colours of the circles distinguish conflict zones, camps, towns and forwarding hubs as illustrated in Figure 6.6. We provide step-by-step VisualFlee construction instructions in Appendix B.

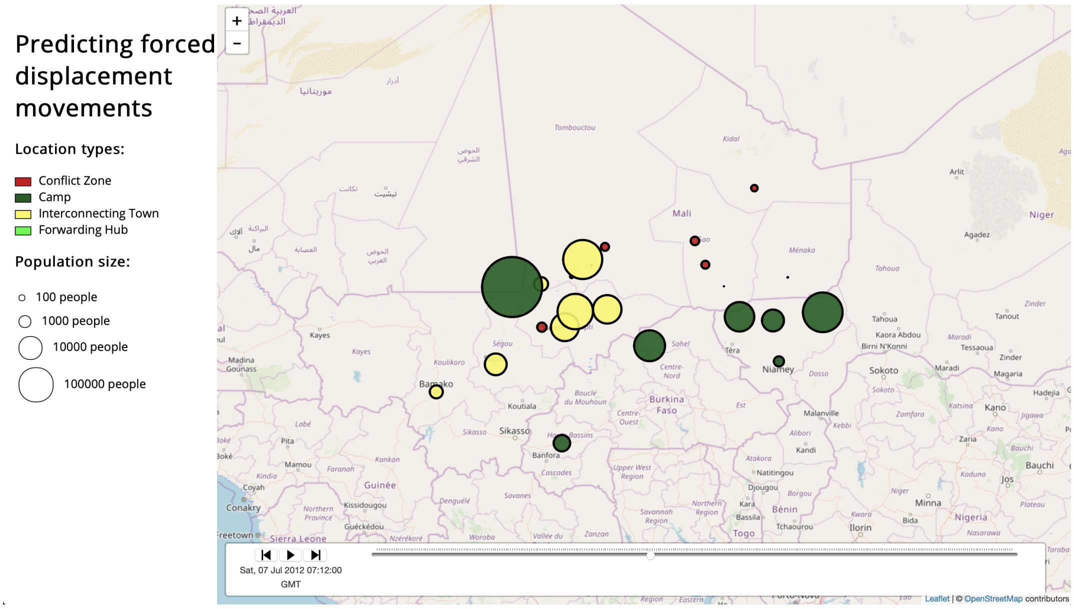

Figure 6.6: VisualFlee representation.

6.4 Conclusion

Through the use of computational modelling techniques and the automation tools, we are able to construct a highly transparent and customised programme and achieve automation of key tasks by simplifying and accelerating activities, including construction, execution and analysis of simulations. In this chapter, we propose an automated SDA using the FabFlee toolkit that provides a platform to construct multiple simulations, execute ensemble simulations, and explore agent-specific parameters and policy decision. In the next chapter, we systematically explore the possible impact of specific policy decisions, such as camp capacity changes, camp and border closures, and forced redirection.

Chapter 7. Modelling Policy Decisions

Based on:

Suleimenova, D. and Groen, D. (2020), “How Policy Decisions Affect Refugee Journeys in South Sudan: A Study Using Automated Ensemble Simulations”. *Journal of Artificial Societies and Social Simulation*, 23(1)2.

7.1 Introduction

We propose the use of automated agent-based simulations to predict forced population movements, to help governments and NGOs to conduct a better-informed allocation of humanitarian resources, and to help inform policy decisions. An automated simulation approach is essential to systematically investigate the effect of policy decisions through simulation, as many scenarios need to be analysed, and manual simulation construction is simply too labour-intensive. Our automated simulation tool uses FLEE to model, and we validate its accuracy using data from UNHCR on real conflict situation of South Sudan.

7.2 Application: The effects of policy decisions on forced population arrivals in South Sudan

To understand the significance and practicality of a generalised and automated SDA, we apply it to construct a new model of the South Sudan conflict, which involves almost 2 million forcibly displaced people fleeing to destination camps (UNHCR, 2018). For many years, Sudan experienced a civil war from which South Sudan declared independence on the 9th July 2011. However, the authorities of South Sudan failed to deliver the basic needs (Reid, 2018), and in December 2013, a conflict between the government and rivals broke out.

Specifically, the civil war in South Sudan started on the 15th December 2013, following fierce fighting between rival units of the Sudan Peoples’ Liberation Movement (SPLM) and the

Sudan People's Liberation Army (SPLA) in the capital, Juba (UNHCR, 2015b). Subsequently, South Sudan's president Salva Kiir announced that former vice president Riek Machar had attempted a coup. Machar escaped from Juba and became the leader of an armed opposition movement, namely the 'SPLM/A in Opposition'. Violence and fighting spread to other parts of the Jonglei, Upper Nile and Unity states, as well as other regions of South Sudan (ICG, 2014), which forced people to flee internally and across neighbouring countries.

Our South Sudan model has a simulation period of 604 days starting from the 15th December 2013 to the 10th August 2015, during which 2.4 million people forcibly escaped the country. We run the simulation for ten camps (listed in Table 7.1) in neighbouring countries, namely Ethiopia, Kenya, Sudan and Uganda. Overview of the geographical network model for South Sudan demonstrated in Figure 7.1.

Countries	**Camp names**
Ethiopia	Tierkidi, Kule, Pugnido and Jewi
Kenya	Kakuma
Sudan	Khartoum and West Kordofan
Uganda	Adjumani, Rhino and Kiryandongo

Table 7.1: List of camps in neighbouring countries of South Sudan

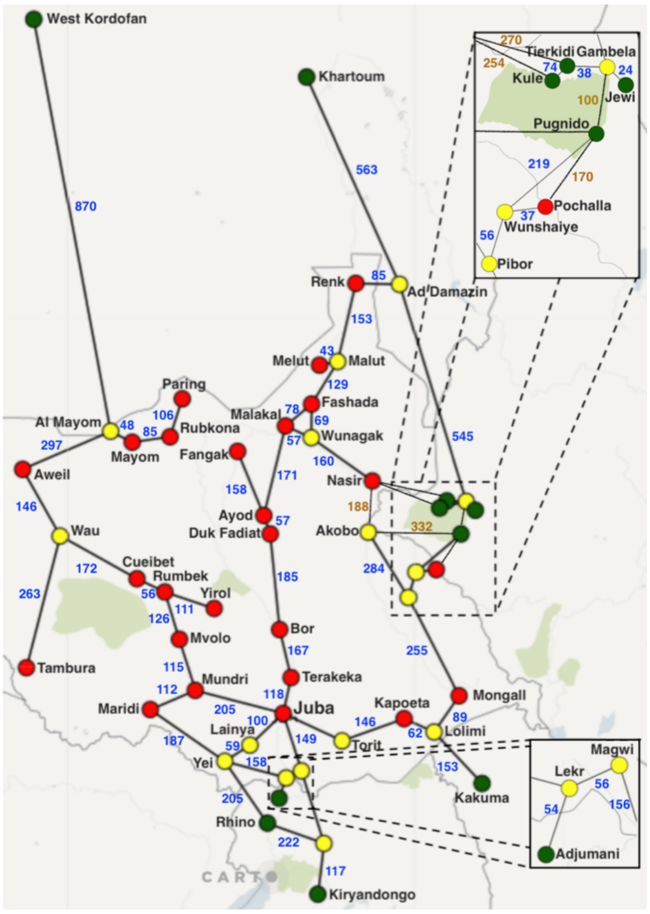

Figure 7.1: Overview of the geographic network model for South Sudan. This includes conflict zones (red circles), camps (dark green circles) and other major settlements (yellow circles). Interconnecting roads and walking routes are given with lines, with adjacent numbers used to indicate their length in kilometres (blue for roads and brown for walking routes). Background maps are courtesy of carto.com created using OpenStreetMap data.

After selecting our conflict country and the simulation period, we then extract data from the sources according to the SDA. Next, we construct our initial model for South Sudan with default settings using the discussed three CSV file formats, namely *locations.csv*, *routes.csv* and *closures.csv*. The initial constructed model, which is the third phase of SDA, is then refined with additional information obtained from reports (fourth phase of SDA). In Figure 7.2, we demonstrate the layout of our simulation tests for the South Sudan conflict. It includes refinements to determine how policy decisions, such as border closures, redirection between camps and changes in camp capacities, can affect the distribution of forced population counts and simulation results. Using our approach, we also automatically create and perform sensitivity analysis study for each of our models. Bearing in mind, we set our default setting to the forced population move speed, which is equal to 200 km per day and the awareness of surrounding is one link.

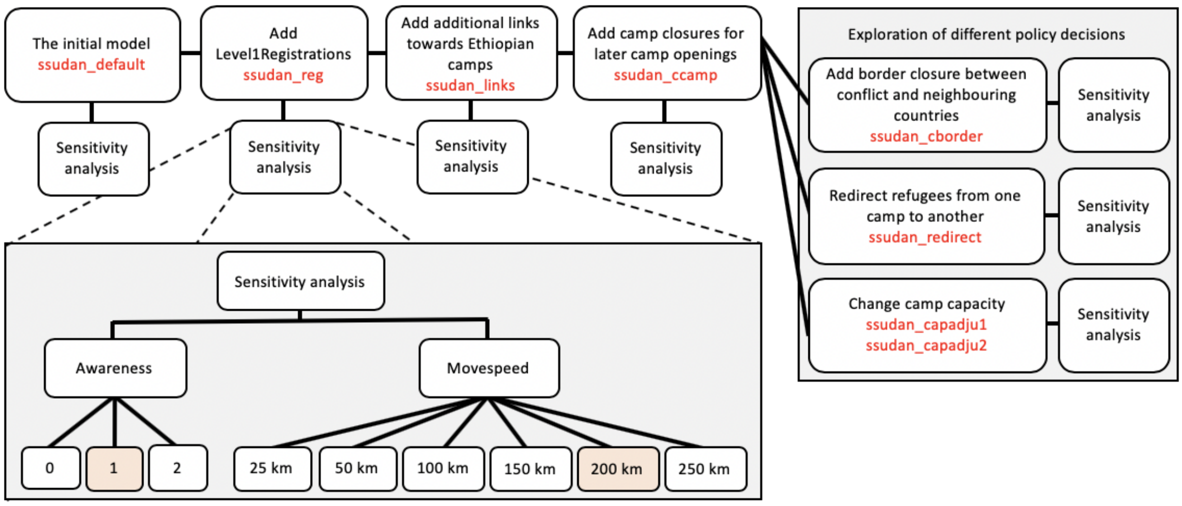

Figure 7.2: Setup of simulation execution for South Sudan. For each execution, we perform ensemble runs for sensitivity analysis. The structure of these ensembles is given in the bottom grey panel.

After constructing the initial South Sudan model (*ssudan_default*), we execute and obtain the initial results. Next, we determine level 1 registrations from the source data and include them to improve the initial model. We name the second model as *ssudan_reg* and execute to observe changes in the results. We further refine the *ssudan_reg* model using additional information obtained from publicly available online reports. Specifically, the UNHCR (2014a) report declares that displaced people arrived at Ethiopian camps on foot, due to the lack of roads. To accommodate this fact, we modify our simulation assumptions, and we incorporate specific 'off-road links' from conflict zones to Ethiopian camps in a modified simulation setup named *ssudan_links*. To reflect the fact that off-road routes are likely to result in slower travel speeds, we multiply the coordinate point-by-point distances by 2 for all walking routes. We also incorporate additional information in regards to later camp openings and closures, which is derived from the UNHCR reports (run *ssudan_ccamp*).

In Figure 7.3, we demonstrate the averaged relative difference for four simulations (*ssudan_default, ssudan_reg, ssudan_links* and *ssudan_ccamp*). Despite the same levels before day 200, the average relative difference for these runs persistently lessens respectively from 0.615 to 0.499 over the simulation period and the refinement of the South Sudan model as we incorporate additional details. Overall, *ssudan_ccamp* is the most refined with the lowest average relative difference in the aggregate level.

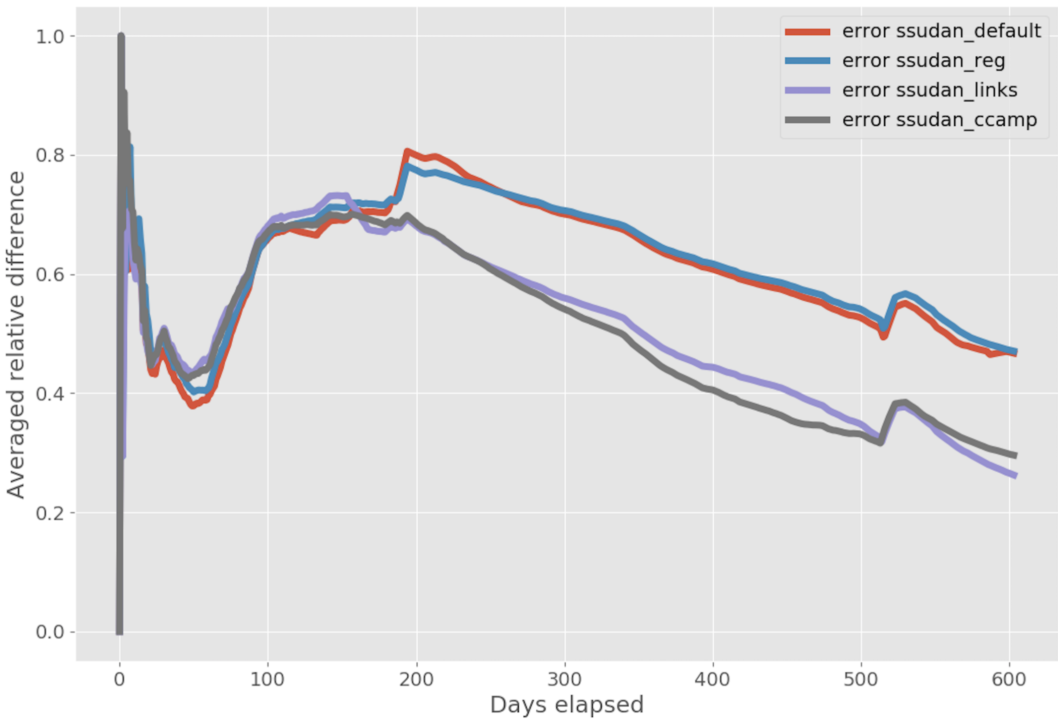

Figure 7.3: Overview of the averaged relative differences for ssudan_default (red line), ssudan_reg (blue line), ssudan_links (violet line) and ssudan_ccamp (grey line) simulations.

