Dynamic contagion spread modelling over emergent spatio-temporal contact networks

by

Avinaba Mistry

Bachelor of Computer Science and Engineering, Jadavpur University, India, 2014

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Computer Science

In the Graduate Academic Unit of Computer Science

Supervisor(s):	Suprio Ray, Ph.D., Computer Science
	Sanjeev Seahra, Ph.D., Mathematics and Statistics
Examining Board:	Kalikinkar Mandal, Ph.D., Computer Science
	Przemyslaw Pochec, Ph.D., Computer Science
	James Watmough, Ph.D., Mathematics and Statistics

This thesis is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

December, 2022

© Avinaba Mistry, 2022

Abstract

Analysis, prediction and decision support for contagion spread and its associated risk with Ordinary Differential Equation models proved to be challenging in terms of incorporating the underlying heterogeneity of emergent spatio-temporal contact networks of the moving objects. These models reduce the granularity and scope of analysis of possible pharmacological and non-pharmacological intervention measures. To address these limitations, we propose to aggregate population level dynamics by modelling individual-level interactions. To that end, first we introduce an individual level agent-based stochastic contagion simulation modelling framework as a possible solution for adapting to rapidly changing parameters of a contagion including SARS-CoV-2. Second, we propose a spatio-temporal index to extract contact networks from distributed real-world mobility data stores.

The framework has been implemented declaratively in R and functionally in Julia and the spatio-temporal index in Python. We conduct thorough experimental evaluation to show the viability and applicability of the proposed approaches.

Dedication

I am dedicating this thesis to three members of my family who have never stopped believing in my dreams. First and foremost, to my sister Oly Mistry, who has always seen the best of me and continues to guide me in my pursuits. Next, my father Amulya Ratan Mistry and mother Nartansree Mistry, who instilled values of hard-work, rigour and empathy from my very childhood.

Finally I dedicate this thesis to all my educators who made sure I kept asking questions.

Acknowledgements

I would like to express my deepest gratitude to my supervisors Dr. Suprio Ray, Associate Professor of Computer Science at the University of New Brunswick and Dr. Sanjeev Seahra, Professor and Department Chair of Maths and Statistics at the University of New Brunswick and Director of Atlantic Association for Research in the Mathematical Sciences (AARMS). I am also deeply indebted to Dane Sheppard, Chief Technology Officer of our research partner organisation: The Black Arcs. Without their unyielding trust, guidance, patience and feedback I could not have undertaken this journey. I would like to extend my sincere thanks to the members of the research groups, Bilal Husain, Gaia Noseworthy, Jarod Kelly, Robert Santacruz and Suvam Kumar Das for their inputs and inspiration. Furthermore, this undertaking would not have been possible without generous support from Mitacs and The Black Arcs, who financed my research. Finally, I'd like to mention my family and friends, their emotional support has been instrumental for keeping a level head and motivations high during the course of this journey. I would also like to thank the local coffee roasters for fuelling late night study sessions.

Table of Contents

Abstract **ii**

Dedication **iii**

Acknowledgments **iv**

Table of Contents **v**

List of Tables **xi**

List of Figures **xii**

Abbreviations **xvi**

1 Introduction **1**

1.1 Motivation 1

1.2 Thesis problem statement 2

1.3 Aim of this thesis 3

1.4 Significance of this thesis 4

1.5 Research and development methodology 4

1.6 Techniques implemented as part of this thesis 5

1.6.1 Estimation of contagion risk based on simulation on emergent social network 5

1.6.2 Facilitate efficient generation of individual contact network from massive uncertain mobility trajectory datasets 5

1.7 Main contributions of this thesis 6
1.8 Summary 6

2 Background and Related Work **7**
2.1 Agent-Based Modelling 7
2.1.1 Utility of ABMS 7
2.1.2 ABMS application domains 8
2.1.3 Structure of ABMS 9
2.1.4 Definition of an agent 10
2.1.5 Individual level agent-based modelling 10
2.1.6 IL-ABMS for contagion modelling 12
2.1.7 Human mobility modelling 13
2.1.7.1 Citisketch™ - The Black Arcs [46] 14
2.1.8 Toronto Area Scheduling Model for Household Agents 14
2.2 Infectious disease modelling as compartments 16
2.2.1 SIR model as systems of ODEs 16
2.2.2 Parameterised β as social network 17
2.2.3 Outline of compartmental models 18
2.2.4 Simulation of contagion as a social experiment 19
2.3 SARS-CoV-2 21
2.3.1 Diagnostic testing 22
2.3.2 Contact tracing 25
2.3.3 Effects of vaccination 25
2.3.4 Effect of other non-pharmaceutical interventions 26
2.4 Related work on contagion modelling 26
2.4.1 Combination of NPIs 30
2.4.2 Composition of models 31
2.5 Real-world mobility data 34

2.6 Uncertain Trajectories 34
2.6.1 Beads and Necklaces 35
2.7 Measuring contact over uncertain trajectories 39
2.7.1 Indexing uncertain trajectories for efficient access 39
2.8 UTM-tree construction challenges 40
2.9 Metric space indexing 43
2.10 Alternative candidate indexing techniques 43
2.11 Geometric Near-Neighbour Access Tree 44
2.11.1 Split-point and Tree Construction 44
2.11.2 Spatial pruning 45
2.11.3 Visualising range query for 3D metric spaces 46
2.12 Summary 47

3 Problem analysis and research overview 48
3.1 Contagion simulation workflow 48
3.2 Incorporating real-world contact network 50
3.3 Description of the data available 50
3.4 Research overview 51
3.5 Summary 51

4 Contagion Simulator 52
4.1 Designing compartmental contagion model 52
4.1.1 Encoding a contagion model as a directed graph 53
4.1.2 Transition matrix 54
4.1.3 Non-infection transition as Markov Chain 55
4.2 Interpreting contact events 55
4.3 Infection propagation 56
4.3.1 Infection probability function 56

4.4 Vaccination 57
4.5 Diagnostic testing and isolation 57
4.5.1 Diagnostic Test transition parameters 61
4.5.2 Diagnostic Test scheduling 61
4.6 Contact tracing 62
4.6.1 Contact gathering flow 64
4.7 Hospitalisations and deaths 64
4.8 Infectiousness scaling 67
4.9 Non pharmaceutical interventions 69
4.10 Simulation algorithm 69
4.10.1 SIR simulation loop 69
4.10.2 Augmented simulation loop 71
4.10.2.1 Naive simulation algorithm 72
4.10.2.2 Optimised simulation algorithm 72
4.10.2.3 Functional simulation algorithm 75
4.11 Implementation 77
4.11.1 Complete simulator: R 77
4.11.2 Simulation loop: Julia 77
4.11.3 Experimental setup 77
4.12 Results 77
4.12.1 Visualising infection risks 79
4.12.1.1 Multiple instances of single scenario 79
4.12.1.2 Comparing scenarios 80
4.13 Summary 82

5 Spatio-Temporal Indexing 85
5.1 GNAT index for non-metric data 87
5.2 Extending GNAT for uncertain trajectory 88

5.2.1 UT-GNAT node structure 88
5.2.2 Distance function for symmetric range in metric space 89
5.2.3 Doubly ranged search query and space-time pruning 90
5.3 Encoding GNAT on regular dataframe as data-store 92
5.3.1 Column encoding for pandas dataframe 92
5.4 Encoding GNAT on distributed dataframe as data-store 92
5.4.1 Unpacking complex data type to columns for Modin dataframe 92
5.4.2 Equivalence with Python object based implementation 93
5.5 Experimental evaluation 93
5.5.1 Single UT-GNAT index 94
5.5.2 Performance loss due to inefficient encoding 94
5.6 Efficiency gain by parallel and distributed indices 95
5.6.1 Parallel: Implementation and evaluation 95
5.6.2 Parallel: Results 96
5.6.3 Distributed: Implementation and evaluation 96
5.6.4 Distributed: Results 100
5.7 Summary 100

6 Conclusion and Discussion 103
6.1 Individual level contagion simulation 103
6.1.1 Limitations and possible extensions 104
6.2 Efficient uncertain trajectory indexing 105
6.2.1 Limitations and possible extensions 105
6.3 Future work 106

Bibliography 107

Appendix 120

A **121**

A.1 Disease models as transition matrix 121

B **122**

B.1 Parallel query execution experiment results 122

B.2 Distributed query execution experiment results 129

Vita

List of Tables

2.1 Example ABMS problem statements 11

2.2 List of notations of SIR-ODE, Equation 2.1 17

2.3 List of notations of Transmission Rate Equation 2.3 21

2.4 Parameter set of network contagion model by Firth et al. [29] 23

2.5 Aggregated diagnostic test accuracy by Dinnes et al. [26] 24

2.6 Diagnostic test confusion matrix by Trevethan et al. [88] 24

2.7 Contact tracing model parameters by Kretzschmar et al. [55] 25

4.1 SIR transition matrix 54

4.2 SLIR transition matrix 54

4.3 List of notation for extended SIR model shown in Figure 4.9 66

4.4 Typical simulation configuration parameters 70

4.5 Comparing simulation time varying disease model and implementation 78

5.1 Parameter list for initial UT-GNAT experiments 93

A.1 SIR disease model - transition matrix 121

A.2 SIR with Testing and Isolation disease model - transition matrix . . . 121

List of Figures

2.1 Structure of a typical agent in Macal et al. [61] 10

2.2 Haslemere experiment workflow 20

2.3 Compartmental model of Episim in Manout et al. [63] 28

2.4 Schematic of SARS-CoV-2 transmission model in Olivera et al. [71] . 32

2.5 Example of composing a COVID-19 model from three smaller models in Baez et al. [5] 33

2.6 Typical Trajectory in Pfoser et al. [77] 35

2.7 Uncertain Trajectory Taxonomy referred to by Trajcevski et al. [87] from Lange et al. [58] 36

2.8 Approximating Beads as Circles 37

2.9 Necklace projected on GIS data provided by Yuan et al. [96] 37

2.10 Necklace projected on Cartesian plane 37

2.11 Contact similarity for overlap between a virtual landmark and two beads from different moving objects, Zhang et al. [98] 38

2.12 3 UTM Tree indexing 30 beads generated from a real-world taxi trajectory from Yuan et al. [96] demonstrating overlaps of bounding circles of root node of each tree. 40

2.13 2D GNAT Space splitting in Brin [12] 45

2.14 Pruning branches using ranges, Brin [12] 46

2.15 Range query (in blue) in 3D metric space with GNAT 47

2.16 Beads in $X \times Y \times T$ 47

3.1 Contagion simulator workflow 49

4.1 Compartmental models as directed graph, SIR on the left, and SLIR on the right. 53
4.2 Composition of base models to capture effect of vaccination on disease dynamics 58
4.3 Diagnostic testing flow 59
4.4 SIR Compartmental model augmented for testing. 59
4.5 SIR Compartmental model augmented for testing and isolation. . . . 61
4.6 SIR Compartmental model augmented for testing, isolation and contact tracing. 62
4.7 Contact gathering flow 65
4.8 Shorthand for compartments with diagnostic testing 67
4.9 SIR compartmental model augmented for testing, isolation, contact tracing, recoveries, deaths and hospitalisations. 68
4.10 Daily active cases for 200 runs 79
4.11 Comparing school closure as a strategy to mitigate infection risk . . 80
4.12 Comparing the effects of diagnostic testing strategy on infection risk 81
4.13 Comparing the effects of reactive venue closures on infection risk . . . 82
4.14 Comparing relative infection risks for venue types 83

5.1 Beads in $X \times Y \times T$ 86
5.2 On the time-flattened figure on the left, the initial range of the space time query is coloured red. The gray circle is the query range augment by r^i_{max} which produces 6 candidate Beads to check for overlap to find the actual result. The figure on the right is visualising the final result in 3D space for clarity. 88
5.3 Single UT-GNAT node visualised with JSON tree visualiser [47] . . . 89

5.4 Max and min spatial distances between two Beads 91
5.5 Time complexity of (Pythonic) UT-GNAT Please note that there are separate logarithmic time axes for building and querying on the left and right side respectively 94
5.6 Low selectivity, parallel query on 1000000 trajectory segments 96
5.7 Comparing effects of increasing data volume on execution time of parallel query with low selectivity 97
5.8 High selectivity, parallel query on 1000000 trajectory segments 98
5.9 Comparing effects of increasing data volume on execution time of parallel query with high selectivity 99
5.10 Comparing effects of increasing data volume on execution time of distributed query with low selectivity 101
5.11 Comparing effects of increasing data volume on execution time of distributed query with high selectivity 102

B.1 Low selectivity, parallel query on 1131 trajectory segments 123
B.2 Low selectivity, parallel query on 2000 trajectory segments 123
B.3 Low selectivity, parallel query on 5000 trajectory segments 123
B.4 Low selectivity, parallel query on 10000 trajectory segments 124
B.5 Low selectivity, parallel query on 20483 trajectory segments 124
B.6 Low selectivity, parallel query on 50000 trajectory segments 124
B.7 Low selectivity, parallel query on 100000 trajectory segments 125
B.8 Low selectivity, parallel query on 248728 trajectory segments 125
B.9 Low selectivity, parallel query on 1000000 trajectory segments 125
B.10 High selectivity, parallel query on 1131 trajectory segments 126
B.11 High selectivity, parallel query on 2000 trajectory segments 126
B.12 High selectivity, parallel query on 5000 trajectory segments 126
B.13 High selectivity, parallel query on 10000 trajectory segments 127

B.14 High selectivity, parallel query on 20483 trajectory segments 127
B.15 High selectivity, parallel query on 50000 trajectory segments 127
B.16 High selectivity, parallel query on 100000 trajectory segments 128
B.17 High selectivity, parallel query on 248728 trajectory segments 128
B.18 High selectivity, parallel query on 1000000 trajectory segments 128
B.19 Low selectivity, distributed query on 10000 trajectory segments . . . 130
B.20 Low selectivity, distributed query on 20000 trajectory segments . . . 130
B.21 Low selectivity, distributed query on 50000 trajectory segments . . . 130
B.22 Low selectivity, distributed query on 100000 trajectory segments . . . 131
B.23 High selectivity, distributed query on 10000 trajectory segments . . . 131
B.24 High selectivity, distributed query on 20000 trajectory segments . . . 131
B.25 High selectivity, distributed query on 50000 trajectory segments . . . 132
B.26 High selectivity, distributed query on 100000 trajectory segments . . . 132

List of Symbols, Nomenclature or Abbreviations

Abbreviations	Meaning	Page
AARMS	Atlantic Association for Research in the Mathematical Sciences	iv
ABMS	Agent based modelling and simulation	7
AESA	Approximating Eliminating Search Algorithm	106
BLE	Bluetooth Low Energy	34
BST	Bisector Tree	44
CI	Confidence interval	24
CSQ	Contact Similarity Query	39
EPT	Extreme Pivot Table (tree)	44
FNR	False Negative Rate (of a diagnostic test)	61
FPR	False Positive Rate (of a diagnostic test)	62
GNAT	Geometric Near-Neighbour Access tree	44
GPS	Global Positioning System	20
GPU	Graphics processing unit	104
IBb	Individual bead bound (for uncertain trajectory segment)	36
IL-ABMS	Individual level agent based modelling and simulation	12
k-NN	k Nearest Neighbour	44
LB	Large Bound (for uncertain trajectory segment)	36
MATSIM	Multi-Agent Transport Simulation	13
MVPT	Multiple Vantage Point tree	44
NPI	Non pharmaceutical intervention	26
ODE	Ordinary Differential Equation	2
PCR	Polymerase chain reaction (diagnostic test)	60
PCT	Proximity based Contact Tracing	34
POCT	Pick-Up Point of Care (diagnostic test)	60

continued on next page

Abbreviations	Meaning	Page
R	Basic reproduction number	29
RAT	Rapid Antigen (diagnostic test)	60
RT-PCR	Reverse transcription polymerase chain reaction (diagnostic test)	22
SARS-CoV-2	Severe acute respiratory syndrome coronavirus 7	ii
SEIR	Susceptible-Exposed-Infectious-Recovered (disease model)	29
SIR	Susceptible-Infectious-Removed (disease model)	16
SIS	Susceptible-Infectious-Susceptible (disease model)	16
SLIR	Susceptible-Latent-Infectious-Recovered (disease model)	26
TASHSA	Toronto Area Scheduling Model for Household Agents	14
TF	Temporal-First (search approach)	39
TNR	True Negative Rate (of a diagnostic test)	60
TPR	True Positive Rate (of a diagnostic test)	60
UBb	Uniform bead bound (for uncertain trajectory segment)	36
UT-GNAT	Uncertain Trajectory GNAT	88
UTM-tree	Uncertain Trajectory M-tree	39
WHO	World Health Organisation	1

Chapter 1

Introduction

1.1 Motivation

At the time of writing this thesis, 214 days have passed since the World Health Organisation (WHO) declared Covid-19 as a pandemic [22]. The cumulative positively identified infections has risen to 622,232,266 with a death toll of 6,559,392 as reported by COVID-19 Dashboard by the Center for Systems Science and Engineering at Johns Hopkins University [27]. The original contagion strain showed rapid mutation and exhibited changes in infection propagation, latent period, symptoms and fatality proportions.