Moreover, we perform a range of sensitivity analysis tests. First, we execute ten replicas of *ssudan_ccamp* with default settings to determine the range of the output due to the probabilistic nature of the simulations. Over these ten executions, the average relative difference ranged between 0.495 and 0.502. Second, we perform a sensitivity analysis for each run by varying the level of agent awareness range and a speed limit of forced population. Here, the awareness range represents the level of knowledge of migrants about nearby locations. They may know only the distance to the adjacent locations in the graph (path distance only), or the type of location for adjacent locations (1 link away), or the location type of locations adjacent to those (2 links away). We present the results of this analysis in Table 7.2. For the most refined models, the averaged relative difference is the lowest when agents are aware of locations 1 link away, though the difference is marginal compared to simulations with an awareness range of 2 links away. Our simulations are clearly sensitive to the maximum move speed parameter, and in particular move speeds below 100km/day result in significantly higher validation errors. This parameter sensitivity is in line with our simulations of previous conflicts discussed in Chapter 5.

Run type	**ssudan_default** (least refined)	**ssudan_reg** ...	**ssudan_links** ...	**ssudan_ccamp** (most refined)
normal (default)				
1 link away, 200km/day	0.615	0.621	0.509	0.499
awareness range				
Path distance only	0.627	0.630	0.530	0.522
1 link away	0.613	0.621	0.510	0.500
2 link away	0.611	0.614	0.517	0.507
max. move speed (km/day)				
25	0.667	0.673	0.575	0.570
50	0.634	0.643	0.535	0.527
100	0.621	0.629	0.519	0.503
150	0.616	0.625	0.514	0.501
200	0.611	0.622	0.511	0.502
250	0.616	0.624	0.509	0.502

Table 7.2: Averaged relative difference values, averaged over time and all four base type of simulations using different agent awareness ranges, and different speed limits for agents. Note that we present results from 3 separate executions of the default type run: in the first data row, the third data row (labelled ‘1 link away’) and the ninth data row (labelled ‘200’).

We present *ssudan_ccamp* simulation results for all ten camps validated against the UNHCR forced population registration data in Figure 7.4. The most populous camp in our simulation is Adjumani with more than 140,000 forcibly displaced people over the simulation period and slightly overpredicted after 200 days into simulation compared to the data. The reason being that it is the closest camp for forced population fleeing from the South Sudan conflict. The forecast forced population counts in Kiryandongo and Kakume (at the start prior to 100 days) are in close agreement with the UNHCR data, while our simulations underpredict for Kule, Jewi and Khartoum camps. South Sudan has a record of conflicting prior to our simulation start date. Kakuma (45,239), Pugnido (42,044), Rhino (5,313) and Kiryanongo (15) camps had registered number of forced displacement fled prior to the simulation start; these, therefore, do not count towards the forced population arrival numbers.

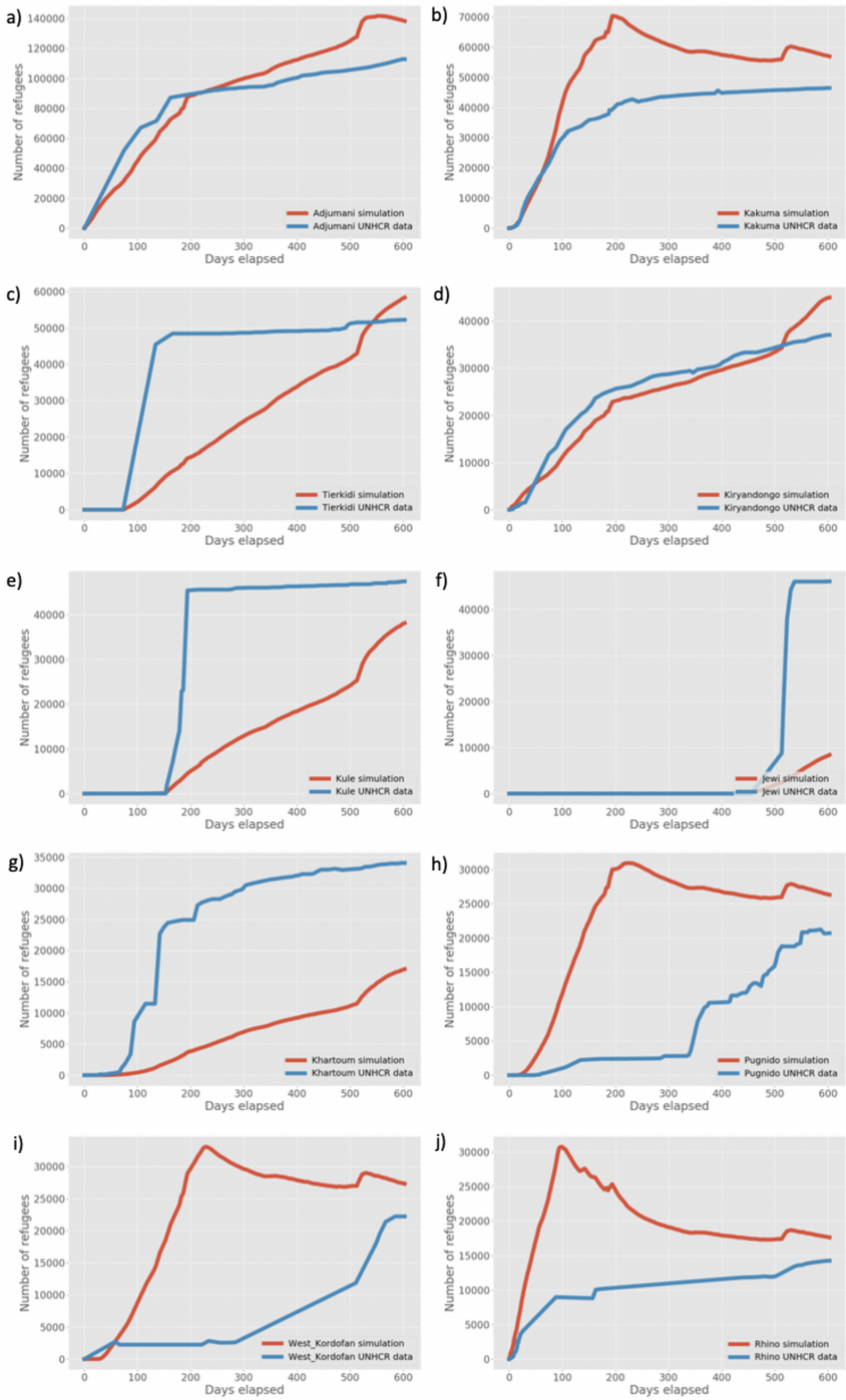

Figure 7.4: Number of forced migrants as forecast by our ssudan_ccamp simulation and validated against the UNHCR data for the South Sudan conflict. (**a-j**) Graphs are ordered by camp population size, with the most populous camp on the top to the smallest one on the bottom.

There is no arriving forced population at the start of simulation period for several camps, namely Tierkidi, Kule and Jewi, illustrated in Table 7.3, as they opened after the conflict has commenced according to the UNHCR data. For instance, the Tierkidi camp has no arrivals prior to 73 days of simulation, but the forced population counts to increase over the simulation period and overpredict UNHCR data by the end of the simulation period. In addition, the Jewi, Kule and Khartoum camps show slowly increasing and underpredicted forced population counts. Whereas, the Pugnido, West Kordofan and Rhino camps are considerably overpredicted according to simulation results by almost 25,000 displaced people decreasing to 5,000 people per each camp by the end of the simulation period.

Countries	**Camp names**	**Camp opened on**
Ethiopia	Tierkidi	26th of February 2014
	Kule	17th of May 2014
	Jewi	15th of March 2015

Table 7.3: A list of camps that opened after the South Sudan conflict has commenced in neighbouring countries.

7.2.1 Policy decisions by camp capacity changes

To explore how changes in camp capacities affect simulation results, we change the capacity of the most populated camp, namely Adjumani. For the first instance, we decreased the original capacities of 112,734 forced population by half. The second instance involves an increase in the original capacity by 50%. In Figure 7.5, we present the number of forced population for Adjumani camp *ssudan_adjumani1* (capacity: 56,367 forced migrants) and *ssudan_adjumani2* (capacity: 169,101 forced migrants). We find that a reduction of capacity in Adjumani results in up to 16% fewer forced population arrivals in camps, which implies considerably longer displacement travel times. However, increasing the capacity at Adjumani by allocating more resources appears to only result in a very limited increase in forced population arrivals ($<$ 4%). Based purely on these results, we may observe that in a setting where aid resources are heavily constrained, the default capacity of this camp is close to optimal.

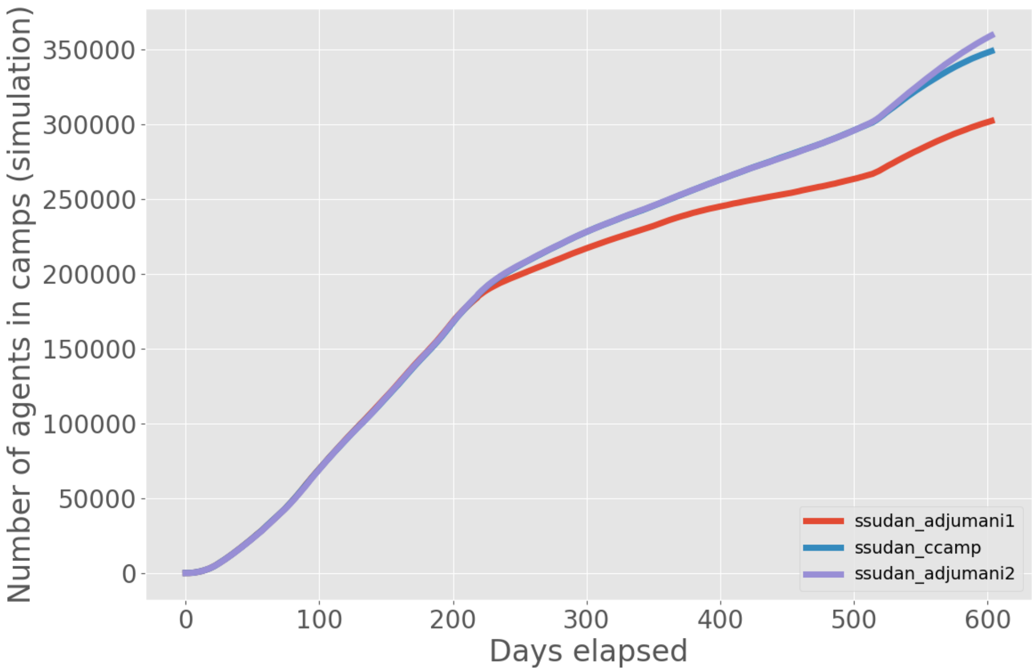

Figure 7.5: Comparison of number of forced migrants in camps between three simulations with capacity change for Adjumani camp in comparison to the base model of ssudan_ccamp.

7.2.2 Policy decisions by camp and border closures

To investigate the effects of policy decisions, such as camp and border closures, we compare the forced population arrivals in camps between a model without camp closures (*ssudan_links*), a model with camp closures incorporated in *ssudan_ccamp*, and a model which contains an additional border closure between South Sudan and Uganda, enforced until day 302, which is halfway into the simulation (*ssudan_cborder*). We present our comparison results in Figure 7.6, and find no significant differences between *ssudan_links* and *ssudan_ccamp*. However, we do find differences between the first two models and *ssudan_cborder*, which results in 40% fewer forced population arrivals on day 302. It implies an increasingly long travel time for displaced people up to day 302, the day that the border is again reopened. In addition, the delaying effect of border closures lingers in our simulation results after borders have been reopened, with approximately 15% fewer arrivals on day 400. This emergent behaviour, by definition, cannot be validated against reality. However, explanations for such delays are possible. For instance, the forced population may fear that recently opened borders are more likely to be closed again, or may not be immediately aware that a previously closed border has again reopened.

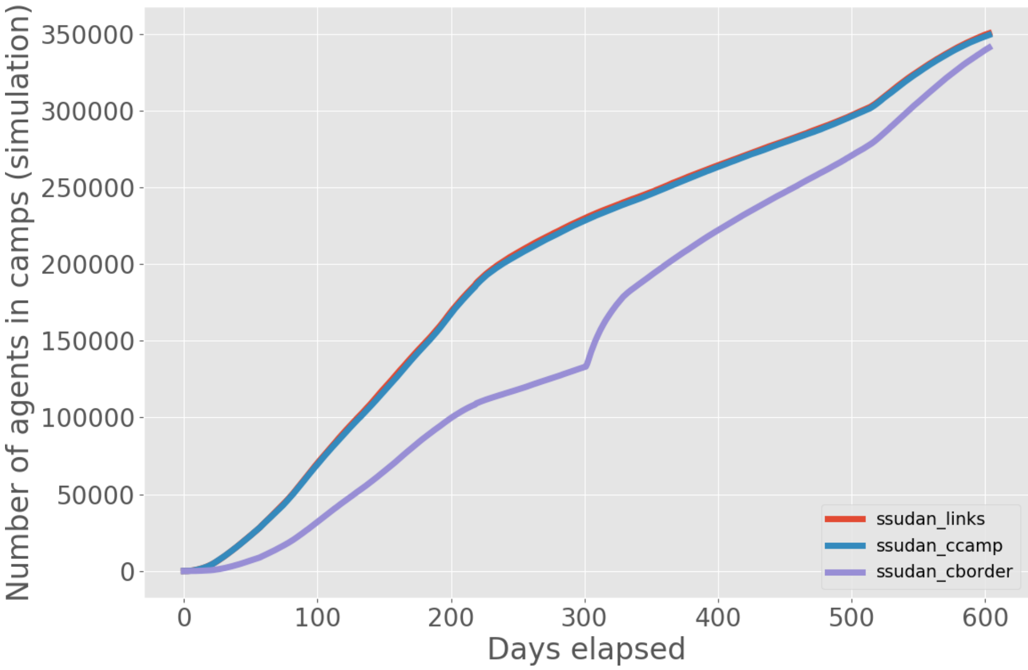

Figure 7.6: Comparison of the total number of forced population in camps between ssudan_links (red line), ssudan_ccamp (blue line) and ssudan_cborder (violet line) simulations.

7.2.3 Policy decisions by forced redirection

We explore how the enforced redirection of arriving forced displacement from one camp to another can affect the distribution of forced population across all the camps. As an exemplary, we created a model (*ssudan_redirect*) where all displaced people arriving in Kule, Jewi and Pugnido are redirected to the Tierkidi camp, which has its capacity increased accordingly, creating a counterfactual situation where Tierkidi is the single central camp in Ethiopia receiving forced population from South Sudan. This kind of centralised management of incoming forced population has been known to occur in some other conflict situations, such as Mauritania (Mbera camp) in the North Mali conflict in 2012.

We present a comparison of arrivals across seven camps in both models in Figure 7.7. Here, Kule, Jewi and Pugnido are excluded from the comparison, as they do not host any displaced people in the modified simulation. In comparison to the *ssudan_ccamp* simulation results for individual camps, we attain different distribution of forced population across camps in *ssudan_redirect*. By Day 180, Tierkidi has received twice as many arrivals in *ssudan_redirect* than in *ssudan_ccamp*, while the other six camps retain similar arrival rates. However, after Day 180 the number of forcibly displaced people in the other six camps becomes lower in *ssudan_redirect* than in *ssudan_ccamp* while the number of forced population counts in Tierkidi remains considerably higher. This behaviour can primarily be attributed to the Pugnido camp, which reaches full capacity around Day 180 in *ssudan_ccamp* (see Figure 7.4), but which is redirected to Tierkidi in *ssudan_redirect*, a camp with a higher (combined) capacity.

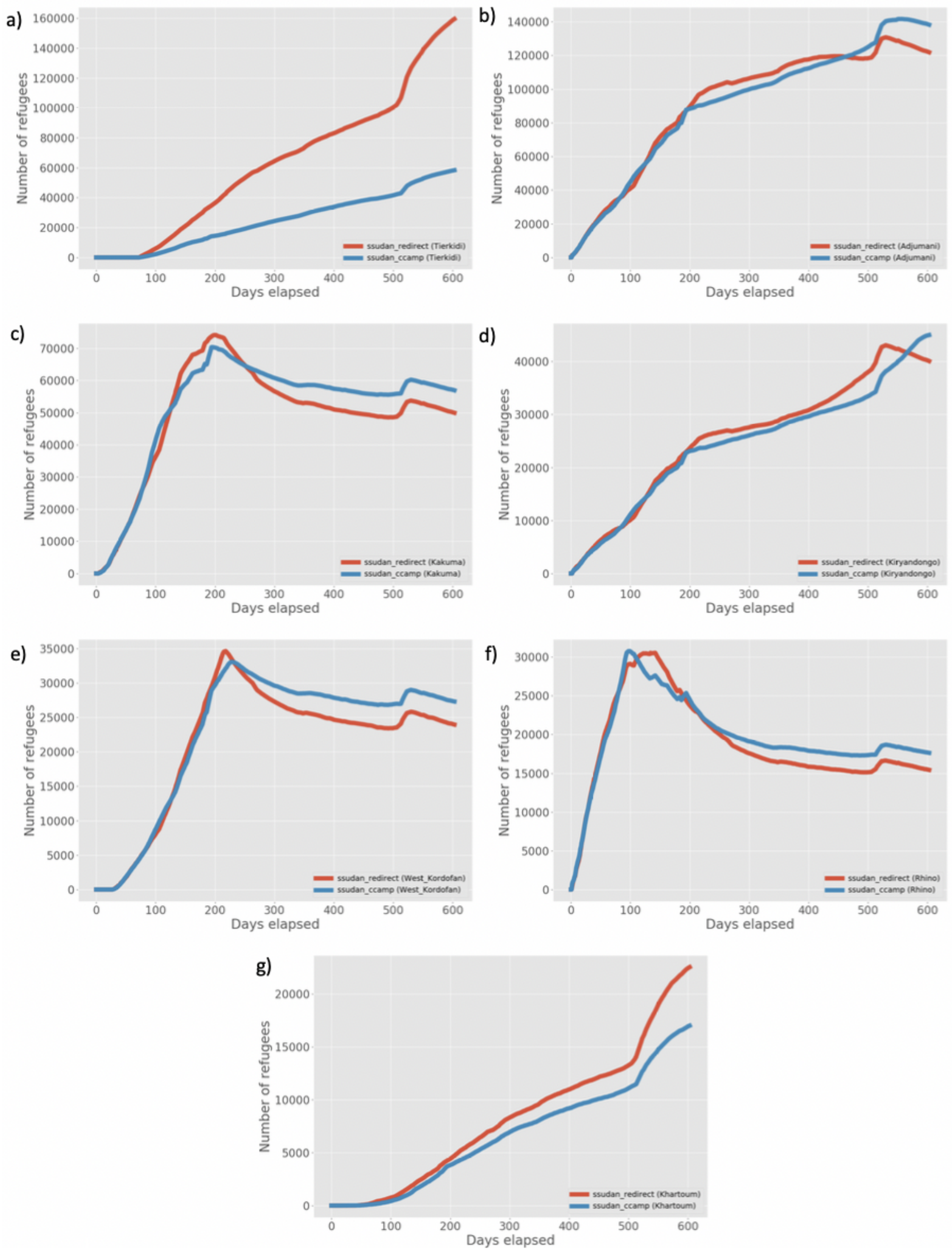

Figure 7.7: Comparison of the number of forced population in seven camps as forecast by our *ssudan_ccamp* and *ssudan_redirect* simulations for the South Sudan conflict. (**a-g**) Graphs are ordered by camp population size, with the most populous camp on the top to the smallest one on the bottom.

7.3 Conclusion

Forecasting forced displacement is both very important and very challenging. We predict the distribution of forced population arrivals to potential destinations, as governments and NGOs can efficiently allocate humanitarian resources and provide protection to vulnerable people. Through the use of computational simulations and the automation approach presented in Chapter 6, we are able to systematically explore the possible impact of specific policy decisions, while accounting for the sensitivity to at least some of individual parameters and assumptions in the model. To achieve this, we use an automated simulation development approach, and use it to forecast forced population arrivals in camps in the South Sudan crisis. Our approach relies on the FabSim3-based FabFlee toolkit.

We apply our automated ensemble simulation approach to analyse the effect of policy decisions on forced population journeys in the South Sudan conflict. This conflict is relatively difficult to simulate, primarily due to the lack of roads and difficult food circumstances. We update the model, include walking routes from east South Sudan to camps in the Gambela region of Ethiopia, and achieve a much lower validation error (averaged relative difference) as a result. All policy decisions presented here are purely hypothetical, and primarily derived from having observed similar decisions being made in the three conflicts we analysed in the validation study chapter.

In terms of policy decision examples, we incorporated two camp capacity changes, a border closure, and a forced redirection. We find that a reduction in camp capacity induces up to 16% fewer forced population arrivals while an increase in camp capacity results in a limited increase in forced population arrivals ($< 4\%$) at the destination camps. In addition, border closure results in 40% fewer force population arrivals and an increasingly long travel journey to other camps. There is also a lingering effect in prolonged force population journey times once a border is again reopened and a clear boost in forced population arrivals when forced population are redirected to a reduced number of camps with larger capacities. We believe these policy decisions in particular warrant more in-depth investigation, using simulation and data analysis approaches that take into more relevant factors and circumstances, and can also leverage the benefits from the automation approach we presented in the previous section.

Chapter 8. Conclusions

8.1 Research summary

In this thesis, we predict the distribution of people forced to flee because of war, armed conflict and/or political instability among destination camps. We simulate forced population movements using a generalised and partially automated agent-based simulation development approach.

In Chapter 2, we reviewed existing migration theories, forced displacement models and prediction techniques, such as econometric and early warning models. Importantly, we identified that there is a lack of prediction models for forced displacement and hence explored the use of an agent-based model (ABM), which is a computational modelling technique widely adopted in recent population and migration studies. ABM has the potential to contribute to a better understanding of population displacement patterns.

In Chapter 3, we examined existing ABM development processes and determined the demand for a generic simulation development approach (SDA). Subsequently, we proposed our research methodology, which is a generalised SDA using ABM for situation-specific scenarios involving two development processes, namely the generic model and simulation development for validation. The former has one-time construction phases, while the latter applies to individual situation-specific scenarios following concurrent phases. We also present the development approach for forecasting simulations with an additional step of forecasting scope and metrics for model refinement. We stress that our generalised SDA for validation and forecasting is applicable to other simulation models.

In Chapter 4, we presented a generalised ABM simulation approach for forced population displacement and described each phase for constructing and executing forced displacement simulations. We discussed how a generalised SDA uses input data from publicly available data sources, such as the United Nations High Commissioner for Refugees (UNHCR) database, the

Armed Conflict Location and Event Data Project (ACLED), and the Bing Maps platform, to construct an initial model for simulation execution. We also presented how we conduct analysis and validate our simulation results against the UNHCR data using the Mean Absolute Scaled Error (MASE), among others.

In Chapter 5, we applied our generalised SDA to model three African forced displacement conflict scenarios, namely Burundi, Central African Republic and Northern Mali. These conflict situations have forced the population fleeing from their origin country due to armed conflict situations. We identified conflict zones, camps and intermediate towns for three African countries, constructed and refined the initial models and executed these simulations using our FLEE code. We run our simulation model, predicted the distribution of incoming forced displacement across destination camps and reproduced at least 75% of the forced population movement destinations in all three conflicts.

In Chapter 6, we explored automation tools and techniques to construct rapid, consistent and efficient ABM simulations as manual simulation construction and execution is simply too labour-intensive. We used the FabSim3 automation toolkit, automated our generalised SDA by simplifying development phases and introduced quick and more systematic activities for developing forced displacement simulations.

In Chapter 7, we applied our generalised and automated SDA to analyse the effects of policy decisions, such as camp capacity changes, camp and border closures, and forced redirection, on forced population journeys in the South Sudan conflict. Through the use of computational modelling and the automation approach presented in Chapter 6, we aimed to assist governments and NGOs to allocate humanitarian resources efficiently and provide protection to vulnerable people.

8.2 Research contributions

The work presented here makes several contributions to the research area of forced population displacement:

- We have proposed a new forced displacement simulation technique which allows for the 'virtual implementation' of policy decisions, such as camp capacity changes, camp and border closures, and forced redirection. This may allow governments and NGOs to study counterfactual outcomes and conduct a better-informed allocation of humanitarian resources. We apply our generalised and automated simulation toolkit to four African countries. We are the first to attempt such predictions across multiple major conflicts using a single simulation approach and to investigate the effects of policy decisions on forced displacement movements.

- We have developed an agent-based simulation development approach targeting situation-specific scenarios, such as the prediction of the forced population movements across destination camps. The scope of our SDA is unique compared to other abstractions and can be applied to other situation-specific scenarios, such as human and animal movements. To the best of our knowledge, we are the first to test such an abstraction in several validations and forecasting settings.