This predicament necessitates utilising public health interventions to contain and mitigate the risk of infection by obstructing the propagation of contagion. However, depending on the selection of intervention measures, the time of implementation and the coverage area of its effects, the efficacy of the intended outcome exhibit significant variance. In addition to the technical complexities, executing these interventions requires mobilisation of large amount of resources and has psycho-social consequences. Therefore it is of utmost importance for public health policy makers to be informed with optimal intervention strategies for a given scenario.

Keeling, et al. [49] notes that a large portion of disease models are based on a compartmentalisation of individuals according their disease status. And that the basic models describe the count or proportion of the population that are susceptible to, infected and recovered from a particular disease, thereby simplifying the details of progression of infection. This approach generates sets of Ordinary Differential Equation (ODE), that has a long and successful history. Although this approach is effective for modelling long enough periods of time that ensures a more or less homogeneous mixing of the population, it proves to be challenging to encode the inherent heterogeneity of emergent social networks and changing dynamics of interventions and contagion. Recent efforts in disease surveillance utilise mobile app-based distance logging between participating individuals. As Cicala et al. [19] notes, these strategies are limited by a lack of participation of individuals. Moreover, the collected spatio-temporal data suffers from uncertainty because of inaccurate sensors and unpredictable human behaviour.

One of the approaches to tackle the heterogeneity of demography, geography, mobility, multiple concurrent strains of a disease, effects of intervention measures and uncertainty of disease surveillance would be to model interactions of individuals to observe the population level outcome. Hence, the research activities undertaken organisations around the world have been investigating individual level agent based modelling approaches to simulate spatio-temporal disease propagation outcomes for a population and to incorporate real-world spatio-temporal data integration to increase model accuracy.

1.2 Thesis problem statement

Overall the thesis problem centres around providing decision support to public health efforts for estimating risk posed by a contagion, for a specific socioeconomic and ge-

ographic context with respect to changing intervention scenarios through individual level models. This involves providing forecasting and counterfactual simulation capabilities to public health with observable parameters for strategic decision making. This approach entails efficient generation of individual level contact data from high volume of uncertain spatio-temporal data. Please note, for the scope of this thesis, we are considering contagions transmitted by person to person contacts.

The questions that are addressed in this theses are as follows:

RQ1 How might we go about incorporating individual level human mobility and contagion spread model for a given geography?

RQ2 How might we facilitate efficient processing of real-world mobility data for contagion spread analysis?

RQ3 How might we provide building and analysis of "what-if" scenarios for future public health policies?

RQ4 How might we effectively communicate the possible outcomes from a stochastic simulation tool?

1.3 Aim of this thesis

The aim is to design and implement a framework that captures the real-world spatio-temporal complexities of modelling a rapidly changing contagion with respect to intervention strategies and geo-spatial variance. First, the framework needs to be extensible and provide different levels of control for different groups of users, for example, an epidemiologist may be able to modify the disease model as they see fit, a public health policy maker may be able to modify and implement new intervention strategies with relative ease. Second, is to propose an efficient indexing for uncertain spatio-temporal data to facilitate generation of individual level contact data.

The objectives of this thesis are as follows:

- To extend previous work on contagion model to predict risk at contact venues utilising pre-configured road networks.
- Test model(s) with real-world road networks.
- Visualise contagion spread in virtual community.
- Incorporate parameterised contagion intervention strategies.
- Propose efficient spatio-temporal indexing of real-world mobility data.

1.4 Significance of this thesis

The results and outcomes from this thesis are already being incorporated by the partner organisation for their contagion simulation layer of their existing simulation platform. This provides their clients with the ability to simulate relevant "what-if" scenarios and facilitates conversation between public health policy makers and general public.

1.5 Research and development methodology

Our methodology is broadly divided in four phases as follows:

1. **Literature Survey:** We reviewed and analysed existing approaches for modelling of transmission of communicable diseases.
2. **Design and Implementation:** We proposed a framework to extend existing contagion models for incorporating contact through uncertain mobility patterns, intervention strategies and contagion mutations. We implemented instances of it, which is incorporated by the partner organisation.

3. **Datasets:** We have simulated disease propagation on road-network and virtual landmark datasets for small communities in New Brunswick, Canada.

4. **Software Testing and Model Validation:** We have tested simulation outcomes with observed data and cross-validated with independently built instances of the simulator following the proposed framework.

1.6 Techniques implemented as part of this thesis

In this section we list down two technical implementations produced as solutions to the research questions.

1.6.1 Estimation of contagion risk based on simulation on emergent social network

- Utilisation of the mobility simulation of our partner organisation to obtain synthetic, emergent contact networks for individual level agents for a specific geography and mobility pattern.

- Based on the contagion modelling framework, we posited possible contagion models and intervention strategies and estimated their associated risk.

1.6.2 Facilitate efficient generation of individual contact network from massive uncertain mobility trajectory datasets

- Proposed utilisation of a metric index for indexing non-metric uncertain trajectory segments.

- Extended metric-index to incorporate uncertain trajectory segments for doubly-range search. Implemented efficient strategy for parallel query processing on proposed extension.

1.7 Main contributions of this thesis

The main contributions of this thesis are as follows:

- Introduction of an individual level network based stochastic simulation framework for contagion dynamics.
- Extension of contagion model and implementation of such, as instances of said framework in both declarative and purely functional paradigm.
- Extension of metric index to accommodate non-metric uncertain trajectory encoding.
- Implementation of parallel spatio-temporal index for efficient uncertain trajectory range search.

1.8 Summary

In this chapter we outlined the motivation behind providing decision support to public health administrators and specified two lines of research that compliment each other. First, an individual level contagion modelling framework capable of answering present and future public health questions. Second, an efficient indexing of uncertain mobility data to inform the modelling efforts with individual level contact data. The rest of the thesis is organised as follows. In Chapter 2 we review the relevant background and related work on the individual level contagion modelling framework and spatio-temporal indexing. In Chapter 3 we analyse our thesis problem statement and list specific research activities. In Chapter 4 we describe the individual level contagion modelling and simulation framework, its implementation and outputs. In Chapter 5 we propose an efficient spatio-temporal index to facilitate incorporating uncertain real-world mobility data into our simulation framework. Finally we draw conclusions and outline future research in Chapter 6.

Chapter 2

Background and Related Work

In this chapter we review background literature and related works on agent-based modelling in Section 2.1, contagion modelling as compartments in Section 2.2, observations of Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) in Section 2.3 and spatio-temporal indexing of uncertain trajectory data in Section 2.5.

2.1 Agent-Based Modelling

Macal et al. [61], states that agent-based modelling and simulation (ABMS) is a 'ground-up' approach to model a complex system as a heterogeneous composition of interacting entities. With this approach, it is possible to create conditions for emergent complex patterns not explicitly programmed, arising from agent interactions based on simplistic rules.

2.1.1 Utility of ABMS

As a result, their proposal differentiates ABMS from traditional ODE based modelling, as stated by Railsback et al. [81]:

> "With computer simulation, the limitation of mathematical tractability

is removed so we can start addressing problems that require models that are less simplified and include more characteristics of the real systems."

Therefore, the resultant ABMS system offers the modeller to answer both of the following questions to some degree:

- what happens to the system because of what its individuals do, and
- what happens to the individuals because of what the system does.

2.1.2 ABMS application domains

Macal et al. [61] further refer to broad areas of application of this technique along with their exemplar citations spanning the following simulation contexts:

- Arthur et al. [4] provides a collection of analyses of economic processes of the United States stock market as interaction of heterogeneous agents with adaptive behaviour and strategies.
- Macal et al. [60], analysed the role of trust in context of supply chain networks between businesses by modelling traders as decision making agents.
- Folcik et al. [32], analysed emergent human innate and adaptive immune responses against a pathogen. They simulated basic cell types, mediators and antibodies as agents with two tissue environment and the lymphatic circulation system.
- North et al. [68], proposed a complimentary agent-based method for analysis for holistic patterns of consumer markets. Their proposal included a system with decision making agents including customers, retailers with different objectives and observation of system level characteristics to solve for business problems posed by Procter and Gamble.

- Kohler et al. [54], provides a collection of articles on utilising agent-based systems to re-create emergent historical phenomenon. This includes migration patterns of Hominins, Resource distribution patterns in a hunter-gatherer society in context to foraging strategies, Demographic decline of Celtic agglomerations in central Europe.

- Moffat et al. [66], proposed a map-aware non-uniform cellular-automata based system that produces emergent behaviour of armies on a simulated real-world battlefield. Their agents were collection of individuals a.k.a companies with decision making behaviours for engagement with opposing forces and mobility.

- Hill et al. [41], extended the searched theory in context to warfare with their proposed multi-agent simulation based on the Allied search for U-boats in the Bay of Biscay during WWII. They validate their approach and utilised to emperically examine modern search patterns.

- Bagni et al. [6], modelled the spread of Bovine Leukemia both as systems of ODE and agent-based approach. Their agents were cows in context to a farm. They had modelled a typical life-cycle of a cow, complex contagion process, diagnostic test based case discovery and bovine movement between insulated groups of cows called sectors.

2.1.3 Structure of ABMS

They further list three typical elements of an agent-based model:

1. A set of *agents*, their attributes and behaviours.

2. A set of agent *relationships* and methods of interaction: An underlying topology of connectedness defines how and with whom the agents interact.

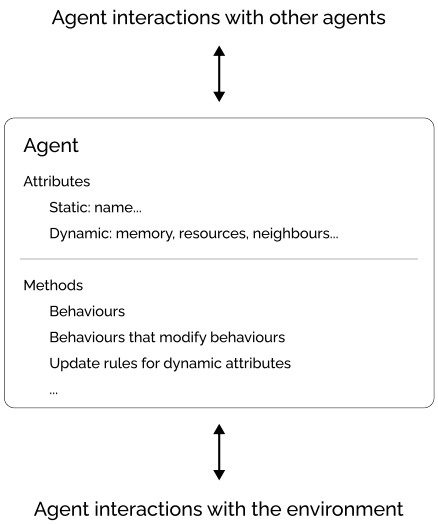


Figure 2.1: Structure of a typical agent in Macal et al. [61]

3. The agents' *environment*: Agents interact with their environment in addition to other agents.

2.1.4 Definition of an agent

Although they provide us with the structure of a typical agent as shown in Figure 2.1, they further state that a consensus for the definition of an agent is yet to be formed. We adopt a broad definition coined by Dorri et al. [28] as a guiding principle, which states the following:

> *Agent: An entity which is placed in an environment and senses different parameters that are used to make a decision based on the goal of the entity. The entity performs the necessary actions on the environment based on this decision.*

2.1.5 Individual level agent-based modelling

Railsback et al. [81] further centres the definition of ABMs as follows:

Definition 2.1.1. *ABMs are thus models where individuals or agents are described as unique and autonomous entities that usually interact with each other and their environment locally. Agents may be organisms, humans, businesses, institutions, and any other entity that pursues a certain goal.*

For example, they provide a list of motivating scenarios and corresponding literature, out of which, four are listed in the Table 2.1.

Table 2.1: Example ABMS problem statements

Problem statement	**Agents**	**Environment**
What factors caused the housing market collapse of 2008, and what policy changes could prevent a recurrence, Gilbert et al [34]; Geanakoplos et al. [33]	Realtors, Buyers and Sellers	Parameterised Synthetic Geospatial Cellular Grid
How do shorebird populations respond to loss or alteration of the mudflats they feed in, and how can the effects be mitigated cost effectively, Goss-Custard et al. [36]	Bird (Oyestercatcher)	Parameterised Synthetic Geospatial Cellular Grid
How is development and maintenance of human tissue regulated by signals from the genome and the extracellular environment and by cellular behaviors such as migration, proliferation, differentiation, and cell death? Peirce et al. [75]	Endothelial cell, Smooth muscle cells, Interstitial precursor cell, Perivascular cell and Platelet derived growth factor - BB	Parameterised Synthetic Microscopic Cellular Grid
What drives patterns of land use change during urban sprawl, and how are they affected by the physical environment and by management policies Brown et al. [13]	Residents and Service Centres	Parameterised Synthetic Geospatial Cellular Grid

Please note, that following Definition 2.1.1, in context to contagion spread modelling it is possible to define agents as collections of humans. For example an agent may be defined as a household with a goal of minimising health risk posed by a contagion

while maximising economic and social well being. However for the scope of this thesis we will focus our attention on ABMS with agents as individual humans. For the sake of brevity, will be referring to them as Individual level ABMS (IL-ABMS).

2.1.6 IL-ABMS for contagion modelling

A complete agent based approach to contagion simulation for a population requires both the simulation of movement and the simulation of disease progression. This approach, in theory provides most detailed construction of a synthetic double for a population, thereby providing a less expensive alternative than real world experiments for mitigation strategies. However, this comes with a trade-off of incurring much higher computational cost than that of their ODE counterparts. This cost is further exacerbated by the following considerations of an ongoing pandemic:

- Rapidly changing context of observable data of disease surveillance which requires modellers to keep up, increasing model creation cost.
- Reduction in no. of possible simulations of mitigation strategies thereby reducing the exploration of solution space.
- Both of the above, along with estimation of downstream cost of execution of policy increases missed opportunity cost for policymakers.

Based on the literature surveyed, three prominent strategies to preserve the advantages of a ABMS are listed below:

1. Utilise an ABMS designing tool or framework which reduces model creation and computational cost by simplifying the underlying complexity. For example Hunter et al. [44] utilises NetLogo [95], that disregards urban mobility topology of road networks and approximates human movements as straight lines.

2. Not compromising on by simplification of mobility or available data but utilising highly scalable and detailed tool or framework which decreases simulation computation cost. However this incurs infrastructure cost for high performance computing and model creation and rapid iteration cost to keep up with changing context. For example, Stroud et al. [86] created EpisimS which is capable of simulating influenza outbreak for 19 million individuals, residing in 6 million households and with their daily activities spread across 2.3 million locations in an overnight batch. The experiments were performed on a 1290 node cluster with 10.2 TB of RAM. Furthermore, their source code of said tool is 31 KLOC.

3. Utilise ABMS to generate detailed mobility and activity patterns of a population and infer the contact network from the simulation output. Simulate the the contagion dynamics on the basis of the contact network. This approach subdivides the contagion simulation problem by taking the output of an IL-ABMS of human mobility as an input for a network based contagion modelling system. This reduces the computational complexity of disease transmission to graph traversal and stochastic graph colouring algorithms. This approach also effectively removes the cost of mobility simulation in the long run. For example, Manout et al. [63] utilise mobility simulation outputs from Multi-Agent Transport Simulation (MATSIM) [43] to generate the contact network of a population to simulate a network based contagion model.

For the scope of this thesis, we chose the 3rd strategy for our contagion simulation framework.

2.1.7 Human mobility modelling

To model a contagion spread in humans, we must be able to extract spatio-temporal overlaps for a given population. Utilising available geo-location data from location

based services introduces challenges of uncertainty and scale that we review in Section 2.5.

An alternative to real-world mobility data is utilising individual level mobility simulators to generate synthetic mobility data.

2.1.7.1 Citisketch™ - The Black Arcs [46]

Our industrial research partner organisation, the Black Arcs provided us with the access to Citisketch™, a Javascript simulator based on PixiJS [20] that generates synthetic mobility data of a population at a community level. They utilise open data from OpenStreetMaps [73] for extracting road networks, buildings and land usage information. They also disaggregate demographic and labour data from Statistics Canada [70] to generate a close approximation of expected population and their residences, places for work, education, shopping and recreation utilising the package synthpop [69].

The daily activities and associated travel times, referred as "schedules" of simulated individuals of a household are computed in accordance to an extension of Toronto Area Scheduling Model for Household Agents (TASHA), Miller et al. [65], which in turn is informed by 1996 Transportation Tomorrow Survey data [38]. From these schedules we infer the list of space-time overlap events between agents, also referred to as the "contact matrix". In the next section we briefly review the TASHA model and how it informs mobility.

2.1.8 Toronto Area Scheduling Model for Household Agents

This is a prototype activity scheduling microsimulation model to generate a typical 24-hour workday or *schedule* for individuals of a household proposed by Miller et al. The model has been subsequently validated by Roorda et. al [83] for the Greater Toronto Area, Canada for the 1996 travel survey data from which the model is derived

from and the comparison with forecast for 2001 travel survey data. This model simplifies the the spatio-temporal complexities of an urban mobility by aggregating them into *zones* and maintaining a *distance matrix* and *travel time matrix* between them. The activity scheduling includes stationary engagement at a location as well as travel and are termed as *activity episodes*. The episodes are incorporated within a concept of *project* which is defined as a set of activities tied to a common goal or outcome. This model considers the following four project types for individuals: *work*, *school*, *shopping* and *other*. Multiple individuals of the same household coordinate to fulfil the requirements for *joint shopping projects*, *joint other projects* and *serve dependent projects* like adults transporting their children to school. The types of projects are based on the categorisation scheme of the purpose of trips reported in the 1996 Transportation Tomorrow Survey [38]. The survey also serves to generate an *activity episode's* probability distributions for frequency, start-time and duration and heuristics to ensure the sequence of activities remain logical.