- Another contribution is the development of FLEE simulation code and the FabFlee automation toolkit that are now in use by two large EU research projects, and have a significance in predicting forced population movements. Notably, FLEE and FabFlee are central for the development of a software toolkit to support VVUQ for multiscale applications in the Verified Exascale Computing for Multiscale Applications (VECMA, €3 million) and one of three pilot applications in the HPC and Big Data Technologies for Global Systems (HiDALGO, €8 million) projects.

- There is also wider applicability of forced displacement modelling technique to internally displaced persons and child migration. The simulation model of FLEE and FabFlee can be extended to other migration scenarios though we would expect such an extension to be challenging. It may also bring large interest from outside as it is a current matter known by all.

- Overall, through this research, we established links with universities, external organisations, such as UNHCR, the Search and Rescue Observatory for the Mediterranean (SAROBMED) and Medecins Sans Frontieres (MSF), and other NGOs. This collaborative work has the potential to add value to our research, as well as our contributions to academia and other authorities. We also provided valuable contributions to funding proposals, two of which were funded (VECMA and HiDALGO), and two under review (PANACEA and MIPS).

8.3 Research limitations

In this section, we acknowledge our research limitations in predicting forced population displacement movements. First, predicting forced displacement and forced population movements itself are under-explored areas, which could lead to a lack of focus and relevant knowledge concerning forced population, and their drivers for movement. There is also limited research on forecasting forced displacement and techniques to forecast future events and patterns. However, predicting forced displacement is an entry point for exploring forced population counts and their displacement patterns. There is also a need for conflict evolution models to do actual forecasts for which we took the first steps to address this (Groen et al., 2019).

Second, it is true that most of the data on refugees or IDPs are incomplete or inconsistent, and interpolated in the simulation analysis. It has certainly improved from its initial availability since decades ago, but still, not all information is accessible. Moreover, obtaining and extracting the available data is labour-intensive and inefficient. Hence, there is a demand for an automated data collection for forced population displacement. Our research also has a dependency on non-static data sources as forced population patterns are not monitored thoroughly.

Third, our generic model omits a range of factors which are considered important according to the empirical literature, but for which we could not find accurate and tractable means to convert empirical conclusions to simulation parameters. In some cases, such as GDP and the presence of existing conflicts, the significance of these factors has been confirmed on a country-by-country level, but not on a city-by-city level. In other cases, such as religion and ethnicity, we simply did not find reliable statistical information on a local level for these conflicts. Despite, it is important to note that these factors may be crucial to the outcome of simulation results.

Finally, since each forced displacement crisis is unique, the impact of some information used to model the conflict can be different across different scenarios. Previously observed factors may have little or no role in future crises. Such models are based on assumptions that may also be specific to a particular context. For instance, in our simulations, we assume that forced population travel no more than 200 km/day. However, this assumption may differ depending on forced migrants state of health, means of transport and travel circumstances. Hence, it is key to consider the right assumptions and appropriately incorporate them within forced displacement simulations.

8.4 Areas of future research

Predicting forced displacement is an alluring research area today. There are many countries in Africa and Asia facing various economic and political instabilities, which force people to flee. In addition, more and more data is publicly available and computational advancements are alongside. As a result, these factors are attracting the attention of researchers and organisations to explore and examine forced displacement with new perspectives.

There are many possibilities on how this research can evolve. First, our simulation approach for forced displacement can also be applied to predict IDPs fleeing conflict situations, who might have much more complex movement patterns and difficult to validate. Second, this research can improve by considering economic impacts in simulation models, such as conflict evolution, the recovery process of conflict regions and operation of camps on smaller scales. Third, in Chapter 3, we proposed a generalised SDA for forecasting forced population displacements, which we have not used due to research scope but it can be applied to simulate newly erupted conflict situations in future research.

Moreover, our proposed generalised SDA is partially automated, and thus, there is room for improvement by building an end-to-end automated solution. It needs to eliminate the remaining manual processes, such as data collection, and consider other human interactions with the tool that have basic programming skills. An end-to-end automated simulation toolkit can be seen as a solution that is easily used, analysed and scaled using supercomputers.

We are also a member of the SAROBMED consortium, which aims to collect, analyse and disseminate reliable data on forced displacement and violations in the Mediterranean. We work closely with researchers and NGOs, and it allows us to contribute to data aspects of forced displacement and forecast forced population movements at sea, which is an on-going activity.

Furthermore, the research of forced displacement using agent-based simulation is a stepping stone to two projects (VECMA and HiDALGO) funded by the European Union Horizon 2020 research and innovation programme. Both of these EU projects expand the research scope of forced population displacement and provide an opportunity to investigate forced displacement from various perspectives, which are illustrated in Figure 8.1.

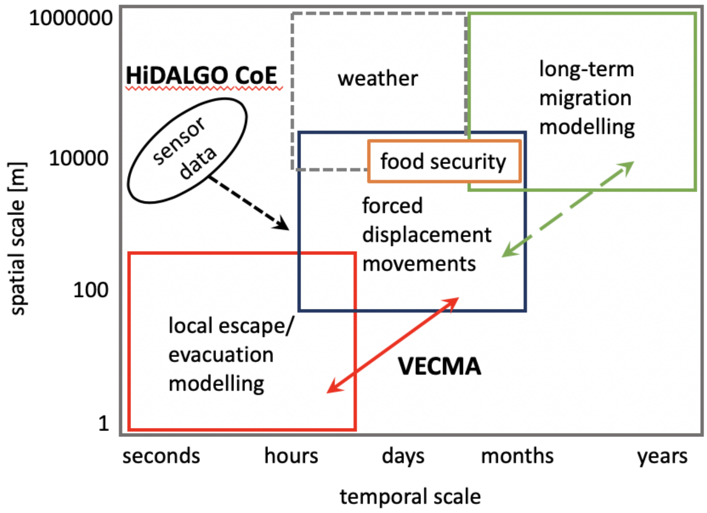

Figure 8.1: Ways forward: Forced population displacement modelling.

The VECMA project aims to develop a software toolkit to support VVUQ for multiscale applications, one of which is forced population modelling. High-quality VVUQ on simulations will help NGOs and governments to prepare for forced population arrivals more effectively and make predictions in regions where existing data is incomplete. The awareness of counterfactual outcomes can then be communicated to the general public and research communities, as well as to humanitarian agencies. While in HiDALGO, forced population displacement modelling with the FLEE code and the FabFlee toolkit are central to one of three pilot applications including weather and social media. We aim to couple our approach to weather and environmental simulations, particularly in situations where the weather is known to play a determinant role in the choice of destination for forced population (e.g., South Sudan). We also aim to couple our models to microscale simulations of cities in the country of conflict, which are currently under development, as well as large-scale migration simulations, connecting different simulations of forced population movements. However, there is a risk on the availability and quality of data.

Bibliography

Abar, S., Theodoropoulos, G. K., Lemarinier, P. and O'Hare, G. M. (2017). Agent Based Modelling and Simulation tools: A review of the state-of-art software, *Computer Science Review* **24**: 13–33.

Abel, G. J., Brottrager, M., Cuaresma, J. C. and Muttarak, R. (2019). Climate, conflict and forced migration, *Global Environmental Change* **54**: 239–249.

Alhanaee, A. and Csala, D. (2015). Motives to flee: Modeling Syrian refugee crisis, *Technical report*, Masdar Institute of Science and Technology.

Allan, R. (2010). Survey of agent-based modelling and simulation tools, Science and Technology Facilities Council. Available at: http://purl.org/net/epubs/manifestation/5601/DLTR-2010-007.pdf.

Amblard, F., Geller, A., Neumann, M., Srbljinovic, A. and Wijermans, N. (2010). Analyzing social conflict via computational social simulation: A review of approaches, *in* K. Martinás, D. Matika and A. Srbljinovic (eds), *Complex Societal Dynamics - Security Challenges and Opportunities*, IOS Press, Amsterdam, pp. 126–141.

Anderson, J., Chaturvedi, A. and Cibulskis, M. (2007). Simulation tools for developing policies for complex systems: Modeling the health and safety of refugee communities., *Health Care Management Science* **10**(4): 331–339.

Anderson, J., Chaturvedi, A., Lengacher, D. and Cibulskis, M. (2006). Modeling the health of refugee camps: An agent-based computational approach, *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*.

Arora, P., Malhotra, P., Lakhmani, I. and Srivastava, J. (2017). Agent based software development, *International Journal of Scientific Research Engineering and Technology* **6**(4).

Balci, O. (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study, *Annals of Operations Research* **53**: 121–173.

Bijak, J. (2006). Forecasting international migration: Selected theories, models, and methods, *Technical report*, Central European Forum for Migration Research, Warsaw, Poland.

Bijak, J. and Wisniowski, A. (2011). Statistical modelling of international migration flows, *Technical report*, Southampton Statistical Sciences Research Institute, Southampton, UK.

Birkmann, J., Chang Seng, D. and Setiadi, N. (2013). Enhancing early warning in the light of migration and environmental shocks, *Environmental Science and Policy* **27S**: S76–S88.

Birta, L. G. and Arbez, G. (2013). *Modelling and simulation: Exploring dynamic system behaviour simulation foundations, methods and applications*, 2 edn, Springer, London.

Boano, C., Rottlaender, A., Alberto, S. and Viliani, F. (2003). Bridging the gap: Involuntary population movement and reconstruction strategy, *Technical report*, Centro Regionale d'Intervento per la Cooperazione, Reggio Calabria, Italy.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems, *Proceedings of National Academy of Sciences* **99**(3): 7280–7287.

Borshchev, A. and Filippov, A. (2004). From system dynamics to agent-based modeling, *The 22nd International Conference of the System Dynamics Society*, p. 25–29.

Boswell, C. (2002). Addressing the causes of migratory and refugee movements: The role of the European Union, *Technical Report 73*, United Nations High Commissioner for Refugees.

Boyce, M. and Vigaud-Walsh, F. (2015). "You are either with us or against us": Persecution and displacement in Burundi, *Technical report*, Refugees International.

Brück, T., Dunker, K. M., Ferguson, N. T., Meysonnat, A. and Nillesen, E. (2018). Determinants and dynamics of forced migration to Europe: evidence from a 3-D model of flows and stocks, IZA Discussion Papers 11834, Institute for the Study of Labor.

Bunoiu, M.-D. and Udroiu, I. (2016). Spotting trouble in migration flows: An indicator-based early warning model, *in* N. Iancu and I. Chiru (eds), *Proceedings of the XXIst International Conference. Intelligence in the Knowledge Society*, Bucharest, pp. 95–104.

Castle, C. and Crooks, A. (2006). Principles and concepts of agent-based modelling for developing geospatial simulations, *Technical report*, Centre for Advanced Spatial Analysis (UCL), UCL (University College London), Centre for Advanced Spatial Analysis (UCL).

Cirillo, P. and Gallegati, M. (2012). The Empirical Validation of an Agent-based Model, *Eastern Economic Journal* **38**(4): 525–547.

Collins, A. J. and Frydenlund, E. (2016). Agent-based modeling and strategic group formation: A refugee case study, *Proceedings of the 2016 Winter Simulation Conference* pp. 1289–1300.

Crooks, A., Castle, C. and Batty, M. (2008). Key challenges in agent-based modelling for geo-spatial simulation, *Computers, Environment and Urban Systems* **32**(6): 417–430.

Curcin, V. and Ghanem, M. (2008). Scientific workflow systems - can one size fit all?, *Biomedical Engineering Conference, CIBEC*, IEEE, Ciaro International, pp. 1–9.

Davenport, C., Moore, W. and Poe, S. (2003). Sometimes you just have to leave: Domestic threats and forced migration, 1964-1989, *International Interactions* **29**(1): 27–55.

Davidsson, P., Holmgren, J., Kyhlbäck, H., Mengistu, D. and Persson, M. (2007). Applications of agent based simulation, *in* L. Antunes and K. Takadama (eds), *Multi-Agent Based Simulation VII. MABS 2006*, Vol. Lecture Notes in Computer Science, 4442, Springer, Berlin, Heidelberg, pp. 15–27.

De Jong, G. F. and Fawcett, J. T. (1981). Motivations for migration: An assessment and a value-expectancy research model, *Migration Decision Making* pp. 13–58.

deMenocal, P. B. and Stringer, C. (2016). Human migration: Climate and the peopling of the world, *Nature* **538**(7623): 49–50.

Disney, G., Wiśniowski, A., Forster, J. J., Smith, P. W. F. and Bijak, J. (2015). Evaluation of existing migration forecasting methods and models, *Technical Report October*, Centre for Population Change, Southampton, UK.

Echevarria, J. and Gardeazabal, J. (2016). Refugee gravitation, *Public Choice* **169**: 269–292.

Edwards, A. (2016). UNHCR viewpoint: ‘Refugee’ or ‘migrant’ – Which is right?, Available at: https://www.unhcr.org/news/latest/2016/7/55df0e556/unhcr-viewpoint-refugee-migrant-right.html.

Edwards, S. (2008). Computational tools in predicting and assessing forced migration, *Journal of Refugee Studies* **21**(3): 347–359.

Elsenbroich, C. (2012). Explanation in agent-based modelling: Functions, causality or mechanisms?, *Journal of Artificial Societies and Social Simulation* **15**(3): 1.

Entwisle, B., Williams, N. E., Verdery, A. M., Rindfuss, R. R., Walsh, S. J., Malanson, G. P., Mucha, P. J., Frizzelle, B. G., McDaniel, P. M., Yao, X., Heumann, B. W., Prasartkul, P., Sawangdee, Y. and Jampaklay, A. (2016). Climate shocks and migration: An agent-based modeling approach, *Population and Environment* **38**(1): 47–71.

Epstein, J. M. (2008). Why model?, *Journal of Artificial Societies and Social Simulation* **11**(4).

Epstein, J. M. and Axtell, R. (1996). *Growing artificial societies: Social science from the Bottom up*, MIT Press, Cambridge, USA.

Estrada, L. E. P., Groen, D. and Ramirez-Marquez, J. E. (2017). A serious video game to support decision making on refugee aid deployment policy, *Procedia Computer Science* **108**: 205–214.

Faist, T. (2000). *The volume and dynamics of international migration and transnational social spaces*, Oxford University Press, Oxford.

Fleagle, J., Shea, J., Baden, A., Leakey, R. and Grine, F. (2010). *Out of Africa I: The first Hominin colonization of Eurasia*, Springer, Heidelberg.

Gebbie-Rayet, J., Shannon, G., Loeffler, H. H. and Laughton, C. A. (2016). Longbow: A lightweight remote job submission tool, *Journal of Open Research Software* **4**(1): 1–8.

Gilbert, N. (2008). *Agent-based models*, Quantitative Applications in the Social Sciences, Sage Publications.

Gleditsch, K., Salehyan, I. and Schultz, K. (2008). Fighting at home, fighting abroad: How civil wars lead to international disputes, *Journal of Conflict Resolution* **52**(4): 479–506.

Groen, D. (2016). Simulating refugee movements: Where would you go?, *Procedia Computer Science* **80**: 2251–2255.

Groen, D., , Bell, D., Arabnejad, H., Suleimenova, D., Taylor, S. E. J. and Anagnostou, A. (2019). Towards modelling the effect of evolving violence on forced migration, *in* N. Mustafee, K. H. G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas and S. Y. J. (eds), *Proceedings of the 2019 Winter Simulation Conference (WSC19)*, IEEE, National Harbor, Maryland.

Groen, D., Bhati, A. P., Suter, J., Hetherington, J., Zasada, S. J. and Coveney, P. V. (2016). FabSim: Facilitating computational research through automation on large-scale and distributed e-infrastructures, *Computer Physics Communications* **207**: 375–385.

Groen, D., Hetherington, J., Carver, H. B., Nash, R. W., Bernabeu, M. O. and Coveney, P. V. (2013). Analysing and modelling the performance of the HemeLB lattice-Boltzmann simulation environment, *Journal of Computational Science* **4**(5): 412–422.

Hagen-Zanker, J. (2008). Why do people migrate? A review of the theoretical literature, *Technical report*, Maastricht Graduate School of Governance.

Hassani-Mahmooei, B. and Parris, B. W. (2012). Climate change and internal migration patterns in Bangladesh: An agent-based model, *Environment and Development Economics* **17**(06): 763–780.

Hattle, A., Yang, K. S. and Zeng, S. (2016). Modeling the Syrian refugee crisis with agents and systems, *The UMAP Journal* **37**(2): 195–213.

Heath, B., Hill, R. and Ciarallo, F. (2009). A survey of agent-based modeling practices (January 1998 to July 2008), *Journal of Artificial Societies and Social Simulation* **12(4)**(9).

Hilderink, H., van der Gaag, N., van Wissen, L., Jennissen, R., Roman, A., Salt, J., Clarke, J. and Pinkerton, C. (2002). Analysis and forecasting of international migration by major groups part iii, *Technical report*, Office for Official Publications of the European Communities, Luxembourg.

Huang, V. and Unwin, J. (2019). Markov chain models of refugee migration data, Available at: https://arxiv.org/abs/1903.08255.

Huzdik, K. (2014). *Migration potential and affecting factors in Hungary in the first decade of the 21st century*, PhD thesis, Szent István University.

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy, *International Journal of Forecasting* **22**: 679–688.

Hébert, G., Perez, L. and Harati, S. (2018). An agent-based model to identify migration pathways of refugees: The case of Syria.

ICG (2014). South Sudan: A civil war by any other name, International Crisis Group. Africa Report N217, Brussels, Belguim. Available at: https://www.crisisgroup.org/africa/horn-africa/south-sudan/south-sudan-civil-war-any-other-name.

IRRI (2016). Burundi: A country on the edge, *Technical report*, International Refugee Rights Initiative.

Jacobsen, K. (1996). Factors influencing the policy responses of host governments to mass refugee influxes, *International Migration Review* **30**(3): 655–678.

Jandl, M. (2011). Methods, approaches and data sources for estimating stocks of irregular migrants, *International Migration* **49**(5): 53–77.

Johnson, R. T., Lampe, T. A. and Seichter, S. (2009). Calibration of an agent-based simulation model depicting a refugee camp scenario, *Proceedings of the 2009 Winter Simulation Conference*, pp. 1778–1786.

Kalipeni, E. and Oppong, J. (1998). The refugee crisis in Africa and implications for health and disease: A political ecology approach, *Social Science & Medicine* **46**(12): 1637–1653.

Klabunde, A. and Willekens, F. (2016). Decision-making in agent-based models of migration: State of the art and challenges, *European Journal of Population* **32**(1): 73–97.

Kniveton, D. R., Smith, C. D. and Black, R. (2012). Emerging migration flows in a changing climate in dryland africa, *Nature Climate Change* **2**(6): 444–447.

Kniveton, D., Smith, C. and Wood, S. (2011). Agent-based model simulations of future changes in migration flows for Burkina Faso, *Global Environmental Change* **21**: S34–S40.

Kunz, E. F. F. (1973). The refugee in Flight: Kinetic Models and Forms of Displacement, *International Migration Review* **7**(2): 125–146.

Kurekova, L. (2011). Theories of migration: Conceptual review and empirical testing in the context of the EU East-West flows, *Interdisciplinary conference on Migration, Economic Change, Social Challenge*, pp. 1–37.

Latek, M. M., Mussavi Rizi, S. M. and Geller, A. (2013). Verification through calibration: An approach and a case study of a model of conflict in Syria, *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, IEEE Press, pp. 1649–1660.

Law, A. M. (2005). How to build valid and credible simulation models, *in* M. E. Kuhl, N. M. Steiger, F. B. Armstrong and J. A. Joines (eds), *Proceedings of the 2005 Winter Simulation Conference*, pp. 25–32.

Law, A. M. (2008). How to build valid and credible simulation models, *in* S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson and J. W. Fowler (eds), *Proceedings of the 2008 Winter Simulation Conference*, pp. 39–47.

Lee, E. S. (1966). A theory of migration, *Demography* **3**(1): 47–57.

Lin, L., Carley, K. M. and Cheng, S.-F. (2016). An agent-based approach to human migration movement, *Proceedings of the 2016 Winter Simulation Conference*, pp. 3510–3520.

Lopez-Lucia, E. (2015). Early warning models for irregular migration, *Technical report*, GS-DRC Applied Knowledge Services.

Macal, C. M. and North, M. J. (2010). Tutorial on agent-based modelling and simulation, *Journal of Simulation* **4**(3): 151–162.

Macal, C. M. and North, M. J. (2014). Introductory tutorial: Agent-based modeling and simulation, *in* A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley and J. A. Miller (eds), *Proceedings of the 2014 Winter Simulation Conference*, pp. 6–20.

Martin, A. (2005). Environmental conflict between refugee and host communities, *Journal of Peace Research* **42**(3): 329–346.

Martineau, J. S. (2010). Red flags: A model for the early warning of refugee outflows, *Journal of Immigrant & Refugee Studies* **8**(2): 135–157.

Massey, D. S. (1990). Social structure, household strategies, and the cumulative causation of migration, *Population Index* **56**(1): 3–26.

Massey, D. S., Arango, J., Hugo, G., Kouaouci, A., Pellegrino, A. and Taylor, J. E. (1993). Theories of international migration: A review and appraisal, *Population and Development Review* **19**(3): 431–466.

Mernik, M., Heering, J. and Sloane, A. M. (2005). When and how to develop domain-specific languages, *ACM Computing Surveys* **37**(4): 316–344.

Moore, W. H. (2006). Refugee or internally displaced person?: To where should one flee?, *Comparative Political Studies* **39**(5): 599–622.

Moore, W. H. and Shellman, S. M. (2004). Fear of persecution: Forced migration, 1952-1995, *Journal of Conflict Resolution* **48**(5): 723–745.

Moore, W. and Shellman, S. (2007). Whither will they go? A global study of refugees' destinations, 1965-1995, *International Studies Quarterly* **51**: 811–834.

Neumayer, E. (2005). "Bogus refugees? The determinants of asylum migration to Western Europe, *International studies quarterly* **49**(3): 389–410.

Park, H. J., Jo, W. S., Lee, S. H. and Kim, B. J. (2018). Generalized gravity model for human migration, *New Journal of Physics* **20**: 1–10.

Radu, D. and Straubhaar, T. (2012). Beyond 'Push-Pull': The Economic Approach to Modelling Migration, *in* M. Martiniello and J. Rath (eds), *An Introduction to International Migration Studies*, Amsterdam University Press, chapter 2, pp. 25–55.

Raleigh, C., Kishi, R., McKnight, J. and Moody, J. (2016). Country techreport: Burundi crisis year one, *Technical report*, Armed Conflict Location and Event Dataset.

Raleigh, C., Linke, A., Hegre, H. and Karlsen, J. (2010). Introducing ACLED-Armed Conflict Location and Event Data, *Journal of Peace Research* **47**(5): 651–660.

Raymer, J. and Smith, P. W. F. (2010). Editorial: Modelling migration flows, *Journal of the Royal Statistical Society* **173**(4): 703–705.

Reid, K. (2018). South Sudan conflict, hunger: Facts, FAQs, and how to help, World Division. Available at: https://www.worldvision.org/refugees-news-stories/south-sudan-refugee-crisis-facts.

Reuters (2014). Chad closes border with Central African Republic, Available at: http://www.reuters.com/article/us-centralafrica-chad-idUSBREA4B0M320140512.

Richmond, A. H. (1988). Sociological theories of international migration: The case of refugees, *Current Sociology* **36**(2): 7–25.

Robinson, S. (2008). Conceptual modelling for simulation Part I: Definition and requirements, *Journal of the Operational Research Society* **59**(3): 278–290.

Salehyan, I. (2014). *Forced migration as a cause and consequence of civil war*, Routledge Handbook of Civil Wars. Routledge., chapter 21, pp. 267–278.

Sargent, R. G. (2011). Verification and validation of simulation models, *in* S. Jain, R. Creasey, J. Himmelspach, K. White and M. Fu (eds), *Proceedings of the 2011 Winter Simulation Conference*, pp. 183–198.

Sato, J. B. and Stansen, B. J. (2007). A system dynamics approach to analyzing violence, death and displacement in Darfur, *25th International Conference of the System Dynamics Society*, Boston, pp. 1–22.

Schelling, T. C. (1971). Dynamic models of segregation, *Journal of Mathematical Sociology* **1**: 143–186.

Schmeidl, S. (1997). Exploring the causes of forced migration: A pooled time-series analysis, 1971-1990, *Social Science Quarterly* **78**(2): 284–308.

Schmeidl, S. (2003). The early warning of forced migration: State or human security?, *Refugees and Forced Displacement: International Security, Human Vulnerability, and the State*, newman, e. and selm j.v. edn, United Nations University Press, Tokyo, pp. 130–155.

Schmeidl, S. and Jenkins, J. C. (1998). The early warning of humanitarian disasters: Problems in building an early warning system, *International Migration Review* **32**(2): 471–486.

Shami, S. (1996). Transnationalism and refugee studies: Rethinking forced migration and identity in the Middle East, *Journal of Refugee Studies* **9**(1): 3–26.

Shellman, S. M. and Stewart, B. M. (2007). Predicting risk factors associated with forced migration: An early warning model of Haitian flight, *Civil Wars* **9**(2): 174–199.

Sjaastad, L. A. (1962). The costs and returns of human migration, *Journal of Political Economy* **70**(5): 80–93.

Sokolowski, J. A. and Banks, C. M. (2014). A methodology for environment and agent development to model population displacement, *Proceedings of the 2014 Symposium on Agent Directed Simulation*, Society for Computer Simulation International.

Sokolowski, J. A., Banks, C. M. and Hayes, R. L. (2014). Modeling population displacement in the Syrian city of Aleppo, *Proceedings of the Winter Simulation Conference 2014*, IEEE, pp. 252–263.

Stouffer, S. A. (1940). Intervening opportunities: A theory relating to migration and distance, *American Sociological Review* **5**(6): 845–867.

Suleimenova, D., Bell, D. and Groen, D. (2017). A generalized simulation development approach for predicting refugee destinations, *Scientific Reports* **7:13377**.

Suter, J. L., Groen, D. and Coveney, P. V. (2015). Chemically specifi C multiscale modeling of clay-polymer nanocomposites reveals intercalation dynamics, tactoid self-assembly and emergent materials properties, *Advanced Materials* **27**(6): 966–984.

Swaminathan, J. M., Smith, S. F. and Sadeh, N. M. (1998). Modeling Supply Chain Dynamics, *Decision Sciences* **29**(3): 607–632.