For selection of activity location, this model expects the usual place for *work* and *school* type activities of an individual as inputs and employs an entropy model for *shopping* and *other* type of activities. The entropy model is parameterised on employment, population, activity density and distance. Post initial activity *agendas* are generated, a rule-based conflict resolution is performed to get final *schedule* for each individual. The household vehicle allocation is performed with a random utility model with simulated normal error terms.

Citisketch™'s mobility model is an extension of TASHA and for a given geography and population data, the software outputs origin-destination location pair of journeys of each individual with departure and arrival time.

2.2 Infectious disease modelling as compartments

Keeling et al. [49], states in these models, individuals of a population are put into compartments in relation to the contagious disease which are denoted by labels like *Susceptible*: individuals who may get infected, *Infectious*: individuals who may infect susceptible individuals, *Latent*: an individual who is infected but not yet infectious, *Removed*: Individuals who are not participating in the propagation of disease due to immunity or death and so on. At any given time, a person may be in exactly one of the compartments. One of the simplest of contagion models comprises of three compartments, namely Susceptible, Infectious and Recovered and is called the SIR model.

2.2.1 SIR model as systems of ODEs

Keeling et al. [49] further states, that simplifying the problem space by, neglecting the details of progression of the disease through the population and behavioural heterogeneity of each individuals have been successful historically in modelling contagions and provides the foundation of mathematical epidemiology as the susceptible-infectious-recovered (SIR) model and susceptible-infectious-susceptible (SIS) model. The system of ODEs for SIR are as follows:

$$\frac{dS}{dt} = bN - \lambda S - dS, \tag{2.1a}$$

$$\frac{dI}{dt} = \lambda S - gI - dI, \tag{2.1b}$$

$$\frac{dR}{dt} = gI - dR, \tag{2.1c}$$

Here, λ, a.k.a the force of infection with respect to transmission rate of the infection is calculated as follows:

Table 2.2: List of notations of SIR-ODE, Equation 2.1

Notation(s)	**Description**
S, I and R	Number of Susceptible, Infectious and Recovered individuals respectively in a population size of N
b	Birth rate
d	Natural death rate
g	Rate of recovery from infection
λ	Rate at which susceptible individuals are infected, a.k.a "Force of infection"

$$\begin{aligned} \lambda &= \text{transmission rate} \\ &\times \text{effective number of contacts per unit time} \\ &\times \text{proportion of contacts infectious} \\ &\approx \tau \times \hat{n} \times \frac{I}{N} = \beta \frac{I}{N} \end{aligned} \tag{2.2}$$

This leads to a non-linear term, $\beta SI/N$, which represents the transmission of infection. Here, N is the size of the population and β is the model parameter.

For a reasonable period of time, i.e., for a fraction of mean lifetime of the individuals and for a population closed from exogenous mobility, we may simplify this equation further with the birth rate and natural death rate to be 0.

2.2.2 Parameterised β as social network

Keeling et al. [49] further states, that the heterogeneity of different levels of mixing of sub-population may be represented as a matrix of transmission parameters, β. Also the number of contacts of an individual is considerably smaller than the population size and introduces the concept of 'adjacency matrix' or 'sociomatrix' $\mathbf{A}$, where $\mathbf{A}_{ij} = 1$, if there is a connection for individual i may infect individual j or otherwise,

$\mathbf{A}_{ij} = 0$.

Exploring the infection propagation on the basis of idealised network structures and they posit that attempting to predict population-level dynamics from individual-level observations, it is paramount that the network structure is taken into account. As possible extensions of their work, they propose future individual level modelling of contagion spread to allow for more complex interventions strategies and incorporation of emergent networks.

2.2.3 Outline of compartmental models

Brauer [11] traces the history and present topics of interest in mathematical epidemiology and lists the following developments in compartmental model:

1. **Infection age epidemic models**: These models allow for configurable length of stays at each compartment and sequence of infective compartments. These models may include compartments for treatments and isolation as well.

2. **Endemic disease models**: These models focus on the possible final states of an epidemic and their stability including considerations for multiple strains, transmissions from mother to offspring and waning immunity periods.

3. **Disease transmission by vectors**: These models are useful when diseases are transmitted between humans occurs indirectly, mediated by a vector organism. For example, a well known vector is mosquito that transfers blood between humans causing the transmission of infectious diseases including malaria, dengue fever and Zika virus. These models usually maintain separate compartments for hosts and vector species.

4. **Heterogeneity of mixing**: These models divide the population into subgroups of interest for which the dynamics of the infectious disease may be different. For example, the age of an individual of a population may have

different chances for reproduction or survival. Also, diseases often exhibit different infection rates and mortality rates for different age groups.

5. **The next generation method**: This is a generalised approach to yield reproduction number for each combination of possible infection caused by an individual from any of the infectious compartments to any of the susceptible compartments.

Brauer [11] further states that at the beginning of an outbreak, instead of a simplistic compartment model of deterministic solutions, treating infection as a branching stochastic process incorporates the patterns of contact between members of population. This pattern may be modelled with assumptions on variation of degree of distribution of vertices signifying individuals of a population with edges signifying contacts. Alternatively, we may model the complete contact network.

2.2.4 Simulation of contagion as a social experiment

Klepac et al. [52] explore generation of emergent social network with respect to voluntary participation of individuals in a citizen science experiment from a small town named Haslemere in south west Surrey, England. In one part of this experiment the volunteers self-reported their location via a smartphone app for 3 days for which the data was made available in a subsequent publication, Kissler et al. [51]. The data-collection, formalisation of "social-contact" network for subsequent disease modelling is summarised in the Figure 2.2.

This process proved to be a fundamental building block for SARS-CoV-2 disease modelling as we will review in Section 2.3.

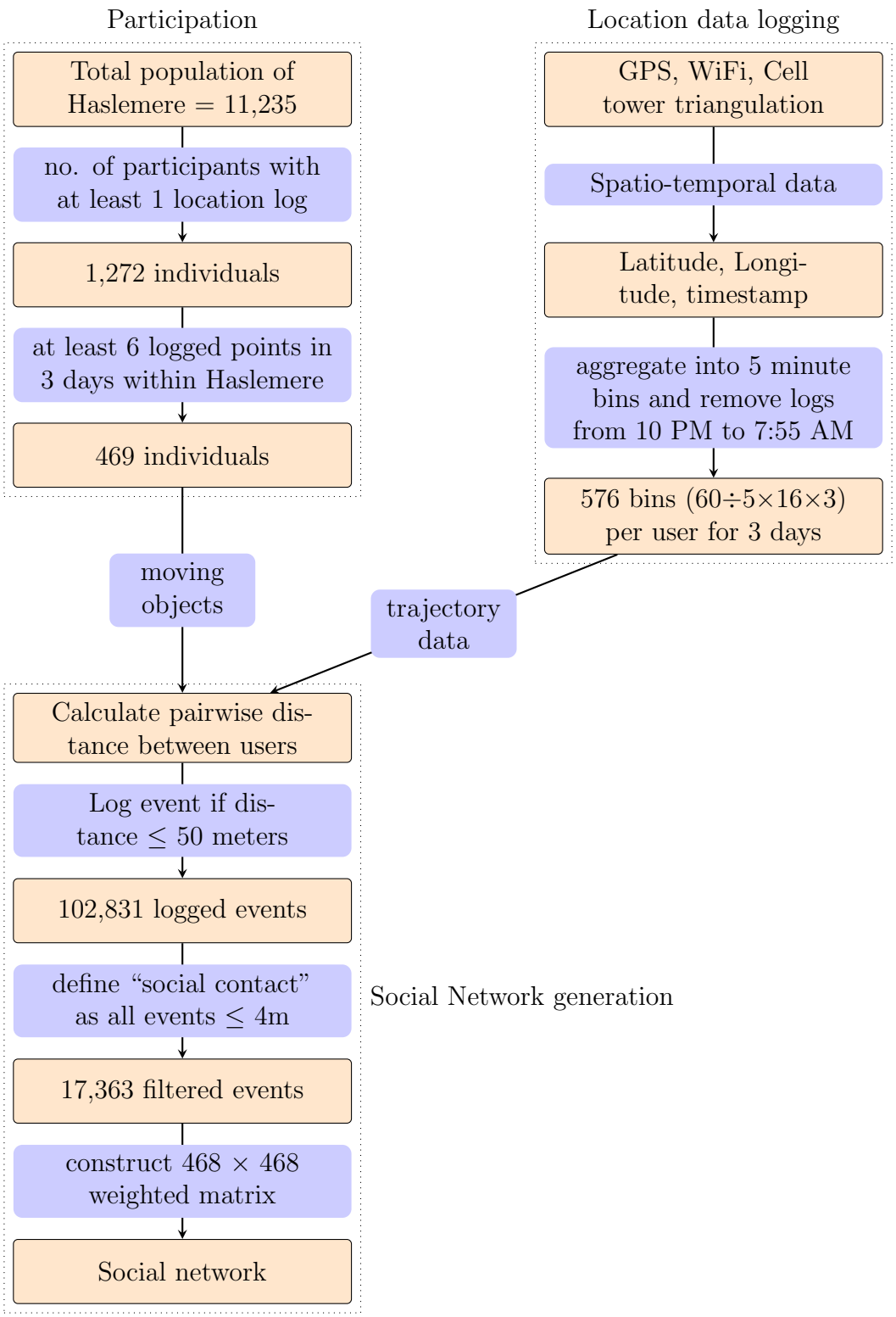


Figure 2.2: Haslemere experiment workflow

2.3 SARS-CoV-2

Firth et al. [29] utilised the outcomes from the Haslemere experiment, Kissler et. al [51] and quantified 3 days of social contact network edges to an ordinal scale as follows:

- **0**: corresponding to **no** contact,
- **1**: corresponding to **weak** contact,
- **2**: corresponding to **moderate** contact,
- **3**: corresponding to **strong** contact

This transformation enabled simulation of a dynamic disease model over a reasonable period of time that goes beyond 3 days.

They defined the transmission rate between a pairwise individuals, λ' with equation 2.3.

$$\lambda'(t, s_i, p_i) = A_{s_i} I_{ei} \int_{t-1}^{t} f(u; \mu_i, \alpha_{p_i}, \omega_{p_i})\, du \tag{2.3}$$

Table 2.3: List of notations of Transmission Rate Equation 2.3

Notation(s)	**Description**
t	Number of days since infector i was exposed
s_i	Infector's symptom status, boolean
p_i	Infector's pre-symptomatic status, boolean
A_{s_i}	Scaling factor of infector's symptomatic status
I_{ei}	Weighting of the edge in the contact network
μ_i	Infector's onset time as location parameter
α_{p_i}	Slant parameter for skewed normal distribution
ω_{p_i}	Skew parameter for skewed normal distribution
$f(u; \mu_i, \alpha_{p_i}, \omega_{p_i})$	Probability density function w.r.t generation time.

Further, they modelled the probability of infection for a Susceptible-Infectious pair of individuals t days after the infector's exposure time as equation 2.4.

$$P(t, s_i, p_i) = 1 - e^{-\lambda'(t, s_i, p_i)} \tag{2.4}$$

This frame of modelling enables capturing the population level dynamics from individual level models of disease transmission. And they could reasonably model diagnostic testing, quarantining, physical distancing and the likes with different constructions of equivalent social contact networks. The parameter values for their their epidemic model are listed in Table 2.4.

2.3.1 Diagnostic testing

In a recent study by Dinnes et al. [26], an analysis of two types of commercially available, rapid point-of-care tests: antigen and molecular tests' accuracy with respect to the standard Reverse transcription polymerase chain reaction (RT-PCR) test for Covid-19 is performed. They reviewed of 64 studies, containing 24,087 samples, investigating 165 different antigen tests and 5 different molecular tests.
A portion of their finding is shown in Table 2.5.

Trevethan et al. [88] defines *Sensitivity* and *Specificity* by equation 2.5 and 2.6 respectively in terms to the confusion matrix in Table 2.6.

$$Sensitivity = [a/(a + c)] \times 100 \tag{2.5}$$

$$Specificity = [d/(b + d)] \times 100 \tag{2.6}$$

Table 2.4: Parameter set of network contagion model by Firth et al. [29]

Parameter(s)	**Description**	**Notes**
Sampled		
Incubation period (mean and s.d.)	5.8 days (2.6)	
Serial Interval	Location = incubation period For post-symptomatic transmission, slant = ∞, scale = 2; For presymptomatic transmission, slant = -∞, scale = incubation period	
Delay from onset/tracing to isolation, and from isolation to testing (median and 5th-95th percentiles)	0.9 days (0.3-3.9) days ('short'); 3.5 days (0.7-8.2) days ('medium')	
Fixed		
Initial cases	1, 5	Assumed
Scaling parameter (and corresponding empirical estimate of the reproduction number R0)	0.5 (2), 0.8 (2.8), 2 (3.5)	
Percentage asymptomatic individuals	20%, 40%	
Infectiousness of asymptomatic individuals	50% (relative, to symptomatic)	Assumed
Percentage individuals infectious pre-onset	20%, 40%	
Outside infection rate	0.0001, 0.001, 0.005,0.01	Assumed
Percentage of contacts traced	30%, 60%, 90%	Assumed
Maximum number of tests	0, 5, 25, 50	Tested
Test false positive rate	0.02	
Test false negative rate	0.1	

Table 2.5: Aggregated diagnostic test accuracy by Dinnes et al. [26]

	Evaluation (studies)	**Samples (SARS-CoV-2 cases)**	**Sensitivity (95% CI) [Range]**	**Specificity (95% CI) [Range]**
Antigen-tests				
Symptomatic	37(27)	15,530(4410)	72.0 (63.7 to 79.0) [0% to 100%]	99.5 (98.5 to 99.8) [8% to 100%]
Symptomatic (up to 7 days from onset of symptoms)	26(21)	2320(2320)	78.3 (71.1 to 84.1) [15% to 95%]	-
Asymptomatic	12(10)	1581(295)	58.1 (40.2 to 74.1) [29% to 85%]	98.9 (93.6 to 99.8) [14% to 100%]
Rapid molecular tests				
-	29(26)	4351(1787)	95.1 (90.5 to 97.6) [57% to 100%]	98.8 (98.3 to 99.2) [92% to 100%]

Table 2.6: Diagnostic test confusion matrix by Trevethan et al. [88]

		True diagnosis		
		Positive	Negative	Total
Screening test	Positive	a	b	$a+b$
	Negative	c	d	$c+d$
	Total	$a+c$	$b+d$	N

2.3.2 Contact tracing

Kretzschmar et al. [55] modelled the mechanics behind different strategies of contact tracing and voluntary isolation and their impact on the rate of transmission of infection. They concluded that an app-based (or same day multi-level traced-testing) has the potential to prevent up to 80% infection transmissions. Their model parameters are listed in Table 2.7.

Table 2.7: Contact tracing model parameters by Kretzschmar et al. [55]

	Isolation	**Conventional contact tracing**	**Mobile app contact tracing**
Testing coverage	80%	80%	80%, 20%, 40%, 60%, 100%
Testing delay (D_1), assuming immediate isolation when testing positive	4 days	4 days	0 days
Time to trace close contacts (D_2)	-	3 days	0 days
Time to trace other contacts, assuming testing and isolation of those who test positive	-	3 days	0 days
Tracing coverage of close contacts	-	80%	80%, 20%, 40%, 60%, 100%
Tracing coverage of casual contacts	-	50%	80%, 20%, 40%, 60%, 100%
Time traced back	-	7 days	7 days

2.3.3 Effects of vaccination

The research on effectiveness of vaccines, as reported by Pritchard et al. [78], Makhoul et al. [62], Layan et al. [59] and Bian et al. [10], confirms that vaccinations reduce the risk of transmission of infection, reduce chances of severe symptoms from SARS-CoV-2 and reduce the duration of infected period. Glasser et al. [35]

states, vaccination strategy is affected by behavioural biases. This is an ongoing and rapidly changing research area as new mutations of the SARS-CoV-2 contagion are appearing, new vaccines are being developed and observational data for vaccinated individuals from diverse socio-economic and geo-spatial context are being gathered, by Killingley et al. [50], Klomjit et al. [53] and Puhach et al. [79].

2.3.4 Effect of other non-pharmaceutical interventions

Abueg et al. [1], Hens et al. [40], Lai et al. [57], Cheng et al. [16] and de Souza Santos et al. [24], state that along with conventional contact tracing, other non-pharmaceutical interventions (NPIs) including *community masking*, *physical distancing*, *social bubbles*, *closures* or *lockdowns* have been effective in reducing the spread of infection during this pandemic.

2.4 Related work on contagion modelling

Contagion modelling of the spread of SARS-CoV-2 viral strains is a rapidly evolving domain of epidemiology research inviting collaborations from researchers from multiple domains including immunology, maths, physics, computer science, behavioural psychology and more, in pursuit of informing public health. In this section we cite a small subset of the available research and techniques. Rambhatla et al. [82] model the infection chances of an individual as a Hawkes Process, extending the model of Zeighami et al. [97] to incorporate their spatial stochastic nature and have modelled contagion dynamics with Susceptible-Latent-Infectious-Recovered (SLIR) like compartments as referenced by Kuddus et al. [56], at national, provincial and city level with real-world mobility data. However, the minimum duration of time-overlap and maximum distance for an infection to occur was chosen to be 1 hour and 11 meters respectively. Chiang et al. [17] also encode infection chance as a Hawkes Process

to fit their model to observed mobility and contagion data for forecasting (without explicitly defining the contagion model). Anastasiou et al. [3] have proposed a predictive route optimisation system "ASTRO" that takes into account risk of infection through outdoor and indoor spaces with respect to mobility and spatio-temporal congestion.

Manout et al. [63] utilise MATSIM [43] ABM with a modified SLIR compartmental model as shown in Figure 2.3 to simulate disease dynamics for the city of Montreal. They focus on answering public health questions along the lines of efficacy of business closures, socio-economic disparities of infection spread and more with respect synthetic population mobility composed of multiple daily activities according to expected age and occupation derived from census data. Both mobility and contagion simulator systems are implemented in Java.

Flaxman et al. [31] analysed the effect of combination of non-pharmaceutical interventions (NPIs) and their timing for select European countries to number of deaths and compared them to a counterfactual model of unmitigated pandemic to estimate the measure of risk mitigation. The NPIs modelled are as follows:

- Case based social isolation mandated
- Social distance encouraged
- Public event banned
- School closure ordered
- Lockdown ordered

Both forecasting and counterfactual models were coded in R.

Hunter et al. [44] proposed a data driven approach to model contagion outbreak. First, utilising publicly available summary population statistics, with a granularity of a "small area" which is an Irish administrative division of 50 to 200 households. They

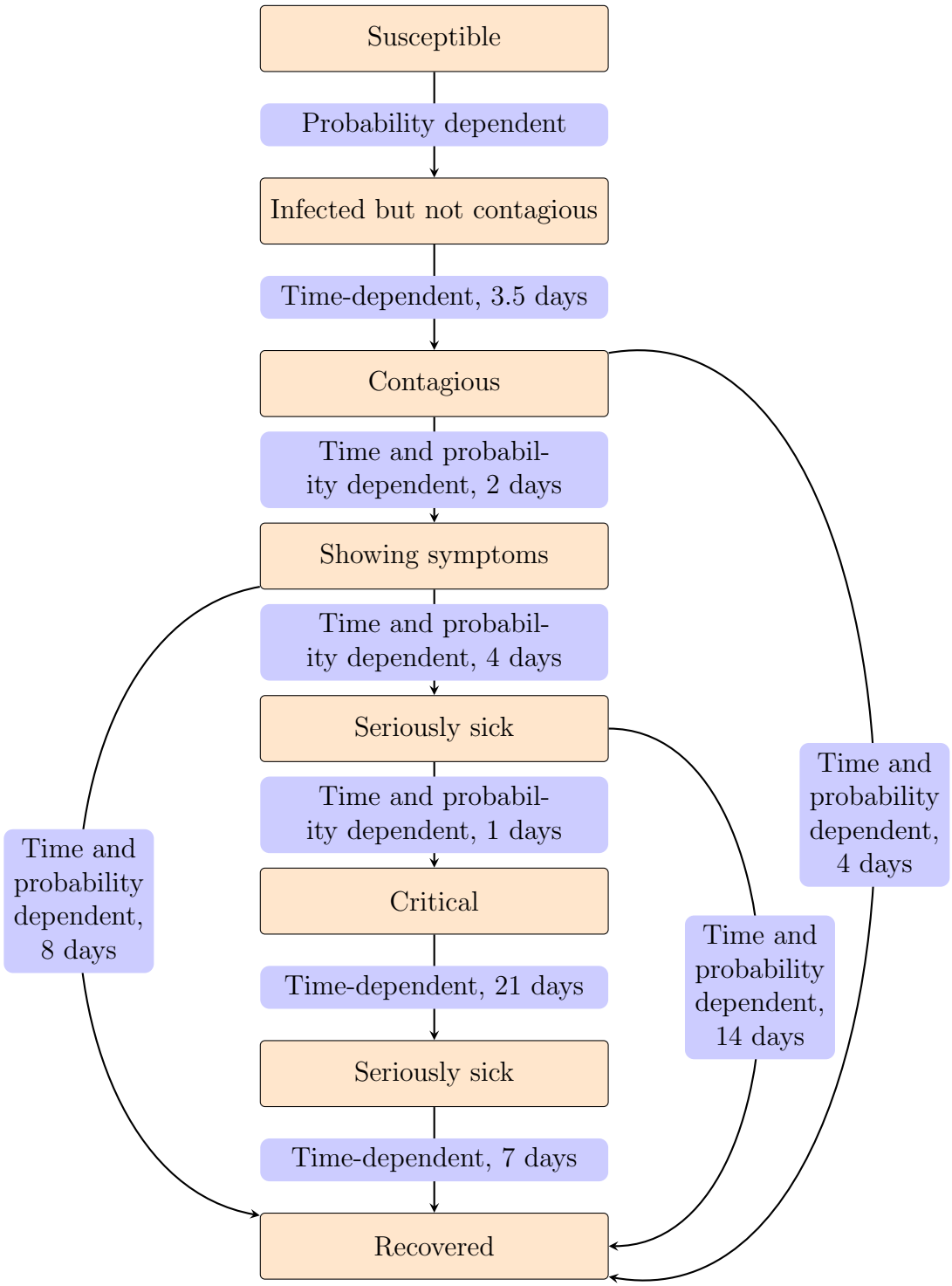


Figure 2.3: Compartmental model of Episim in Manout et al. [63]

dis-aggregated the data to construct individual agents with the same distributions of age, gender, occupation and vaccination status. Second, they utilised land-use and zoning data from public GIS to assign these individuals to households, schools and workplaces. All agents were positioned in a virtual map of the town with a spatial resolution of $111m^2$ and a temporal resolution of 2 hours. Each tick of the simulation was 2 hours and a day was represented as 12 ticks. Part of agents' movements on weekdays were deterministic as a function of their occupation status, such as: being in a school, employed, searching for a job, being a parent or a child and so on. But for all leisure ticks, say time off work or school and weekends, the agents would move randomly through the town. One adaptive behaviour was baked in, it was to have infected agents stochastically deciding to stay home. All movements were simulated as straight lines from origin to destination with ticks of 1. They acknowledge that although this is a naive human mobility model, it suffices for modelling small towns. To simulate a Susceptible-Exposed-Infectious-Recovered (SEIR) infection model, first, a contact between a pair of agents was defined as sharing the same patch or $111m^2$ of virtual area over at least 1 tick of the simulation. When a contact between a susceptible and an infectious agent occurred, a sample was drawn from a uniform random distribution from 0.0 to 1.0. If and only if the sample was less than the probability of infection per contact, the susceptible agent would transition to exposed state. And then eventually to an infectious state after the latency period is over and finally to the recovered state. The transmission probability here is a constant p, which relates to the expected number of individuals to be infected by an infectious individual, R (or the basic reproduction number) as follows:

$$R = c \times p \times d \tag{2.7}$$

Here, c is the number of contacts per unit time and d is the duration of infectiousness. They successfully modelled the 2012 measles outbreak in the town of Schull, Ireland

and stated that percent of outbreaks for a town or small community does not have a strong correlation with the following factors:

- Area or size
- Population
- Population density
- Number of constituent small areas
- Number of students
- Number of not vaccinated individuals
- Probability of transmission per contact

FitzJohn et al. [30] provide a suite of packages in R that may be utilised in tandem with Odin to solve for compartmental models of contagion spread as systems of stochastic processes and infer the parameters based on observational data utilising sequential Monte Carlo techniques. They focus on providing a highly performant simulator to the end user so they may focus in model construction.

Sonabend et al. [85] modelled the effects of easing of NPIs in United Kingdom in light of the emergence of Delta variants of SARS-CoV-2, for which the relative infectiousness, risk of hospitalisation and effect of vaccination was unknown to inform public health of possible near-term outcome scenarios.

Although independent, among the related works, the following two recent works mentioned in Section 2.4.1 and 2.4.2 have a high degree of overlap with the original work done as part of this thesis and its future scope.

2.4.1 Combination of NPIs

Olivera et al. [71] modelled varying effects of public hesitation to vaccination, NPIs with a modified SEIR compartmental model as shown in Figure 2.4 to predict possible

number of deaths at a country level. They extended the disease model Squire package [90] and have the following vaccination parameters for scenario simulation:

- Vaccine mode of action: infection blocking and/or disease blocking.
- Vaccine efficacy.
- Age varying-efficacy.
- Duration of vaccine-acquired immunity.
- Age-dependent targeting and prioritisation strategies.

At the time of writing this thesis age-stratified disease progression or vaccination effects have not been incorporated in the simulator.

2.4.2 Composition of models

Baez et al. [5] implemented a tool in Julia that lets users compose complex stock and flow diagrams out of simple ones, which is widely used in epidemiology. They separate the syntax from the semantics of the stock and flow diagram and provide translation to multiple semantics including ODEs. They describe a simplified version of a COVID-19 model used in Canada, as shown in Figure 2.5.
The constituent models in the Figure 2.5 are as follows:

- **Model A** : Model of natural history of infection, pathogen transmission and hospitalisation.
- **Model B** : Model of vaccination.
- **Model C** : Model of natural history of infection among asymptomatic or oligosymptomatic individuals.

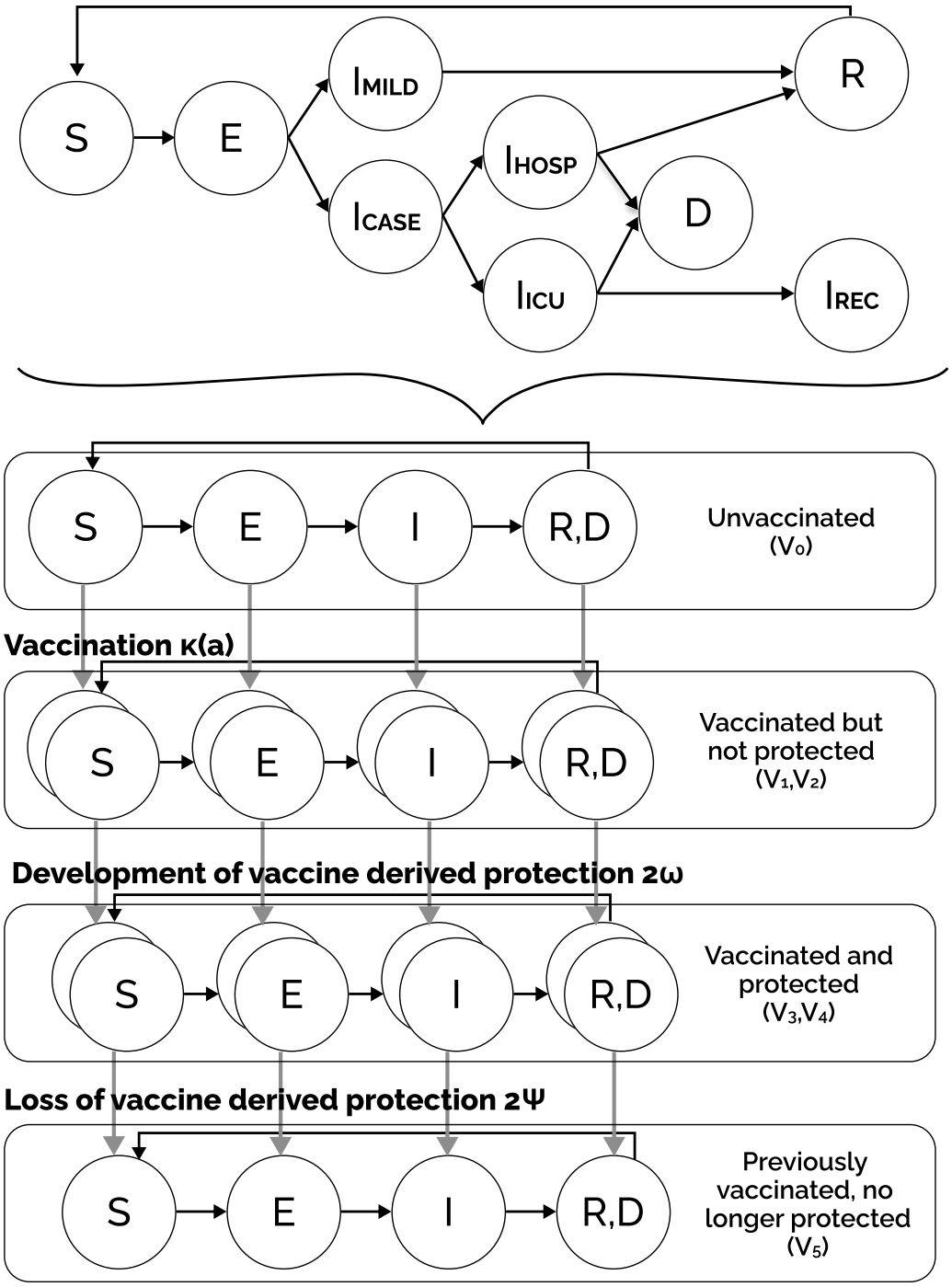


Figure 2.4: Schematic of SARS-CoV-2 transmission model in Olivera et al. [71]

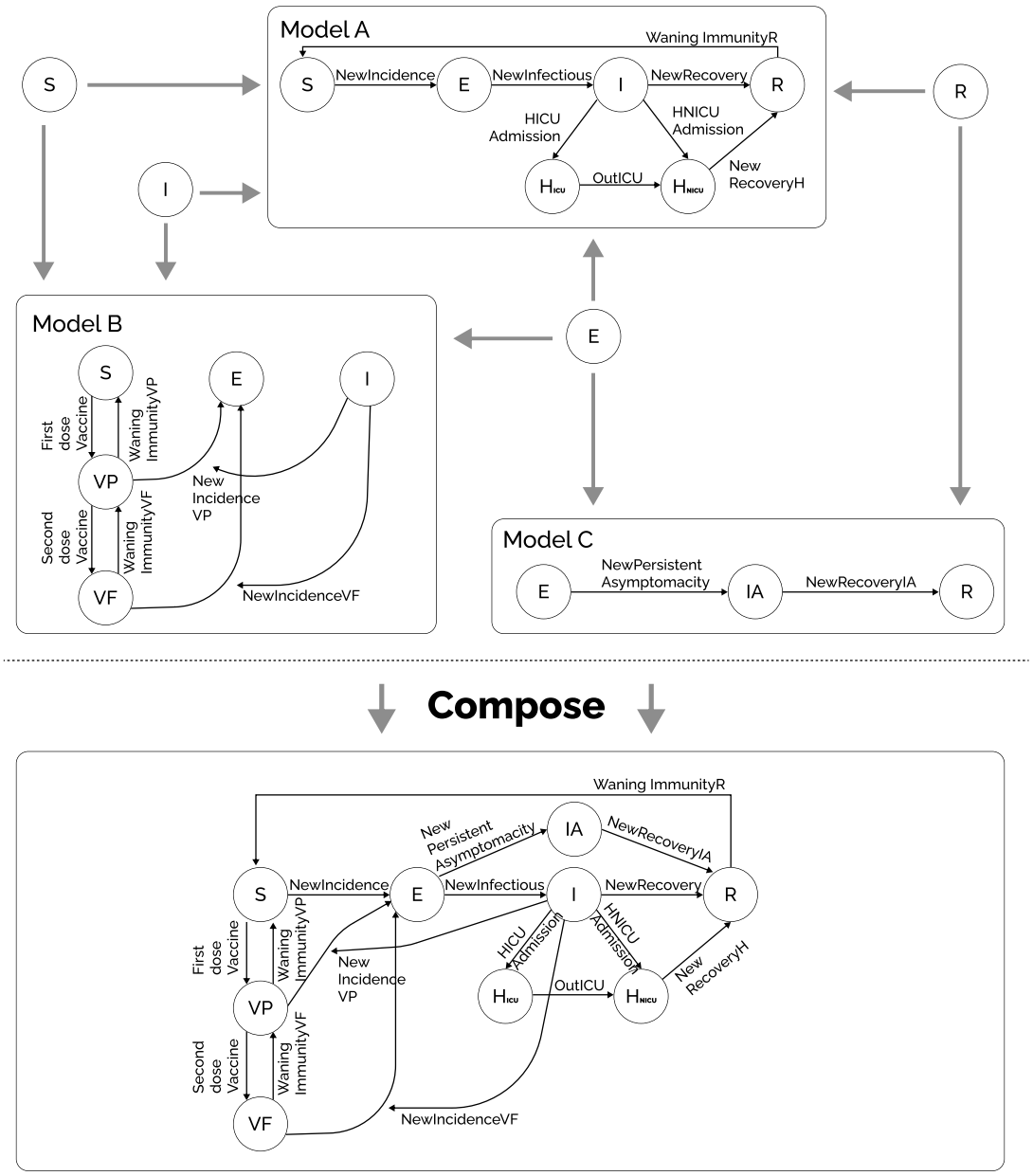


Figure 2.5: Example of composing a COVID-19 model from three smaller models in Baez et al. [5]

We too explore a similar idea of composition of disease progression models instantiated utilising directed graphs. But at the time of writing this thesis, our approach does not have the feature of generating system of ODEs for given directed graph encoding of a disease model and associated augments.