Taylor, J. E. (1984). Differential migration networks information and risk, *in* O. Stark (ed.), *Research in Human Capital and Development*, JAI Press, Greenwich, CT, chapter Vol. 4: Mi.

UNHCR (2010). Convention and protocol. Relating to the status of stateless refugees, *Technical report*, United Nations High Commissioner for Refugees, Geneva.

UNHCR (2012a). UNHCR Mali techreport no. 3, Available at: http://www.unhcr.org/uk/publications/operations/4f7c49999/mali-situation-update-3-30-march-2012.html.

UNHCR (2012b). UNHCR Mali techreport no. 4, Available at: http://www.unhcr.org/uk/publications/operations/4f9952ce9/mali-situation-update-4-24-april-2012.html.

UNHCR (2014a). Inter-agency appeal for the south sudanese refugee emergency, United Nations High Commissioner for Refugees. Available at: https://www.unhcr.org/partners/donors/531f13539/inter-agency-appeal-south-sudanese-refugee-emergency-january-december-2014.html.

UNHCR (2014b). UNHCR Regional update no. 1 CAR situation, Available at: http://data.unhcr.org/car/download.php?id=21.

UNHCR (2015a). Global report 2015, Available at: http://www.unhcr.org/gr15/index.xml.

UNHCR (2015b). South Sudan. Humanitarian response plan, United Nations High Commissioner for Refugees. Available at: https://data2.unhcr.org/en/documents/download/29790.

UNHCR (2018). Figures at a Glance, Available at: https://www.unhcr.org/figures-at-a-glance.html.

Vogel, D., Kovacheva, V. and Prescott, H. (2011). The size of the irregular migrant population in the European Union - Counting the uncountable?, *International Migration* **49**(5): 78–96.

Voutira, E. (1997). Refugees, migrants, oustees, displacees and returnees: A conceptual framework for understanding varieties of displacement, *Technical report*, Narrating Mobility, Boundaries and Citizenship Mangleas.

Wickramasinghe, A. A. I. N. and Wimalaratana, W. (2016). International migration and migration theories, *Social Affairs* **1**(5): 13–32.

Wilde, M., Hategan, M., Wozniak, J. M., Clifford, B., Katz, D. S. and Foster, I. (2011). Swift: A language for distributed parallel scripting, *Parallel Computing* **37**(9): 633–652.

Willekens, F. (2016). Migration flows: Measurement, analysis and modeling, *in* M. J. White (ed.), *International Handbook of Migration and Population Distribution*, Springer Netherlands, Dordrecht, pp. 225–241.

Wood, W. B. (1994). Forced migration: Local conflicts and international dilemmas, *Annals of the Association of American Geographers* **84**(4): 607–634.

Appendix A. Model construction tutorial for forced displacement

A.1 Data extraction

A.1.1 The UNHCR situations

The United Nations High Commissioner for Refugees (UNHCR) database (https://data2.unhcr.org/en/situations) provides an overview of active situations worldwide that are facing forced displacement distress. To construct a new conflict situation:

1. Select an active (conflict) situation of interest from an interactive map (Figure A.1) and click to access data and documentation relevant to a chosen conflict situation (Figure A.2).

Figure A.1: An overview of the UNHCR situations.

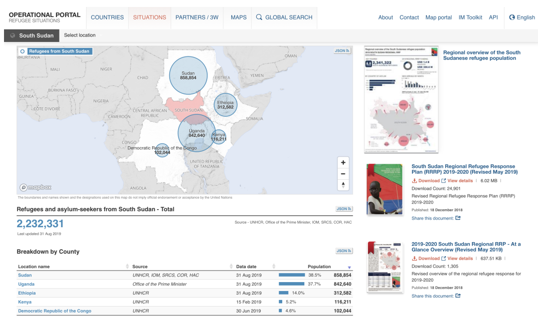

Figure A.2: An overview of the South Sudan situation.

2. Select a simulation period for conflict situation from ‘Refugees and asylum-seekers from *<chosen situation name>* - Total’ timeline, which also presents forced displacement counts for a chosen period (see Figure A.3).

3. Obtain total counts of forcibly displaced people by clicking JSON button of ‘Refugees and asylum-seekers from *<chosen situation name>* - Total’ section.

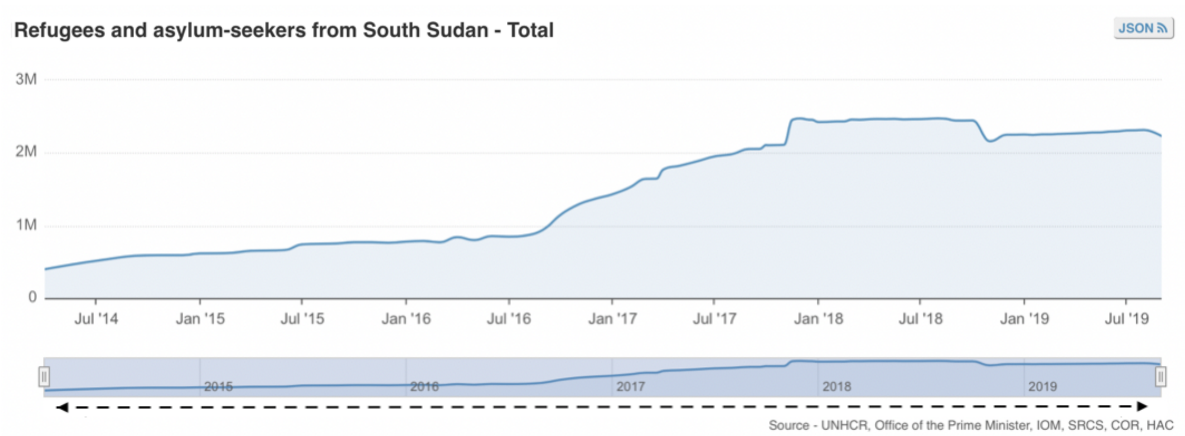

Figure A.3: An example screenshot of total number of refugees and asylum seekers from South Sudan

4. Identify camps for each neighbouring country through 'Breakdown by Country' section of the conflict situation.

5. Collect and save data for each camp (e.g. country_name-camp_name.csv).

A.1.2 The ACLED database

The Armed Conflict Location and Event Data Project (ACLED) database (https://www.acleddata.com/data) provides conflict location data for forced displacement simulations. To obtain data on chosen conflict situation, complete the ACLED data export tool fields (Figure A.4) as follows:

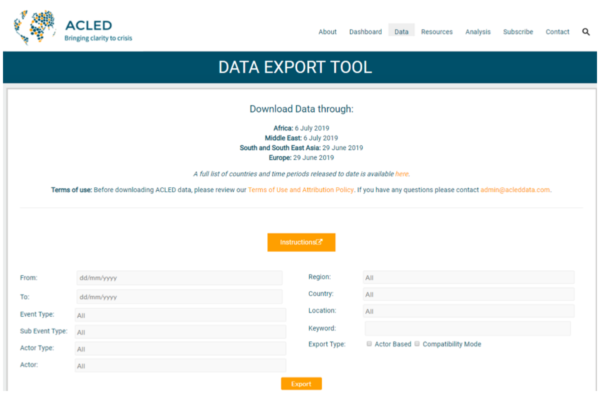

Figure A.4: ACLED data portal tool to obtain conflict locations.

1. Provide dates of interest for conflict situation (i.e. From and To).

2. Select *Event Type: 'Battle'*.

3. Select *Sub Event Type: 'Armed clash', 'Attack', 'Government regains territory' and 'Non-state actor overtakes territory'.*

4. Specify 'Region' and 'Country' of conflict situation choice.

5. Accept 'Terms of Use and Attribution Policy'.

6. <name>.csv file exports to Downloads automatically.

7. Revise the downloaded <name>.csv file:

   - Target the 'fatalities' column and remove all rows in <name>.csv file with fatalities less than 1.

   - Choose the first conflict location occurrence of each location and exclude syndicated (repeated) locations (see an example in Table A.1).

...	Location	Fatalities
...	A	3
...	B	23
...	A	38
...	C	14
...	A	30
...	C	7
...	...	...

Table A.1: An example of conflict locations (A, B and C). Conflict zone A occurs three times with fatality numbers 3, 38 and 30, while conflict zone C repeats twice with fatalities 14 and 7. Choose essence of locations (one of each location) at the first occurrence (e.g. A = 3, B = 23 and C = 14) for simulation purposes.

A.2 Construct input CSV files

A.2.1 Construct an input *locations.csv* file

ACLED conflict data provides conflict locations to construct locations.csv input file (see Table A.2) for simulation purposes. After identifying conflict locations and producing locations.csv, the last column is filled with population data for conflict locations. Population distributions can be obtained from https://www.citypopulation.de or other population databases.

name	region	country	lat	long	location_type	conflict_date	population/capacity
A	AA	ABC	xxx	xxx	conflict	xxx	xxx
B	BB	ABC	xxx	xxx	conflict	xxx	xxx
C	CC	ABC	xxx	xxx	conflict	xxx	xxx
...	...	...	...	...	...	...	...

Table A.2: An example of locations.csv with conflict locations and their attributes.

Input camp names (i.e. destination locations) and their capacity into locations.csv file. Camp capacity is the highest number of forced migrants for each camp and obtained from individual camp CSV files as demonstrated in Table A.3.

...	...
2015-03-31	11470
2015-06-02	12405
2015-07-24	12405
2015-08-31	11359
2015-09-30	8129
...	...

Table A.3: CampZ.csv has the highest number of forcibly displaced people (12405) on 2015-06-02, so we set that as the camp capacity in locations.csv for Camp Z.

Forced displacement registrations for camps have corrections to overcome inaccurate registrations. To consider this factor in the model, we identify level 1 registration representing a decline in forced migrant counts. In the case of this example in Table A.3, there is a drop from 11359 to 8129 and thus we take into account the new registration date – 2015-09-30.

A.2.2 Construct an input *routes.csv* file

Identified conflict zones and camps provide origin and destination locations. We connect these locations to represent how forcibly displaced people flee. We use http://www.bing.com/maps (or other mapping services) to connect conflict zones and camps, and add additional locations (if required) as a location type *town* to locations.csv as illustrated in Table A.4.

name	region	country	lat	long	location_type	conflict_date	population/capacity
A	AA	ABC	xxx	xxx	conflict	xxx	xxx
B	BB	ABC	xxx	xxx	conflict	xxx	xxx
C	CC	ABC	xxx	xxx	conflict	xxx	xxx
Z	ZZ	ZZZ	xxx	xxx	camp	-	xxx
N	NN	ABC	xxx	xxx	town	-	-
...	...	...	...	...	...	...	...

Table A.4: An example of complete locations.csv input file.

Record distances between locations in routes.csv file for simulation by following the format illustrated in Table A.5.

name1	name2	distance [km]	forced_redirection
A	B	x_1	
B	C	x_2	
A	C	x_3	
B	N	x_4	
C	N	x_3	
N	Z	x_5	
...	...	...	

Table A.5: An example illustration of routes.csv input file. forced_redirection refers to redirection from source location (can be town or camp) to destination location (mainly camp) and source location indicated as forwarding_hub. The value of 0 indicates no redirection, 1 indicates redirection (from name2) to name1and 2 corresponds to redirection (from name1) to name2.

A.2.3 Define location and border closures in *closures.csv* file

We identify location or border closure events and document them in closures.csv file. We follow the format illustrated in Table A.6.

closure_type*	name1	name2	closure_start = 0*	closure_end = -1*
location	A	B	xxx	xxx
country	ABC	ZZZ	xxx	xxx
...	...	...	...	...

Table A.6: Illustration of closures.csv. closure_type has 2 possible values: *location* corresponding to camp or town closure and *country* referring to border closure. closure_start and closure_end are given as integers, counting the number of days after the simulation start. The value of 0 indicates the start, while -1 indicates the end of the simulation.

A.2.4 Construct a network map for a conflict situation

Construct an agent-based network map from locations.csv and routes.csv using https://carto.com (see Figure A.5).

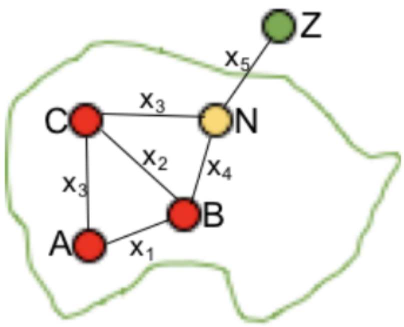

Figure A.5: Example of a network map.

A.3 Validation data

There are three CSV file formats required for validation of simulation outputs:

- <CSV file containing total forced migrant counts>.csv comprises total counts of forcibly displaced people from 'Refugees and asylum-seekers from *<chosen situation name>* - Total' JSON file (explained in section A.1.1) and has the format demonstrated in Table A.7.

...	...
YYYY-MM-DD	xxx
YYYY-MM-DD	xxx
...	...

Table A.7: Illustration of *<CSV file containing total forced migrant counts>.csv* format.

- We obtain data for each camp using the format in Table A.8 and label them as <country_name-camp_name>.csv.

...	...
YYYY-MM-DD	xxx
YYYY-MM-DD	xxx
...	...

Table A.8: Illustration of *<country_name-camp_name>.csv* format that is one data file separate for each camp in the simulation.