2.5 Real-world mobility data

Modelling contagions like the SARS-CoV-2, require figuring out if any of the moving objects in question had the opportunity to cohabit a space and time long enough to transfer viral load for a possible infection. As an example, Public Health Canada defines 15 cumulative minutes over 24 hours for contact identification and management purposes. Recent works [82, 97, 56, 17, 3] have already shown how mobility data may be utilised for assessment of the risks posed by a contagion. However, this approach has inherent challenges of imprecise nature and very high volume of spatio-temporal data. Cicala et al. [19], note that proximity based contact tracing (PCT) protocols, which work by logging distances between smartphones with PCT apps may estimate the distance with a degree of uncertainty. For one, Bluetooth Low Energy (BLE)-based PCT techniques use the signal strength for estimating the distance and two, the individuals may be carrying their smartphones further from their position, for instance in their luggage instead of in their pockets. We explore encoding schemes for modelling imprecise and uncertain of real-world mobility data in the next Section 2.6.

2.6 Uncertain Trajectories

The simplest representation of a two-dimensional trajectory is a sequence of observations $o_1, o_2, ...o_n$, where each o_i is position data and the associated timestamp: $o_i = (x_i, y_i, t_i)$ as shown in Figure 2.6.

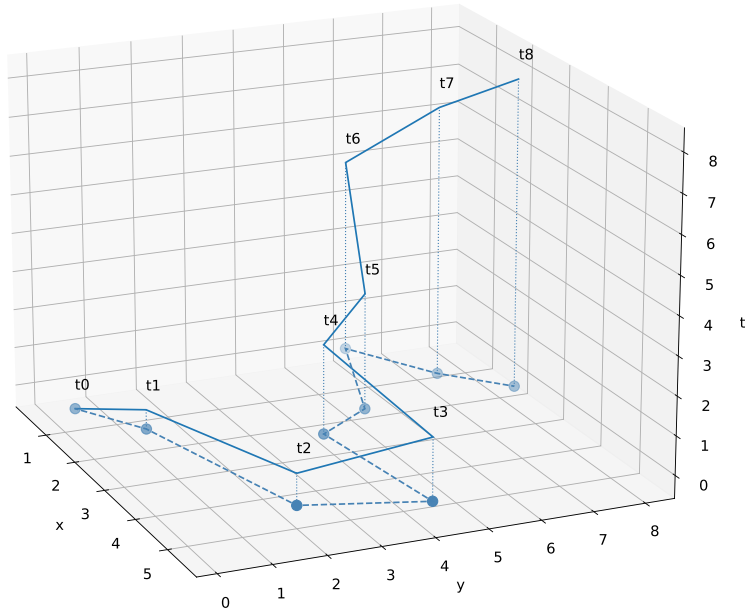


Figure 2.6: Typical Trajectory in Pfoser et al. [77]

To account for inherent imprecision of recorded observation Trajcevski et al. [87] outline two distinct modelling categories:

- **Probability Density Function** (PDF) based encoding the position of a moving object at time t as a two dimensional PDF $l_t : R^2 \rightarrow [0, +\infty)$ and
- **Shape or Geometry** based encoding of a bounding area of a moving object, which may be a rectangle, circle or any arbitrary polygon.

The modelling taxonomy is illustrated in Figure 2.7

2.6.1 Beads and Necklaces

A Bead is a shape-based model for an uncertain trajectory as proposed by Trajcevski et al. [87]. The construction of a Bead relies on a pair of observation (o_i, o_{i+1}) of a moving object m_k and a maximum possible velocity m_k may attain for the particular trajectory segment v^i_{max} dictated by real-world constraints.

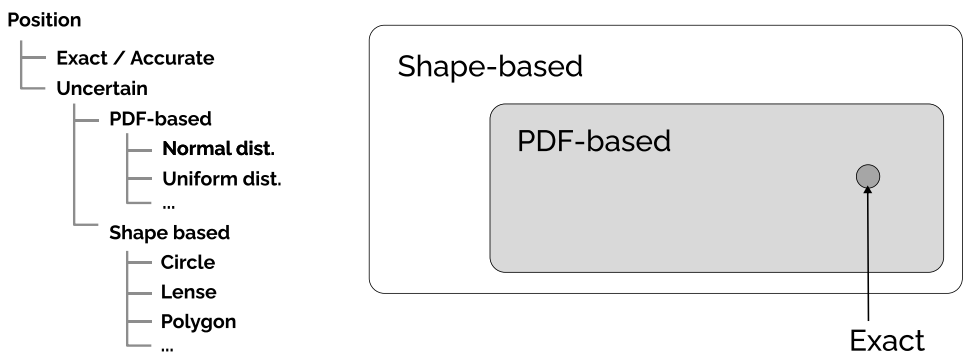


Figure 2.7: Uncertain Trajectory Taxonomy referred to by Trajcevski et al. [87] from Lange et al. [58]

Mathematically, a Bead B_i may be constructed from an observation pair (o_i, o_{i+1}) is an ellipse with the following equation:

$$\frac{(2x - x_i - x_{i+1})^2}{{v^i_{max}}^2(t_{i+1} - ti)^2} + \frac{(2y - y_i - y_{i+1})^2}{{v^i_{max}}^2(t_{i+1} - ti)^2 - (x_{i+1} - x_i)^2 - (y_{i+1} - y_i)^2} = 1 \tag{2.8}$$

A sequence of Beads B_1, B_2, ... B_n created from sequence of observations o_1, o_2, ... o_{n+1} is referred to as a Necklace.

They also recommend the following strategies to approximate Beads and Necklaces with reasonable geometric to balance the conflict between imprecision and encoding and analysis efficiency:

- **Large Bound** (LB): The entirety of a necklace is bounded by a single minimal vertical cylinder.
- **Individual Bead Bound** (IBb): Each Bead B_i is approximated by a circle with their radius r_i as the semi-major axis of the underlying Ellipse. The r_i is dependent on the choice of v^i_{max}, the maximum possible velocity between observations o_i and o_{i+1}.
- **Uniform Bead Bound** (UBb): Similar to IBb, each Bead B_i is approximated by a circle with their radius r_i as the semi-major axis of the underlying Ellipse.

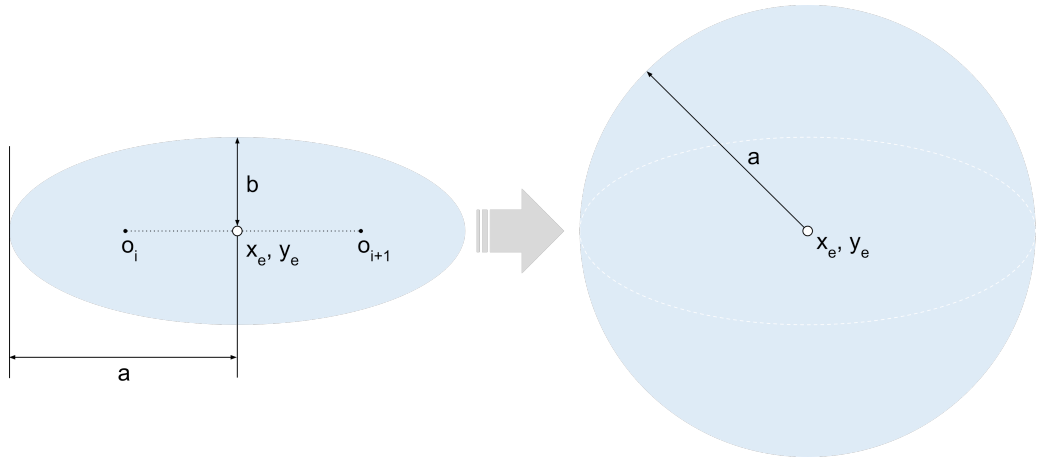


Figure 2.8: Approximating Beads as Circles

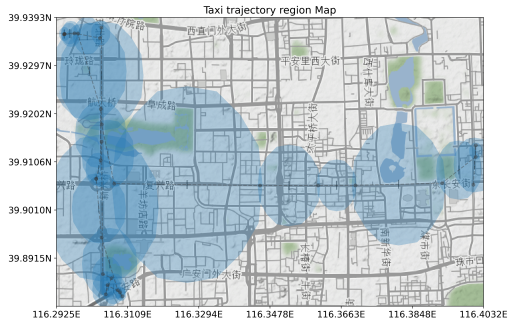


Figure 2.9: Necklace projected on GIS data provided by Yuan et al. [96]

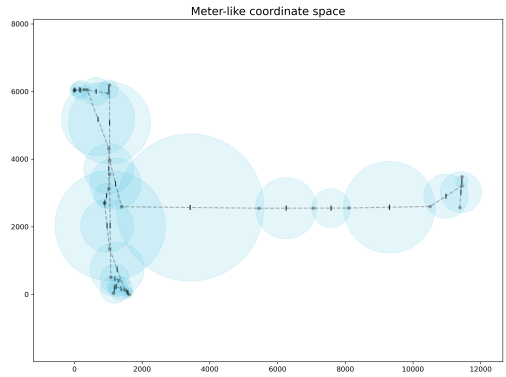
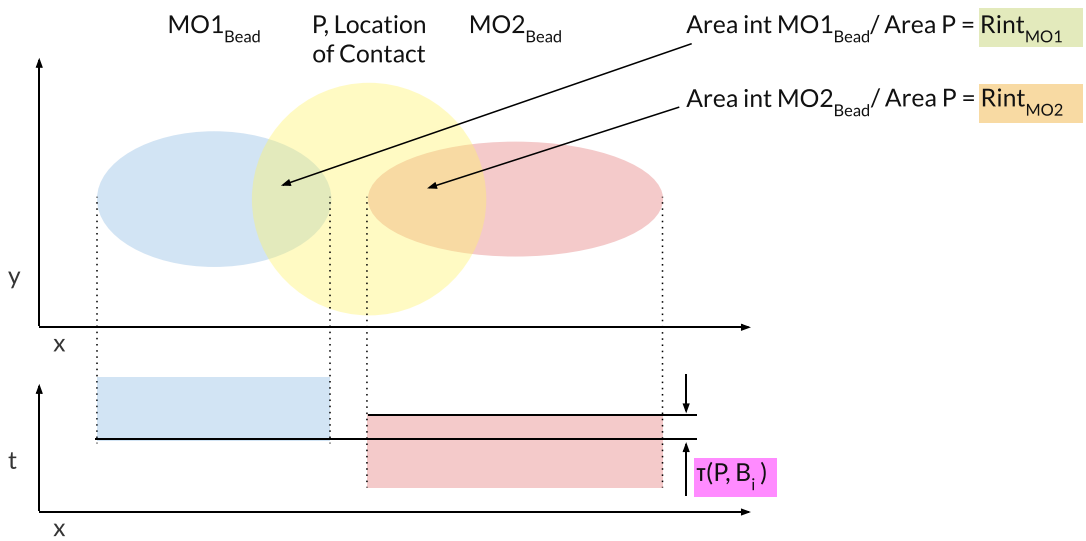


Figure 2.10: Necklace projected on Cartesian plane

But the choice of v^i_{max}, is constant as the maximum possible velocity between any two points o_i and o_{i+1}.

As an example of IBb Bead Necklace based on real-world trajectory data as provided by Yuan et al. [96], please refer to Figures 2.9 and 2.10. Note that, although the beads are circular, they appear as ellipses due to distortions introduced by cartographic projection.



$$Sim_{cont}^{P \cap B_i} = \frac{1 - (1 - R_{int_{MO1}})^{\tau(P,B_i)} + 1 - (1 - R_{int_{MO2}})^{\tau(P,B_i)}}{2}$$

Figure 2.11: Contact similarity for overlap between a virtual landmark and two beads from different moving objects, Zhang et al. [98]

2.7 Measuring contact over uncertain trajectories

Zhang et al. [98] formalise the definition of a Contact Similarity Query (CSQ) for a moving object and a virtual landmark. Given a virtual landmark P with centre $O_c = (x_c, y_c)$ and a radius r_c and any Bead B_i of a moving object, their degree of contact similarity may expressed in terms of area overlap $R_{int} = Area_{int}/Area_P$ and time overlap $\tau(P, B_i) = min\{t_1^e, t_2^e\} - max\{t_1^s, t_2^s\}$, as follows:
$Sim_{cont}^{P \cap B_i} = 1 - (1 - R_{int})^{\tau(P,B_i)}$.
And subsequently they define Contact Similarity measure for the whole uncertain trajectory $B_i \in |\mathbb{B}|$ as follows:

$$Sim_{cont}^{P \cap T} = \frac{\sum_{B_i \in \mathbb{B}} Sim_{cont}^{P \cap B_i}}{|\mathbb{B}|} \tag{2.9}$$

2.7.1 Indexing uncertain trajectories for efficient access

Next, Zhang et al. [98] propose a novel spatio-temporal indexing data structure, namely Uncertain Trajectory M-tree (UTM-tree). This is an extension of a ball partitioning based hierarchical metric index, M-tree, Ciaccia et al. [18], with two significant changes:

- Unlike the classic M-Tree, each node also maintains the overarching time range of their member nodes. This provides a temporal pruning at each Node when executing a CSQ.

- At the leaf level each node of the same trajectory is also maintained as a doubly linked list for efficient extraction of full trajectory once a moving object id is identified.

They evaluate the performance of their proposed index on the Beijing Taxi Dataset [96] outperforming the baseline and Temporal-First (TF) approaches by 6.9× and

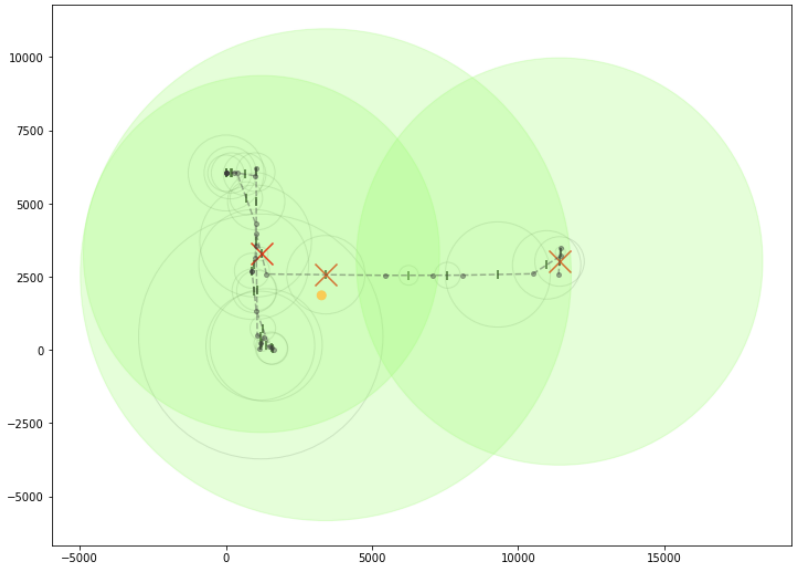


Figure 2.12: 3 UTM Tree indexing 30 beads generated from a real-world taxi trajectory from Yuan et al. [96] demonstrating overlaps of bounding circles of root node of each tree.

3.5× respectively. They further demonstrate that their query running time remains stable and barely change while increasing query parameters of Number of Trajectories, Number of virtual landmarks, Radius of virtual landmarks and Query time interval.

However, the construction of UTM-tree, similar to its inspiration the M-tree is bottom up, i.e. the nodes when accepting a new data-point may split into two, to ensure a height-balanced structure. This proves to be an intrinsic challenge with multiple shuffling of the data-points with respect to the tree structure in a distributed context as described in the following section.

2.8 UTM-tree construction challenges

Zhang et al. [98] describe the construction of a UTM-tree with respect to a bottom-up uncertain trajectory segment insertion algorithm as described in algorithm 1. Here an entry E_i is a composition of a *Bead* B_i with $Ptr(E_p)$ and $Ptr(E_n)$ are

pointers to the next and the previous trajectory segment entry as a doubly linked list. And a UTM-tree node N_j is a composition of $E_j, R_j, t_{min}, t_{max}$ and $ptr(N_j^{sub})$ where, E_j is the "routing object" for the node, R_j is the minimum bounding radius for all entries stored in N_j, t_{min} and t_{max} are the minimum and maximum of all entries stored in N_j and ptr pointing to its child node (N_j^{sub}) if N_j is a non-leaf node. As marked in the algorithm 1 line no. 17, the partitioning of data points for a UTM-tree, like that of a M-tree relies on a bottom-up insertion approach. This step ensures a height-balance and overlap reduction of the nodes of the tree and therefore subsequent spatial indexing efficiency. Therefore, it is prudent to partition the trajectories according to children or sub-trees of a node to ensure efficiency in parallel or distributed context. However, the tree creation process is bottom up, resulting in shuffling of trajectories among partitions making it unsuitable for partition based scaling up as is. Also, UTM-tree, like M-tree, utilises ball-type partitioning that introduces overlaps between sibling nodes (nodes that are at the same depth) with overlaps that may not be effectively mitigated as it is dependent on stochastic selection of routing objects during tree creation. This in turn poses a non-deterministic challenge of selection of partition for further data insertion, which may cause data replication across partitions raising memory cost of indexing to an unsatisfactory degree.

Although, it is possible to partition the trajectory segments before constructing individual UTM-tree indices out of each partition, the resultant data structure would be a hybrid index, for which ensuring overall efficiency would become a multi-part challenge. Alternatively, inspired by the approach of Zhang et al. [98], we evaluate alternative indexing strategies in Section 2.10 after a brief review of metric indexing in Section 2.9.