- data_layout.csv contains camp names for each camp/destination locations.

Total	<CSV file containing total forced migrant counts>.csv
<camp_name1>	<country_name-camp_name1>.csv
<camp_name2>	<country_name-camp_name2>.csv
...	...

Table A.9: Illustration of *data_layout.csv* format.

A.4 Summary list of data files

To sum up, each conflict situation requires:

1. Three input CSV files

    - <csv_input_directory>/locations.csv
    - <csv_input_directory>/routes.csv

- <csv_input_directory>/closures.csv

2. Validation data files

- <validation_data_directory>/<CSV file containing total forced migrant counts>.csv
- <validation_data_directory>/<country_name-camp_name1>.csv
- <validation_data_directory>/<country_name-camp_name2>.csv
- <validation_data_directory>/data_layout.csv

Appendix B. Simulation execution tutorial for forced displacement

B.1 Setup activities for all required software

B.1.1 System requirements

We require the following prerequisites:

- Linux/iOS/Windows environment
- Python3
- Python libraries:
  - NumPy (see https://www.numpy.org)
  - pandas (see https://pandas.pydata.org)
  - matplotlib (see https://matplotlib.org)
  - PyYaml (see https://pyyaml.org)
  - Fabric3 (see http://www.fabfile.org)
- To perform the Py.test tests:
  - pytest (see https://docs.pytest.org)
  - pytest-pep8 (see https://pypi.org/project/pytest-pep8)

B.1.2 Software packages

To start with, create a directory named **Codes** to store all software packages in one directory.

- **FLEE**

  To install an agent-based simulation code, clone the FLEE repository:

git clone https://github.com/djgroen/flee-release.git

- **FabSim3**

  To install the FabSim3 automation toolkit, clone the FabSim3 repository:

  git clone https://github.com/djgroen/FabSim3.git

  - Ensure FabSim3 is in the home directory (e.g. ~/Codes/FabSim3/) and in your $PYTHONPATH environment variable and that the fabsim command can be launched from the command line. An easy way to accomplish this is by adding the following two lines to the end of your **$HOME/.bashrc** file and modifying path according to your FabSim3 directory:

    ```
    export PATH=(path of your FabSim3 directory)/bin:$PATH
    export PYTHONPATH=(path of your FabSim3 directory):$PYTHONPATH
    ```

    Note: You may have to restart the shell for these changes to apply.

  - Go to the **fabric3_base** subdirectory and run the following command:

    pip3 install Fabric3-1.14.post1-py3-none-any.whl

  - Create **machines_user.yml** file by copying **machines_user_example.yml** in the **deploy**:

    cp deploy/machines_user_example.yml deploy/machines_user.yml

    Modify its contents to match with your local settings. For first (local) testing, one must change the settings under the sections **default:** and **localhost:** so as to update the paths of FabSim directory and lammps executable respectively. To demonstrate, change **username: "<username>"** Further, add the FLEE code location in the **machines_user.yml** file under the **default:** section as follows:

    ```
    flee_location: "~/Codes/flee"
    ```

    Note for Mac Users: Make sure to override the default home directory, by switching the **home_path_template** variable by uncommenting the following line:

    ```
    home_path_template: "/Users/$username"
    ```

  - To enable the use of FabSim on your localhost, type:

    fab localhost setup_fabsim

As part of this command, you will be logging in to your machine through SSH once, which can trigger a password prompt. In this case, simply type the password for the machine in which you are running these commands.

Note for Mac Users: In the **deploy/machines.yml**, change

```
runtime_path_template:"$home_path"
```

to

```
runtime_path_template:"~"
```

- FabSim commands can now be launched using **fab** and **fabsim** commands. The two commands can be used interchangably, although the fabsim command gives clearer outputs and can be launched from anywhere (**fab** can only be used within the FabSim installation directories). If **fabsim** command is not found, simply add **PATH** and **PYTHONPATH** to **~/.bash_profile**, as well as set environment for Python3 in the **bin/fabsim** by replacing

```
#!/usr/bin/python3
```

to

```
#!/usr/bin/env python3
```

- By default, FabSim3 comes with the FabMD plugin installed. Other plugins can be installed and are listed in **deploy/plugins.yml**. To install a specific plugin, simply type:

fabsim localhost install_plugin:<plug_name>

- **FabFlee**

  To install the FabFlee plugin, which is an automated forced displacement modelling workflow, type:

  fabsim localhost install_plugin:FabFlee

  - The FabFlee plugin will appear in **~/Codes/FabSim3/plugins/FabFlee**.
  - To list all available commands in FabFlee, type:

  fabsim -l

- **EasyVVUQ**

  To install verification, validation and uncertainty quantification (VVUQ) Python library for a wide variety of simulations (see https://easyvvuq.readthedocs.io/en/latest/ for detailed documentation), clone the EasyVVUQ plugin:

  git clone https://github.com/UCL-CCS/EasyVVUQ.git

  - To install dependencies required for EasyVVUQ, use pip (or pip3):

    pip install -r requirements.txt

  - The library also requires the installation of **setup.py** as follows:

    python3 setup.py install

  - To test the installation of the EasyVVUQ library, type:

    make -C tests/cannonsim/src/

B.2 Simulation execution with FLEE

1. Create **<country_name>** conflict directory:

   - Create **<country_name>_input_data** sub-directory to store input CSV files
   - Create the second sub-directory **source_data** and place inside the following validation data files:
     - <CSV file containing total forced migrant counts>.csv
     - <country_name-camp_name1>.csv
     - <country_name-camp_name2>.csv
     - data_layout.csv

2. Create <country_name>.py file for a conflict situation. To demonstrate, https://github.com/djgroen/flee-release/blob/master/test_csv.py is an example script, which you can copy and modify according to your choice of conflict scenario.

   - Change date in **<country_name>.py** to the start of conflict simulation date:

```
def date_to_sim_days(date):
  return DataTable.subtract_dates(date,"2010-01-01")
  ...
```

```
d = handle_refugee_data.RefugeeTable(csvformat="generic",
... start_date="2010-01-01", ...)
```

- Declare input and validation data locations in **<country_name>.py** file.

```
ig.ReadLocationsFromCSV("<conflict_name>_input_csv/locations.csv")
ig.ReadLinksFromCSV("<conflict_name>_input_csv/routes.csv")
ig.ReadClosuresFromCSV("<conflict_name>_input_csv/closures.csv")
  ...
d = handle_refugee_data.RefugeeTable(csvformat="generic",
data_directory="source_data/<country_name>/", ...)
```

3. To run **<country_name>.py** file:

   - Create an output directory:

     mkdir out<country_name>

   - Run the following command to execute <country_name>.py and obtain the simulation output:

     python3 <country_name>.py <simulation_period> > out<country_name>/out.csv

     For instance:

     python3 car-csv.py 50 > outcar/out.csv

   - Visualise the simulation output using:

     python3 plot-flee-output.py out<country_name>

4. To analyse and interpret simulation output, open **out<country_name>**, which will contain simulation output and UNHCR data comparison graphs for each camp, as well as average relative difference graph for the simulated conflict situation.

B.3 Simulation execution with FabFlee

1. Create a conflict directory **<country_name>** in **~/Codes/FabSim3/plugins/FabFlee/conflict_data**.

2. In **~/Codes/FabSim3/plugins/FabFlee/conflict_data/<conflict_name>**, place three input CSV data files:

- locations.csv
- routes.csv
- closures.csv

3. In ~**/Codes/FabSim3/plugins/FabFlee/conflict_data/<conflict_name>**, create sub-directory **source_data** and place the following validation data files in source_data directory:

   - <CSV file containing total forced migrant counts>.csv
   - <country_name-camp_name1>.csv
   - <country_name-camp_name2>.csv
   - data_layout.csv

4. Clear active conflict directory in ~**/Codes/FabSim3/plugins/FabFlee/conflict_data** to clear out previous simulation instances (if required):

   fabsim localhost clear_active_conflict

5. Load **<country_name>** conflict to working directory:

   fabsim localhost load_conflict:<conflict_name>

6. Modify **run.py** in ~**/Codes/FabSim3/plugins/FabFlee/conflict_data/active_conflict** for an individual conflict scenario:

   - Change date in **run.py** to the start of conflict simulation date:

```
def date_to_sim_days(date):
  return DataTable.subtract_dates(date,"2010-01-01")


  d = handle_refugee_data.RefugeeTable(csvformat="generic",
  data_directory=validation_data_directory, start_date="2010-01-01",
  data_layout="data_layout.csv")
```

   - Change simulation period in **run.py** to the duration of conflict scenario:

```
  end_time = 100
  last_physical_day = 100
```

7. Instantiate the constructed conflict model using the following command:

   fabsim localhost instantiate:flee_<conflict_name>

8. Run the conflict simulation using:

   fabsim localhost flee:flee_<conflict_name>,simulation_period=<number>

9. Copy back any results from completed runs, simply type:

   fabsim localhost fetch_results

   The results will then be in a directory inside **~/Codes/FabSim3/results**, which is most likely called <conflict_name>_localhost_16.

10. Plot the simulation output using:

    fabsim localhost plot_output:<conflict_name>_localhost_16,out

11. To analyse and interpret simulation output, open **~/Codes/FabSim3/results/<conflict_name>_localhost_16/out** containing simulation output and UNHCR data comparison graphs for each camp, as well as average relative difference graph for the simulated conflict situation.

B.4 VisualFlee: Simulation output visualisation

To create a conflict map of interest with simulation output using VisualFlee:

1. To install the VisualFlee visualisation, clone the VisualFlee repository:

   git clone https://github.com/cspgdds/Visualflee.git

2. Copy locations.csv and out.csv files of conflict situation to the VisualFlee repository. *Note:* out.csv file requires simulation output for all locations in the simulation as stated in locations.csv including conflict zones, towns and camps.

3. Create **<conflict_name>_timehistory.py** file by copying **map_camps_timehistory.py** and modify the script as follows:

- Change CSV file names (if required) and the start date of the visualisation according to the conflict situation:

```
def make_features(locations_file='locations.csv',
  timeseries_file='out.csv',
  startdate='2010-01-01'):
```

- Change .jsonp name to <conflict_name>:

```
mgj.write_geojson_from_features('<conflict_name>.jsonp', features)
```

4. Execute **<conflict_name>_timehistory.py** file to process input data of the conflict.

5. Create **<conflict_name>.html** by copying **flight.html** example file and modify the script to

- Position the conflict map by adding coordinates of capital city or main focus location:

```
  enter: new L.LatLng(-0.0, 0.0),
```

- Change .jsonp file in **<conflict_name>.html** by inserting the name of <conflict_name>.jsonp:

```
 <script src="<conflict_name>.jsonp"></script>
```

6. Open **<conflict_name>.html** to visualise simulation output of chosen conflict situation. In Figure B.1, we illustrate the VisualFlee screenshot for the Mali conflict.

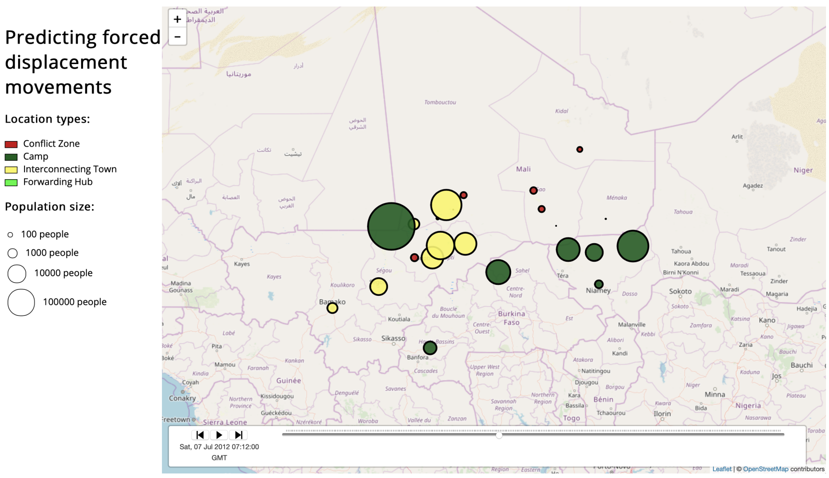

Figure B.1: An example screenshot of simulation output visualisation using VisualFlee for the Mali conflict.

Appendix C. Sensitivity Analysis

Based on:

Suleimenova, D., Bell, D. and Groen, D. (2017), "A generalized simulation development approach for predicting refugee destinations". Scientific Reports, 7 (13377).

We performed a range of sensitivity tests, which we present below. In total we have performed 24 simulations (8x3) to test the accuracy across different movement rate limits, 15 simulations to test the accuracy across different awareness ranges, and, as our simulations have probabilistic components, 30 simulations to test the variability between instances when using identical parameters.

C.1 Variability of runs using identical settings

Using awareness level 1 and forced population movement speed of up to 200km/day, we ran 10 identical simulations of each situation to determine the variability in our error measure due to the randomised elements of our simulations. We present an overview of these variations in Table C.1. Here, we find that for rescaled simulations (the ones for which we present the results in Chapter 5), the 95% confidence interval of the averaged relative difference is within 0.00084 of the obtained averaged relative difference value (see Figure C.1).

Run name	**average**	**confidence** (min)	**confidence** (max)	**confidence(max)** average
Burundi (non-rescaled)	0.4391	0.4397	0.4402	0.0005
CAR (non-rescaled)	0.2795	0.2811	0.2827	0.0016
Mali (non-rescaled)	0.2825	0.2832	0.2839	0.0007
Burundi (rescaled)	0.2949	0.2957	0.2966	0.0008
CAR (rescaled)	0.2877	0.2882	0.2888	0.0006
Mali (rescaled)	0.2299	0.2308	0.2316	0.0008

Table C.1: 95% confidence interval of the averaged relative difference across runs for the three conflict simulations, each of which have been run 10 times in total. Top three rows present non-rescaled simulation runs, while bottom rows represent the rescaled simulations for different instances.

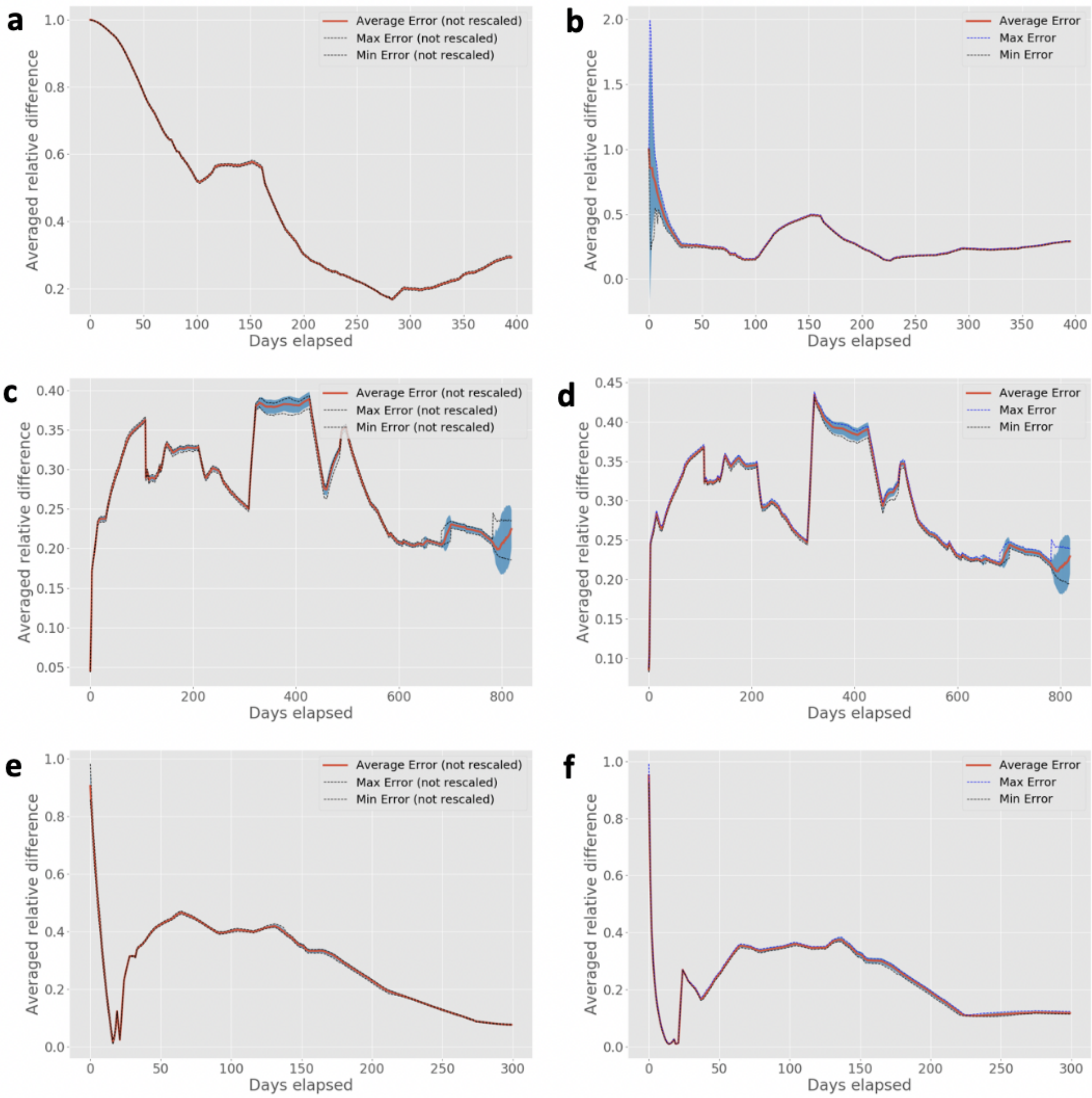

Figure C.1: Visual representation of the 95% confidence interval of the averaged relative difference across runs for the three conflict simulations. These comparisons are respectively for (**a, b**) the Burundi simulations (top row), (**c, d**) the CAR simulations (middle row) and (**e, f**) the Mali simulations (bottom row). The left three columns represent non-rescaled runs while the right three columns present rescaled simulations for all three conflicts.

We observe an apparent variation dispersion for three graphs (b, c, d) in Figure C.1. Specifically, the rescaled average relative difference for Burundi in the first 10-20 days is scattered due to the small sample size of forcibly displaced people in the simulation. While the non-rescaled and rescaled average relative differences for CAR have similar trends in the middle and at the end of the simulation period. These changes are due to border closures between CAR and DRC for the first 200 days and between CAR and Chad after 163 days till the end of the simulation, as well as a new camp opening within DRC (Bili camp) after 500 days into the simulation.

C.2 Comparison of simulation accuracy across different movement rate limits for forced population displacement

We are able to adjust the maximum movement speed of forced population agents within our model. Forced population will never move faster than this limit, but will move slower for example when agents stop at an intermediate (non conflict-zone) locations. We present the averaged relative difference of our simulation as a function of this parameter in Table C.2. Increasing this limit to larger values reduces the error in non-rescaled simulations because it tends to reduce the mismatch between a total number of encamped forced displacement in the simulation, and in the UNHCR data set. Please remember here that we place forced population in conflict areas on the day that they are registered in camps in the UNHCR data, the departure dates of forced migrants are not contained in the dataset.

However, when we look at rescaled simulations, we find that the averaged relative difference decreases only up to a point when we increase the movement rate limit. Beyond a limit of 200 km/day, we do not observe any additional accuracy benefits. For the simulations in our main paper, we, therefore, use a maximum forced population movement speed of 200 km/day.

Maximum forced population speed [km/day]	**Averaged relative difference** Not rescaled	Rescaled
25	0.4484	0.3566
50	0.3949	0.3127
100	0.3539	0.2796
150	0.3432	0.2735
200	0.3370	0.2662
250	0.3336	0.2664
500	0.3253	0.2636
1000	0.3233	0.2611

Table C.2: Averaged relative difference for each simulation using different speed limits of forced population displacement, with an awareness range of 1 link away. Values without rescaling can be found in column 2, and values with rescaling in column 3. The agent-based simulations presented in the main paper have a maximum speed of 200 km/day.

C.3 Comparison of simulation accuracy across different levels of agent awareness

Within our simulation model, we are able to adjust our algorithm to incorporate a more wide or narrow awareness level for the agents. We present the averaged relative difference between our simulations and the UNHCR data as a function of different awareness levels in Table C.3. Agents can be aware of the presence of paths only ("Unweighted"), of only the length of the path to the nearest settlement ("Path distance only"), the type of nearest settlement ("1 link away"), the type of settlements adjacent to neighbouring settlements ("2 links away"), and the type of settlements neighbouring those neighbours of neighbours ("3 links away"). The agent-based simulations presented in the main paper have an awareness range which stretches up to locations 1 link away from their current position, and uses rescaling.

	Averaged relative difference	
Awareness range	Not rescaled	Rescaled
Unweighted	0.3569	0.3416
Path distance only	0.3407	0.2664
1 link away	0.3370	0.2662
2 links away	0.3259	0.2676
3 links away	0.3240	0.2625

Table C.3: Averaged relative difference for each simulation using different levels of agent awareness range. Values without rescaling can be found in column 2, and values with rescaling in column 3.

C.4 Comparison of simulation accuracy across different attractiveness values for camps and conflict zones

Within our simulation model, we are able to adjust our algorithm to increase or decrease the attractiveness value for camp locations, as well as for conflict zones. We present the averaged relative difference between our simulations and the UNHCR data as a function of the different camp attractiveness values in Table C.4, and as a function of the different conflict attractiveness values in Table C.5. For the rescaled simulations, we find that the averaged relative difference varies by less than 0.008 throughout the range of tested parameter values. As a result, the quality of our validation is relatively insensitive to the choice of these two parameters.

Camp weight multiplier	Averaged relative difference Not rescaled	Rescaled
1.5	0.3389	0.2654
1.75	0.3378	0.2658
2 (default)	0.3366	0.2688
2.25	0.3366	0.2702
2.5	0.3366	0.2730

Table C.4: Averaged relative difference for each simulation using different weight multipliers for camps in agent destination selection (with a value of 2.0 making camps twice as likely to be chosen as destination compared to other locations). Values without rescaling can be found in column 2, and values with rescaling in column 3.

Conflict zone weight multiplier	Averaged relative difference Not rescaled	Rescaled
0.15	0.3431	0.2767
0.2	0.3401	0.2721
0.25 (default)	0.3372	0.2699
0.3	0.3348	0.2651
0.35	0.3340	0.2642

Table C.5: Averaged relative difference for each simulation using different weight multipliers for conflict zones in agent destination selection (with a value of 0.25 making conflict zones four times less likely to be chosen as destination compared to other locations). Values without rescaling can be found in column 2, and values with rescaling in column 3.

C.5 Comparison of simulation accuracy across different move probabilities for agents.

Within our simulation model, we are able to adjust our algorithm to increase or decrease the probability that agents move from their current location to a different one on a given day. This probability is set according to the type of location where an agent resides, which is either a conflict zone, camp, or other (default) location. We present the averaged relative difference between our simulations and the UNHCR data as a function of the different conflict zone move probabilities in Table C.6, and as a function of the different camp move probabilities in Table C.7. For the rescaled simulations, we find that the averaged relative difference varies by less than 0.002 throughout the range of tested parameter values for conflict move probabilities and that the quality of our validation is relatively insensitive to the choice of these two parameters. However, our simulation results are somewhat sensitive to camp move probabilities, where a move probability of 0.05 (i.e., an agent will remain in a camp for 20 days on average, rather than 1000 days) will lead to an increase in error of 0.03. Likewise, reducing this parameter to 0.00001 (i.e., an agent will remain in a camp for 100000 days on average) will lead to an error increase of approximately 0.01.

	Averaged relative difference	
Run type	Not rescaled	Rescaled
1.0 (default)	0.3370	0.2695
0.95	0.3370	0.2692
0.9	0.3367	0.2693
0.85	0.3362	0.2684
0.8	0.3376	0.2702

Table C.6: Averaged relative difference for each simulation using different move probabilities for agents which reside in conflict zones (default=1.0). Values without rescaling can be found in column 2, and values with rescaling in column 3.

	Averaged relative difference	
Run type	Not rescaled	Rescaled
1e-05	0.3394	0.2788
0.0001	0.3390	0.2784
0.001 (default)	0.3375	0.2694
0.01	0.3499	0.2668
0.05	0.4092	0.2990

Table C.7: Averaged relative difference for each simulation using different move probabilities for agents which reside in camps (default=0.001). Values without rescaling can be found in column 2, and values with rescaling in column 3.

Table C.8 present the averaged relative difference between these three simulations and the UNHCR data as a function of the different default move probabilities. The default move chance is the main parameter that determines how frequently forced population agents move away from their current location towards an adjacent one. For the rescaled simulations, the simulation results are sensitive to the value of this parameter, with lower values leading to a higher error (up to 0.026 higher with move probability 0.2), and higher values leading to a smaller error (up to 0.029 lower with move probability 1.0). The errors for the non-rescaled simulations decrease more drastically with this move probability, with an error of 0.378 for probability value 0.2 and 0.253 for probability value 1.0.

Run type	Averaged relative difference	
	Not rescaled	Rescaled
0.2	0.3779	0.2938
0.25	0.3557	0.2798
0.3 (default)	0.3368	0.2678
0.35	0.3237	0.2634
0.4	0.3114	0.2578
0.45	0.3010	0.2532
0.5	0.2928	0.2509
0.6	0.2782	0.2467
0.7	0.2680	0.2423
0.8	0.2609	0.2407
0.9	0.2558	0.2398
1.0	0.2526	0.2386

Table C.8: Averaged relative difference for each simulation using different default move probabilities for agents which reside in locations other than camps and conflict zones (default=0.3). Values without rescaling can be found in column 2, and values with rescaling in column 3.

There are multiple possible explanations for these results. For example, it is possible that an initial assumption (forced population have a chance of 0.3 on any given day to depart a non-conflict location in the country of conflict, and therefore stay in such a place for an average duration of 3.33 days) was an underestimation, and that forced population tend to move from place to place much more frequently. However, it is also possible that a higher value for this parameter is unrelated to the accuracy of the associated assumption, and that the error becomes lower simply because the mismatch in forced population arrivals between simulation and data is artificially reduced. The observed narrowing of the gap between the non-rescaled and rescaled errors in the results (from 0.084 for probability 0.2 to 0.014 for probability 1.0) could be explained by this latter theory.