Algorithm 1 UTM-tree insertion algorithm - Zhang et al. [98]

Input: E_i a new entry, a new tree node N_j
Output: A UTM-tree with E_i added

```
if N_j is not a leaf node then
    for each child nodes N_j^sub of N_j do
        Let R_j^sub denote the radius of N_j^sub, calculate d(E_i, N_j^sub)
        if ∃ at least one N_j^sub such that, d(E_i, N_j^sub) + r_i ≤ R_j^sub then
            Insert(E_i, N_j^sub) where d(E_i, N_j^sub) is minimum;
        else
            Insert(E_i, N_j^sub) where d(E_i, N_j^sub) + r_i − R_j^sub is minimum
        end if
    end for
else
    if N_j is not full then
        Add E_i into N_j, update t_min and t_max;
        if d(E_i, N_j) + r_i > R_j then
            R_j = d(E_i, N_j) + r_i
        end if
    else
        Split(E_i, N_j)                                    ▷ Bottom-up splitting
    end if
end if
return An updated UTM-tree after insterting E_i into N_j
```

2.9 Metric space indexing

Chen et al. [15] states that, metric space is a two-tuple (M, d), in which M is an object domain and d is a distance function that measures "similarity" between objects of M. Furthermore, for any p, q and $o \in M$, the distance function d must have the following four properties:

1. **Symmetry**: $d(q, o) = d(o, q)$
2. **Non-negativity**: $d(q, o) \geq 0$
3. **Identity**: $d(q, o) = 0 \iff q = o$
4. **Triangle inequality**: $d(q, o) \leq d(q, p) + d(p, o)$

Therefore, as a corollary, any data combined with a distance function that satisfies these four conditions constitutes a metric space and is applied to a broad range of data types found in real-world application. In the context of uncertain trajectory segments, metric indices may be utilised to facilitate efficient spatio-temporal overlap detection between moving objects as a spatio-temporal range query, which is defined as definition 2.9.1.

Definition 2.9.1. ***Metric Range Query****: Given an object set O, a query object q and a search radius r in a metric space, a metric range query returns the objects in O that are within distance r of q, i.e. $MRQ(q, r) = \{o | o \in O \wedge d(q, o) \leq r\}$*

In the next Section 2.10, we investigate metric indices for efficient range query.

2.10 Alternative candidate indexing techniques

Chen et al. [15] also provided a comprehensive and systematic evaluation of metric-space indices that spans over compact partitioning techniques and pivot based partitioning. Compact partitioning techniques are classified as generalised hyperplane

partitioning, ball partitioning, hash partitioning and a hybrid of these techniques . This extensive evaluation spanned over main-memory and secondary-memory based indices and provided a ranking based on number distance computations and run-time given a range query and k-Nearest Neighbour (k-NN) query. Based on their performance evaluation of query range query on main-memory, the highest ranking technique reported was Multiple Vantage Point tree (MVPT), followed by Extreme Pivot Table (EPT*), Bisector Tree (BST) and Geometric Near-Neighbour Access tree (GNAT) and its variants. Since MVPT and partitioning of BST and EPT also utilises ball partitioning, it posed similar challenges to that of an M-tree in a setting of a distributed cluster of nodes. Since the best hyperplane based partitioning, top-down hierarchical strategy is GNAT, in the next Section 2.11 we review the classic GNAT approach.

2.11 Geometric Near-Neighbour Access Tree

Introduced by Brin [12], GNAT is a hyperplane based metric-space indexing technique offering efficient range search over n-dimensional metric spaces. This technique utilises a selection of pivot points, referred as "Split Points" to partition all the individual points into mutually exclusive sets of "Near Neighbours" associating them to their closest "split point". This process is repeated for each set of "Near Neighbours" till the expected degree or arity of GNAT is established. A 2D example of this process as shown in Figure 2.13.

2.11.1 Split-point and Tree Construction

The author mentions that the effectiveness of the technique hinges upon selection of reasonable split points far apart from each other, thereby splitting the metric-space into balanced regions. For a "good-enough" split point selection the author suggests

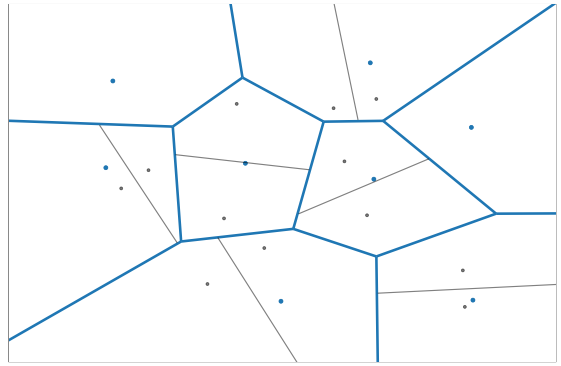


Figure 2.13: 2D GNAT Space splitting in Brin [12]

a heuristic strategy, translated into algorithm 2. Please note that the author does mention that the constant 3 as a heuristic best practice found while experimenting with this strategy. Thus, repeating this process each of the "Near neighbour" set D_{s_i} associated with the resultant split point s_i and adjusting the degree results in a nested membership of all metric-space points.

Algorithm 2 Split point selection strategy - Brin [12]

Input: P, k with $|P| \geq 3k$
Output: S : set of split points

```
Let the total set of metric points be P and the required no. of split point be k
Randomly sample 3k points from P into a candidate set C
Choose the first split-point s_1 from C
Add s_1 to result set S and remove from C
while |S| ≠ k do
    for c_i ∈ C do get distance (c_i, s_j), s_j ∈ S
    end for
    Select c_i with the maximum, distance from all s_j ∈ S
    Add c_i to S and remove from C
end while
```

2.11.2 Spatial pruning

For efficient spatial pruning, during tree creation for each split-point an auxiliary data structure containing the relative minimum and maximum distance from their

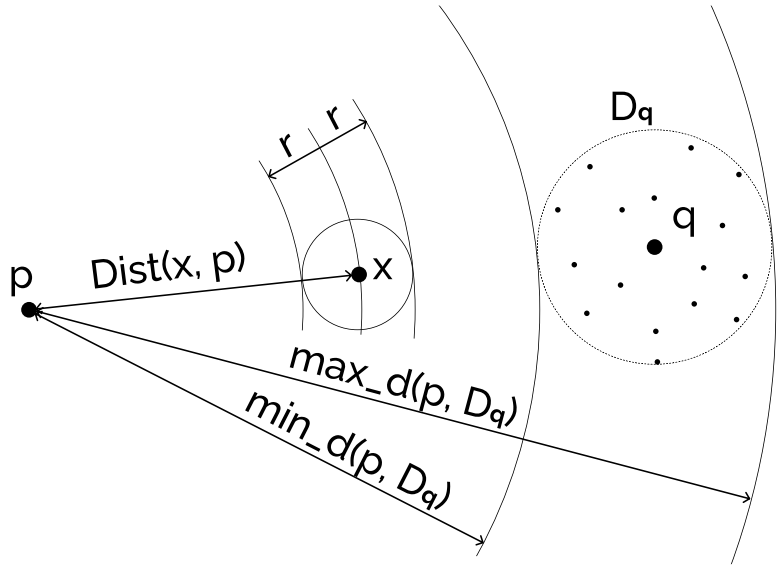


Figure 2.14: Pruning branches using ranges, Brin [12]

sibling split point's "Near neighbour" set including the sibling split point is also created. Utilising this meta-data, an efficient space-pruning strategy is employed during range search. As illustrated by the author in Figure 2.14, given a query point x, and a range r, while evaluating query overlap for a split point p, we can utilise the range table to determine whether we can safely prune a sibling split point q and their "Near-Neighbour" set D_q.

2.11.3 Visualising range query for 3D metric spaces

As an example a GNAT implemented with 3D metric-space points supports spherical volume queries as is with a centre and and a query range as shown in Figure 2.15. However for uncertain trajectories, instead of a metric point our foundational data structure is a space-time cylinder as shown in Figure 2.16.

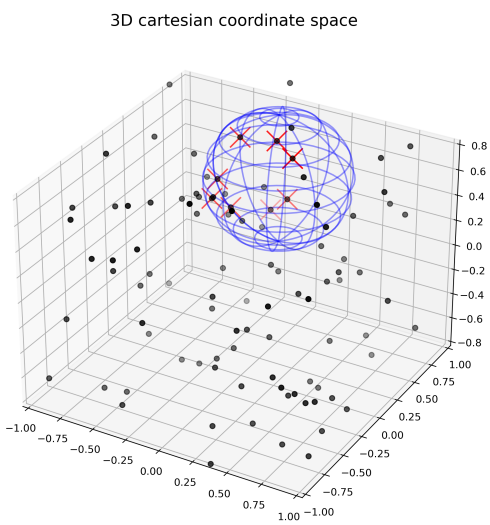


Figure 2.15: Range query (in blue) in 3D metric space with GNAT

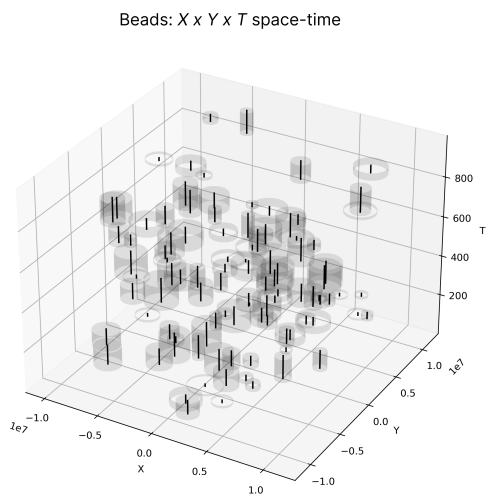


Figure 2.16: Beads in $X \times Y \times T$

2.12 Summary

In this chapter first, from Section 2.1 to 2.4 we review the background on ABMS and their utility in individual level modelling, infectious disease modelling, complexities of modelling SARS-CoV-2 and related relevant modelling work. Second, from Section 2.5 to 2.10 we review relevant background on the complexities of real-world spatio-temporal data, uncertain trajectory encoding, UTM-tree and metric space indexing. Based on this, we refine the overall thesis problem into a workflow of simulation and indexing in the next chapter.

Chapter 3

Problem analysis and research overview

In this chapter we translate our thesis objectives into a list of specific research activities and how they fit into the contagion simulation workflow.

3.1 Contagion simulation workflow

Based on the literature surveyed and the context of rapidly evolving intervention measures and strains we adopt a simulation workflow similar to that of Klepac et al. [52] as described in Figure 3.1. The scope of the simulated population and geographic area was constrained to smaller communities, similar to that of Hunter et. al [44]. Please note, as described in the workflow, when working with synthetic contact network data, we take the output of a agent based simulation of human mobility. This network serves the basis for our network based stochastic contagion simulation. When working with real world mobility data, we expect to efficiently generate through efficient indexing.

The process design was informed to facilitate the following specific goals:

- Design or import compartmental models in epidemiology on-demand.

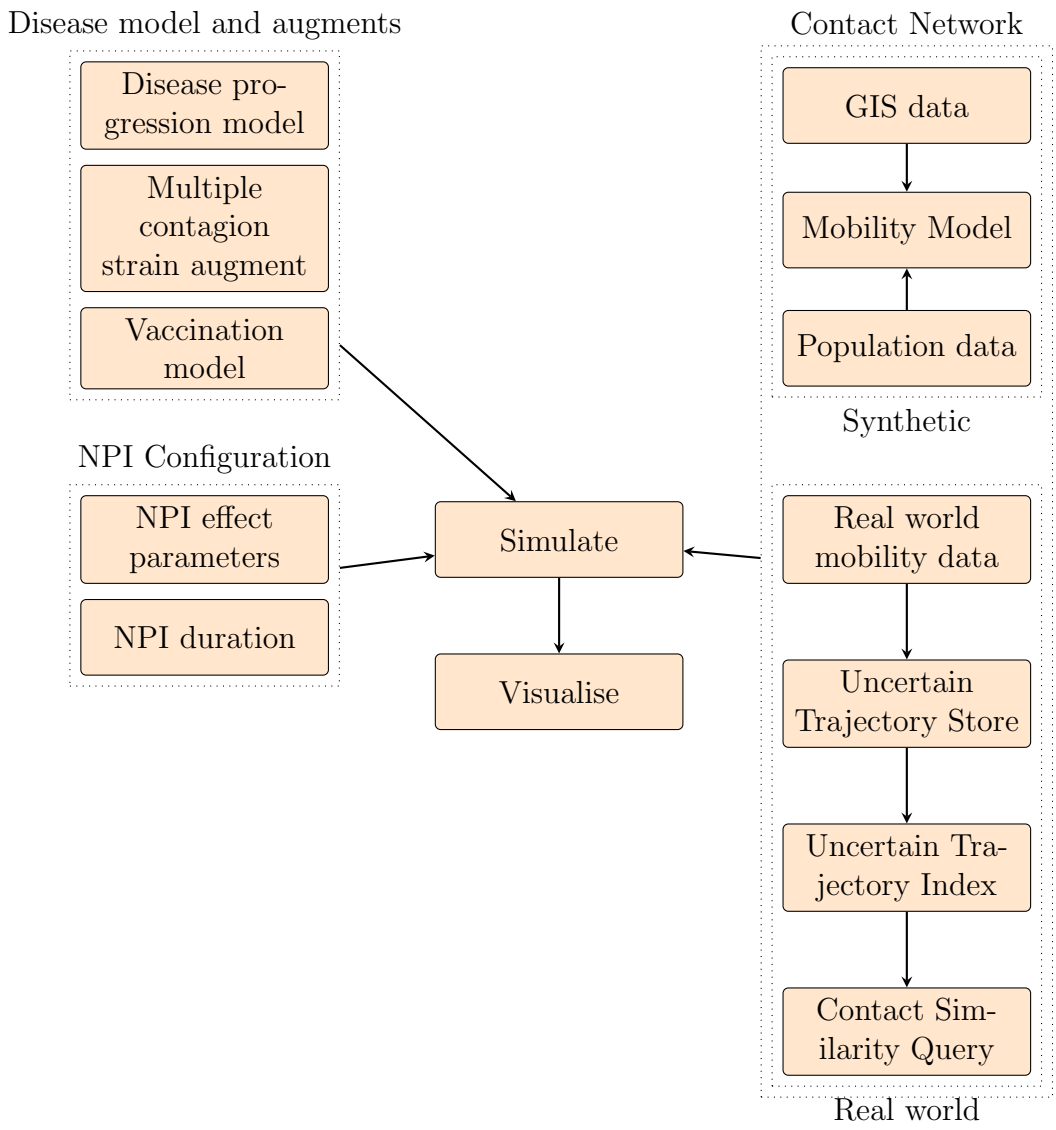


Figure 3.1: Contagion simulator workflow

- Implement efficient spatio-temporal index to generate contact networks from real-world mobility data.
- Maintain multiple contagion strains and their effects on the disease progression.
- Manage initial vaccination status and in-simulation vaccination disbursal and their effect on disease progression.
- Simulate diagnostic tests with sensitivity and specificity along with testing disbursal strategies and contact-tracing strategies.
- Simulate NPIs like physical distancing and masking that reduces intensity of possible infection.
- Simulate closures and lockdowns.
- Report aggregated and individual statistics of relevant compartments, transition and their combinations that provides decision support.

3.2 Incorporating real-world contact network

As described in Figure 3.1, the proposed simulator framework may utilise synthetic and real-world contact networks interchangeably. Therefore, to facilitate Contact Similarity Query (CSQ) as stated by Zhang et al. [98] and similar measures we focus our research on utilising an alternative uncertain trajectory index that is viable for a massive data-store.

3.3 Description of the data available

For the generation of synthetic contact network, the partner organisation utilises OpenStreetMap [73] for building and land use data and Statistics Canada [70] for

demographic and labour data for a community. This provides the input for generating statistically similar individuals with respect to aggregated demography and assigning them expected places for work and school which serves as inputs for the mobility model for Citisketch™. The output from Citisketch™ is a JSON type nested array of arrays. Each element of the outermost array corresponds to a simulated individual. Each element of the nested array is the description of every trip the simulated individual had made in a 24-hour period. The details include source and destination locations, arrival time and departure time. Model parameters for each sub-model is informed from available literature. For initial experiments with spatio-temporal indices, taxi location data by Yuan et al. [96] was utilised and stress tests were performed with uniform random synthetic uncertain trajectory data.

3.4 Research overview

In the next couple of chapters we will focus on the two broad objectives of this work. Chapter 4 focuses on the overall contagion modelling framework with synthetic contact network. Chapter 5 describes how might we utilise an alternative indexing technique for massive uncertain trajectory stores.

3.5 Summary

In this chapter we describe the two lines of research, contagion simulation framework and spatio-temporal indexing in the context of an overall contagion simulator workflow, we list our sources of data and briefly mention the relevance of the next two chapters.

Chapter 4

Contagion Simulator

This chapter focuses on different aspects of the simulator workflow as described in Figure 3.1 except for the real-world contact network generation, which is discussed in Chapter 5. The contact networks referred in this chapter are extracted from the CitiSketch [46] tool that simulates human mobility patterns.

4.1 Designing compartmental contagion model

Depending on the observable data on contagion dynamics, the nature of the contagion and intended public health policies and outcome, one needs to design and augment the underlying compartmental model of disease progression. For example, He et al. [39] noted that for SARS-CoV-2 the infectiousness of individuals peaked before the onset of symptoms, causing an estimated about 44% of subsequent infections which is not the case for many other infectious diseases. Furthermore, there is also evidence of asymptomatic carriers of infection as modelled by Firth et al. [29] which makes real-time observation actual number of infectious individuals with diagnostic testing for a population an insurmountable task. Based on the survey of ongoing modelling efforts, it is clear that most of the compartmental disease models have changed over time based on observable data, surveillance techniques and intervention measures.

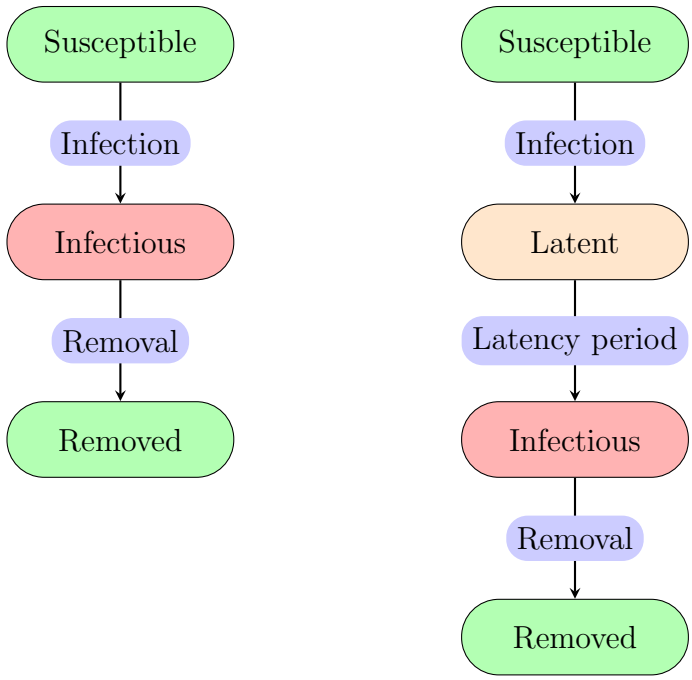


Figure 4.1: Compartmental models as directed graph, SIR on the left, and SLIR on the right.

Therefore, it is reasonable to assume that future models will continue to be augments of simpler models. To ensure that a simulator implementation of this framework is adaptable for present and future complexities, unlike any of the surveyed work, we utilise directed graph: one of the most versatile data structures to encode disease progression model and associated augmentation for the same.

4.1.1 Encoding a contagion model as a directed graph

Consider the two compartment models SIR and SLIR. For an individual level model, they represent the structure of a finite state machine for every simulated individual as shown in Figure 4.1.

Here, for the SIR model, the infection propagating transition is the S-I edge and for the SLIR model, it is the S-L edge. The Infection transition is a function of number of possible infection event a susceptible individual iterates within the simulation. This is a function of each emergent susceptible-infectious pair for a given simulation

Table 4.1: SIR transition matrix

	Susceptible	Infectious	Removed
Susceptible		$f(contact_time)$	
Infectious			γ
Removed			

Table 4.2: SLIR transition matrix

	Susceptible	Latent	Infectious	Removed
Susceptible		$f(contact_time)$		
Latent			$(latency_period)^{-1}$	
Infectious				γ'
Removed				

instance. On the other hand, the latency period and removal may be modelled as transitions of a Markov process with their parameters based on observed mean period of latency and time between infectious and removed (recovering or dying from the contagion) state for all individuals alike. While the directed graph is an intuitive representation of the disease model, we convert it into a transition matrix for computational efficiency.

4.1.2 Transition matrix

Compartmental models SIR and SLIR, in Figure 4.1 yield the following row-wise transition matrices as described in Tables 4.1 and 4.2 respectively.

Here, γ is the inverse of mean length of infectious period or usually termed as "removal rate". And *latency_period* is the mean length of time an individual is carrying a contagion before becoming infectious. Therefore, for SLIR the mean period of an individual carrying the contagion is $latency_period + \frac{1}{\gamma'}$, where γ' is the removal rate from infectious to removed compartments.

Thus for computing non-infectious transitions during simulation, any individual is

constrained to a single row of this matrix.

4.1.3 Non-infection transition as Markov Chain

Let X_i be the state of a an individual X at i^{th} simulation step. Then, X_{i+1}, is a state transition inferred from a transition matrix (barring infecting kind) $A \rightarrow B$ with a constant $k_{A \rightarrow B} \in [0, 1]$ is:

$$Pr(X_{i+1} = B | X_i = A) = k_{A \rightarrow B} \tag{4.1}$$

Please note that the probability of X not transitioning, to stay in state A from i to $i + 1$ simulation step is:

$$Pr(X_{i+1} = A | X_i = A) = 1 - \sum k_{A \rightarrow A'} \tag{4.2}$$

where $A' \in \mathbb{S}_A$ set of all possible transition states from A and, $A \neq A'$.
In the simplest form this is a Markov Chain.

4.2 Interpreting contact events

For a single time-step, this may be abstracted as a function that maps two individuals X, Y and a simulation step i to a shared time t in the aggregated case, or a list of shared times $t_1, t_2, ...t_m$ at l locations for the location specific case with l is independent of m. i.e. two individuals may have multiple shared times at multiple locations.

$$f(X, Y, i) \rightarrow \begin{cases} t, \ if \ \ aggregated \\ t_m \in \mathbb{T}_{X,Y,i} = t_1, t_2, ...t_m, \ if \ \ location\ based \end{cases} \tag{4.3}$$

For venue specific contact events, we may parameterise the scale of the probability of

infection with respect to the type of venue and the nature of activity the individuals are expected to engage in said venue. For example, two students co-located in the school library may have different chances of infection propagation when compared to one of the students and a librarian for the same amount of time overlap.

4.3 Infection propagation

As per the mechanistic model of viral load transmission by Challenger et al. [14], each infectious individual may be thought of as a emitters of air-born virus by virtue of exhalation. If and only if there is sufficient transfer of said virus (a.k.a viral load) from an infectious to susceptible individual, there is a possibility that, infection may spread to the susceptible individual.

4.3.1 Infection probability function

Assuming all infectious individual exhale uniformly (over a period of time) the probability of infection $Pr(infection)$ for each event $\propto$ shared time t. We have chosen the *hyperbolic tan* function for our mapping with a scaling of A as parameterised amplitude and a ramp up time $t_{ramp-up}$ to regulate infection chance our infection function.

Therefore a single infection event for an infected individual $X_I \in I$ to infect $X_S \in S$ with $contact_time = t$, where I and S are sets of infectious and susceptible individuals respectively is a discrete probability governed by the following parameters:

- A : Base infection chance.
- $t_{ramp-up}$: A constant time parameter that offsets the infection chance to be low below a certain threshold.
- μ_{strain} : Relative infectivity of a particular $strain$.

- $\epsilon(inf)_{strain-dose_n}$: Vaccine efficacy for a particular *strain* of contagion carried by X_I with respect to the $dose_n$, i.e. number of doses of vaccines administered to X_S.

Simplifying the parameter names, the infection probability function is the equation 4.4

$$Pr(X_I \xrightarrow{infects} X_S) = (1-\epsilon)\mu A \tanh(\frac{t}{t_{ramp-up}}) \tag{4.4}$$

Please note that the scaling factor A may be modulated to simulate different contexts of relative contact intensity depending on $venue \times activity$ pairs.

4.4 Vaccination

As vaccination and the lack thereof affect the overall proportions and mean infectious and hospitalisation periods, they may be encoded as duplicated contagion compartment model with different parameters as shown in Figure 4.2. The vaccination rates may be modulated during simulation to capture real-world vaccine disbursal data or to simulate counterfactual scenarios of vaccine disbursal strategy. This is one of the aspects of disease progression that is individualised. In the same vein, by utilising compositions of disease progression tracks, it is possible within this framework to account for individual level variations due to age, gender, preexisting health factors and so on.

4.5 Diagnostic testing and isolation

So far, the contagion models and their discussed extensions assume there is full knowledge of the population who are infected. Which may be true for contagions manifesting their effects with apparent symptoms. However, for contagions like

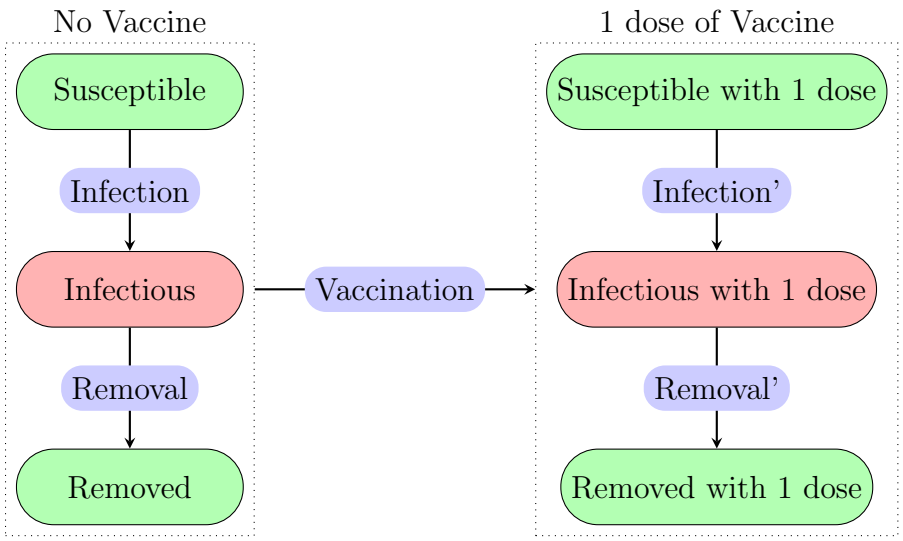


Figure 4.2: Composition of base models to capture effect of vaccination on disease dynamics

SARS-CoV-2 a significant portion of the population may show mild or no symptom at all and there are further complications arising with overlapping symptoms associated with the common flu. Therefore, in these situations, diagnostic tests designed specifically for the contagion in question, are the tools utilised by health authorities to track the (contagion) spread.

Our simulator takes a time-deterministic approach to testing, with the following flow in Figure 4.3 for any individual requesting a test. With the introduction of testing as a sub-model, we split the Infectious compartment into unconfirmed and confirmed infectious compartments respectively as shown in Figure 4.4.

We start with the following assumptions:

- The simulation begins with a first day disease import set to "(unconfirmed) Infectious" compartment.
- All new positive infections are initialised with "(unconfirmed) Infectious" compartment.
- A transition from "unconfirmed" to "confirmed" compartment (of Infectious)

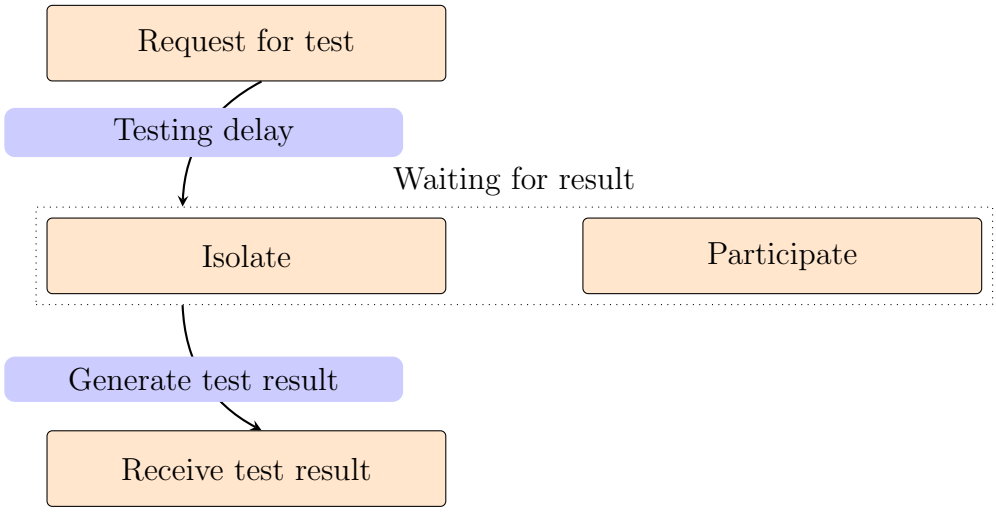


Figure 4.3: Diagnostic testing flow

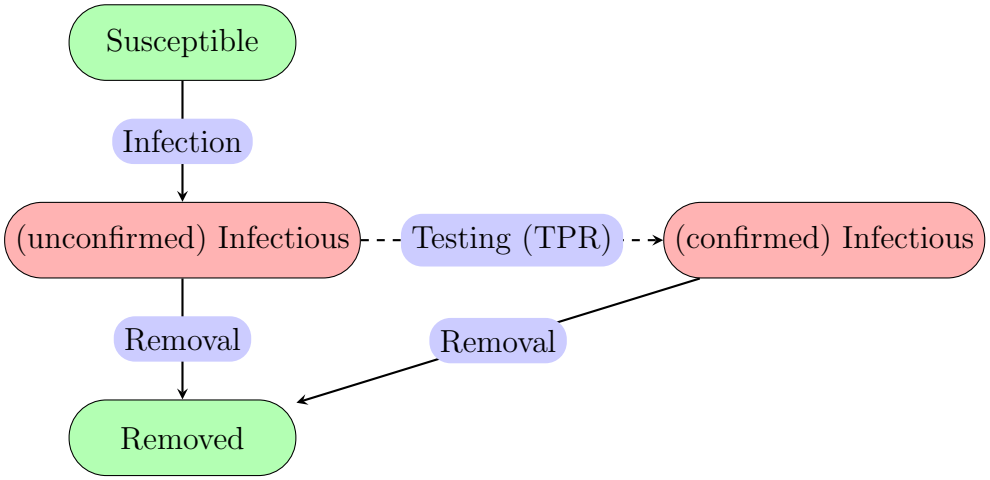


Figure 4.4: SIR Compartmental model augmented for testing.

is a function of "testing" of individuals in "(unconfirmed) Infectious" compartment.

- **Testing assumptions** The individuals in "(unconfirmed) Infectious" population seek tests.
  - For New Brunswick region there are two types of tests available, as stated by the government in their website [37], namely: Pick-Up Point of Care (POCT) Rapid Antigen Test or at home Rapid Antigen Test (RAT) and In-person Polymerase Chain Reaction (PCR) laboratory test which has a delay of 15 minutes and 24 to 48 hours [67] respectively to yield test results. Additional delays may be caused by availability and accessibility of each kind of tests and corresponding resources, including and not limited to test-kits, laboratory technicians and so on.
  - There exists a maximum testing capacity per day.
  - There exists a delay from requesting a test to getting the result. In other words, the state transition from "unconfirmed" to "confirmed" is dependent on test type and availability.
  - There exists test accuracy parameters, namely sensitivity a.k.a True Positive Rate (TPR) and specificity a.k.a True Negative Rate (TNR).

    For this model, the TPR and (1 - TNR) dictates the transition from "unconfirmed" to "confirmed" and staying in "unconfirmed" state respectively.

To accommodate isolation as a consequence of testing we augment the model further as shown in Figure 4.5. Here we introduce "isolated" (as a compartment modifier). This enables the simulation to account for the non-pharmaceutical advisory along the lines of the following: "Symptomatic individuals are requested to test themselves and isolate till they receive test result". We further assume that individuals in

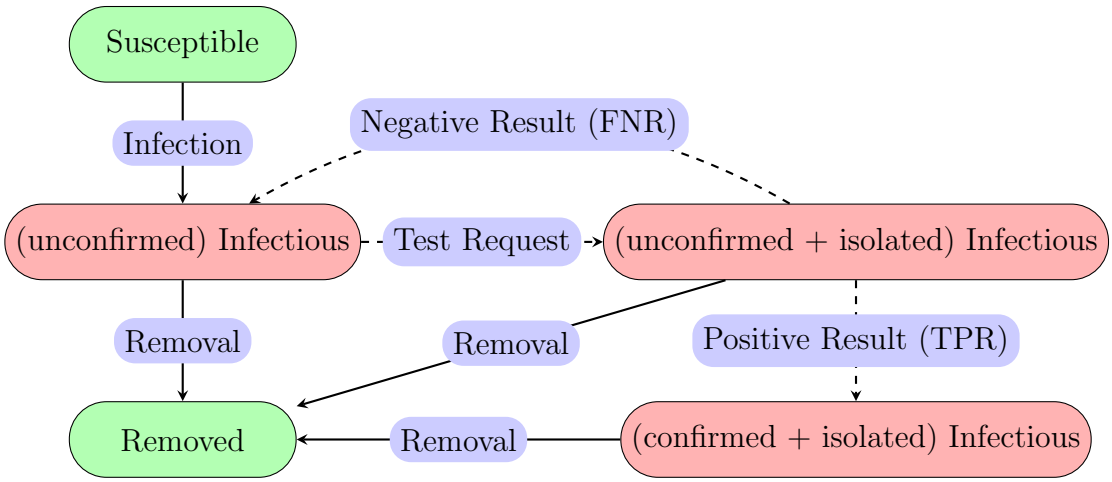


Figure 4.5: SIR Compartmental model augmented for testing and isolation.

"isolated" type compartments do not participate in propagating the contagion along with assumptions made for "SIR with Testing" model.

4.5.1 Diagnostic Test transition parameters

Based on recent research by Dinnes et al. [26], diagnostic tests of different types and brands offer different sensitivity (True Positive rate) and specificity (True Negative rate) for determining positive infection status. Furthermore, these parameters are also dependent on whether the individual is symptomatic or not, or more generally speaking dependent on the age of infection.

4.5.2 Diagnostic Test scheduling

A 2D matrix of dimension $(test\ type, test\ scheduled\ on, result\ day, result) \times no\ of\ individuals$ decouples the compartment model transitions from the testing system and making it flexible enough for even extreme scenarios like cascading failure of testing infrastructure due to unavailability of diagnostic test from roll-over demands.

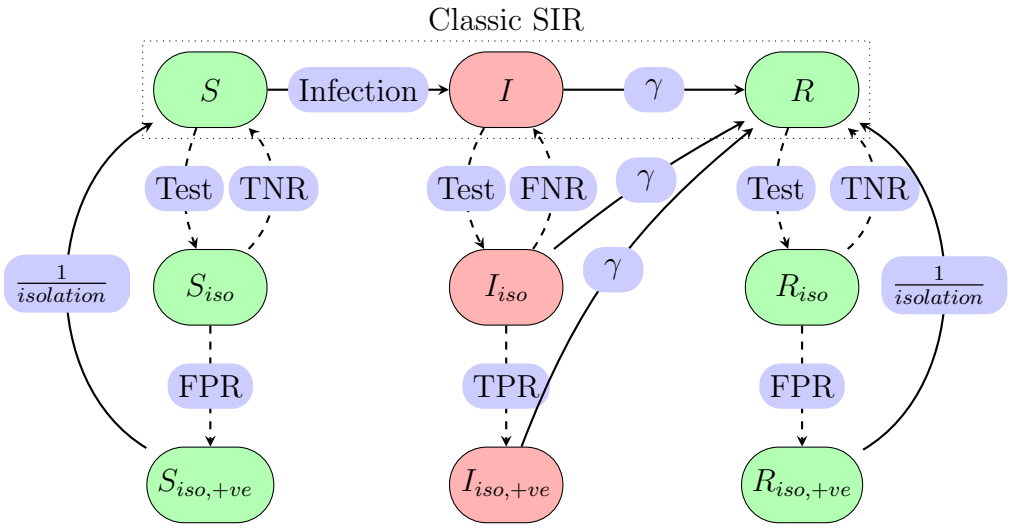


Figure 4.6: SIR Compartmental model augmented for testing, isolation and contact tracing.

4.6 Contact tracing

Our simulator borrows its contact tracing model from Kretzschmar, et al. [55].

To accommodate this, we modify the compartments as shown in Figure 4.6.

Here compartment names are shortened and transitions are parameterised as follows:

- $S, S_{iso}, S_{iso,+ve}$: Susceptible, Susceptible and waiting for result and Susceptible and falsely identified as positive compartments respectively.
- $I, I_{iso}, I_{iso,+ve}$: Unconfirmed Infectious, Unconfirmed Infectious and waiting for result and Confirmed Infectious compartments respectively.
- $R, R_{iso}, R_{iso,+ve}$: Removed, Removed and waiting for result and Removed and falsely identified as positive compartments respectively.
- γ: Removal.
- TNR, FPR: True Negative and False Positive rates respectively for diagnostic test performed on a non-infectious individual.

- TPR, FNR: True Positive and False Negative rates respectively for diagnostic test performed on a infectious individual.

- *isolation*: Mean time period of expected isolation for non-infectious individuals with a positive result.

We assume contact tracing moves an agent from its base state to its corresponding isolated state. And for every new transition to "(confirmed + isolated) Infectious" compartment, contact tracing is performed for the neighbours of the specific agent with a **tracing delay**. The coverage of the trace is determined by **tracing depth** of days and contact levels. For example, at simulation day D, let the set of all contacts of confirmed and isolated infectious agents at $D-1$ (and not already confirmed and isolated Infectious) is as shown in equation 4.5.

$$contact(I_{iso,+ve}) \subset S \cup I \tag{4.5}$$

A trace function over this set, results in the following composition and transitions as shown in equation 4.6.

$$trace(contact(I_{iso,+ve})) := \begin{cases} S \Rightarrow S_{iso} \\ I \Rightarrow I_{iso} \end{cases} \tag{4.6}$$

Further assuming, perfect tests, infinite testing capacity and instantaneous test results, we have the following composition and resultant transitions as shown in equation 4.7.

$$test(trace(contact(I_{iso,+ve}))) := \begin{cases} S \Rightarrow S_{iso} \Rightarrow S, \\ I \Rightarrow I_{iso} \Rightarrow I_{iso,+ve} \end{cases} \tag{4.7}$$

4.6.1 Contact gathering flow

Given an individual and a start day of contact tracing, the flow of gathering contact is shown in Figure 4.7. The behaviour of the tracing strategies are based on the following parameters:

- **Tracing delay**: If a positive result is observed for simulation day D, request test on D + *tracing delay*, for all traced contacts for range from [D - "*tracing depth*" to D].
- **Tracing depth**: The number of days of contact networks to considered from the time of request of contact tracing.
- **Tracing Coverage**: Separate scaling factor for simulating inaccuracy of reporting of close and casual contacts.

4.7 Hospitalisations and deaths

For the sake of visual clarity let us have a short-hand for compartments that includes testing and isolation as shown in Figure 4.8.

Hospitalisation compartments are introduced as an extension of infectious compartments, queued for testing and tested positive. And the removed compartments into separate compartments for recovered and dead. The modified compartment model is shown in Figure 4.9 with the corresponding parameters definitions in Table 4.3.

Please note, with strains of contagion that have shown changes in observed periods of infection, hospitalisation and the introduction vaccination a compartmental model that captures a contagion dynamics with sufficient detail is composed of multiple instances of the augments of extended model shown in Figure 4.9 with respect to constituent contagion strains and vaccination.

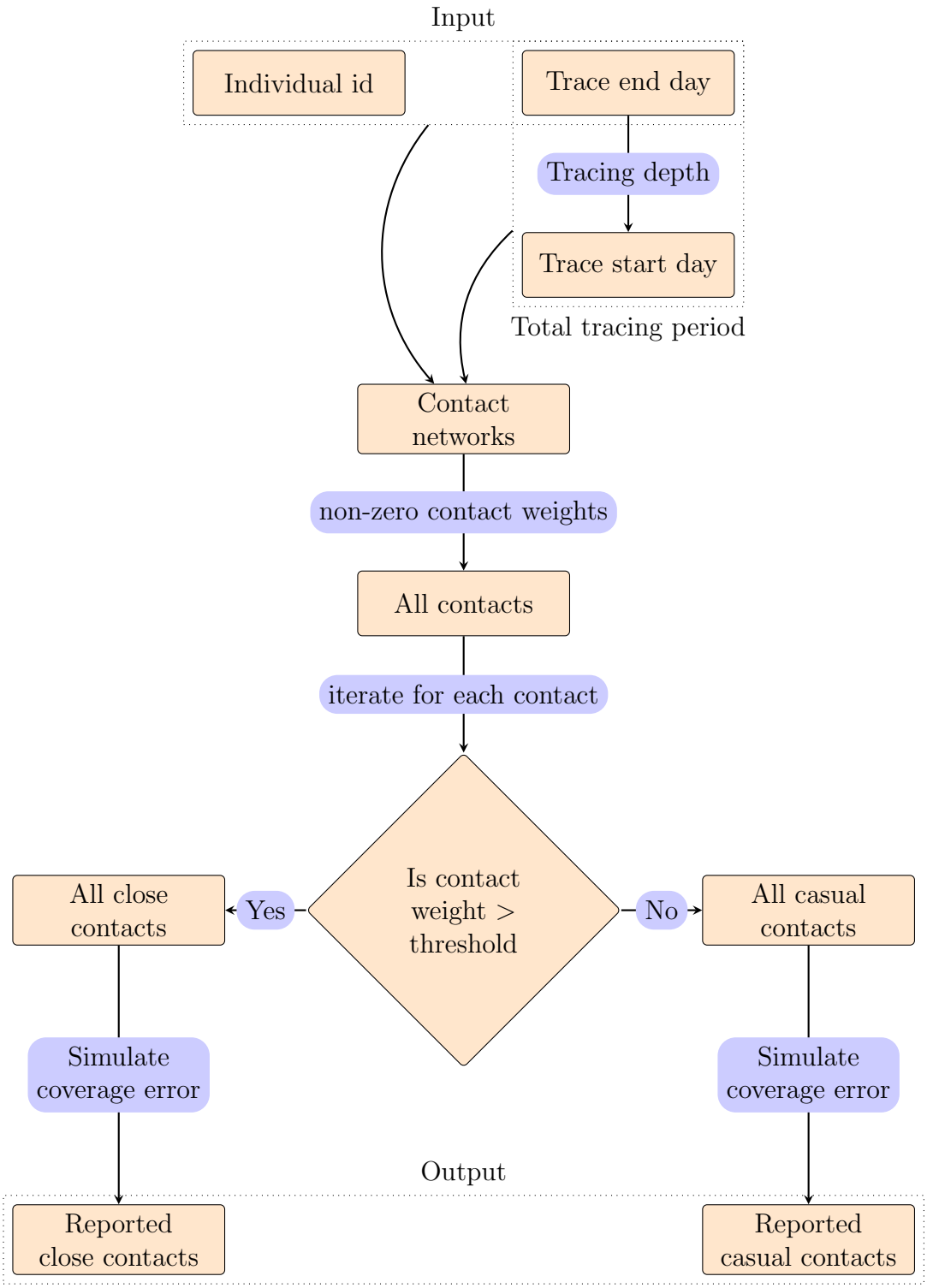


Figure 4.7: Contact gathering flow

Table 4.3: List of notation for extended SIR model shown in Figure 4.9

Notation(s)	**Definitions**
$S_{testable}$	Shorthand for Susceptible, isolated Susceptible and falsely identified as positive Susceptible compartments.
$I, I_{iso}, I_{iso,+ve}$	Unconfirmed Infectious, Unconfirmed Infectious and waiting for result and Confirmed Infectious compartments respectively.
TPR, FNR	True Positive and False Negative rates respectively for diagnostic test performed on a infectious individual.
H, H_{ICU}	Compartments for Hospitalisation and Hospitalisation with ICU respectively.
$R_{testable}$	Shorthand for Recovered, isolated Recovered and falsely identified as positive Recovered compartments.
D	Compartment for individuals died due to contagion.
β'	Infection rate as a function of pairwise contact time
γ	Inverse of total period a person may carry the contagion
f_{gen}	Fraction of infected population dying without hospitalisation
κ	Inverse of mean period before hospitalisation
ρ	Inverse of mean period of hospitalisation without ICU
f_h	Fraction of hospitalised population dying with hospitalisation
ν_c	Fraction of population requiring hospitalisation with ICU
ν_h	Fraction of population requiring only hospitalisation
σ	Inverse of mean period of hospitalisation with ICU
f_c	Fraction of hospitalised population dying with hospitalisation and ICU

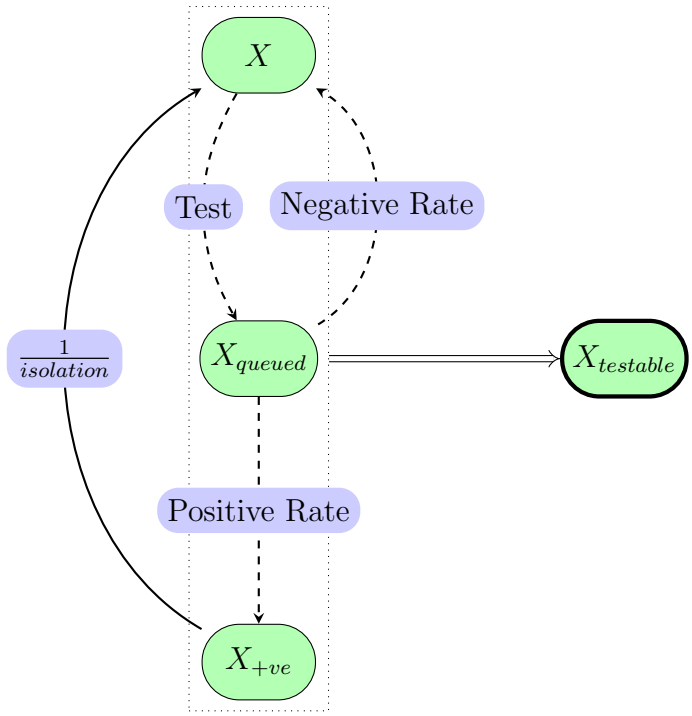


Figure 4.8: Shorthand for compartments with diagnostic testing

4.8 Infectiousness scaling

To simulate a non-infectious period like that of SARS-CoV-2, we also introduce a infection probability scaling function as equation 4.8, provided by Robert Santacruz, a colleague in the research group:

$$A(t') = A \cdot normalised(\frac{t'^4}{e^{0.8t'} + 2}) \tag{4.8}$$

All infectious states' base infectious amplitude A is scaled according to time elapsed since infected t'. Please note, this is an attempt at generalising the "Latent" status of an infected individual as an alternative to having a separate compartment.

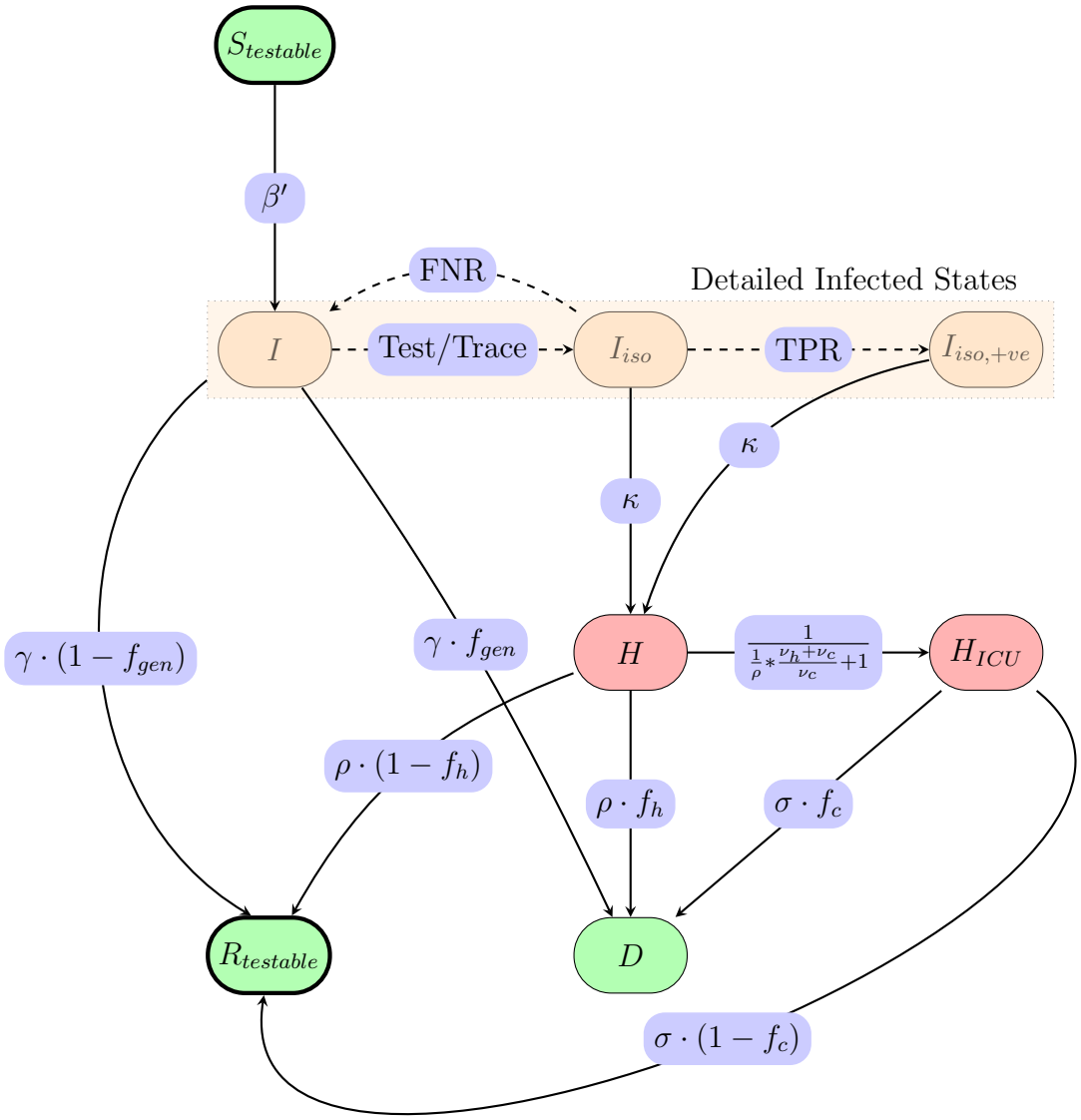


Figure 4.9: SIR compartmental model augmented for testing, isolation, contact tracing, recoveries, deaths and hospitalisations.

4.9 Non pharmaceutical interventions

Physical distancing and masking are simulated as scaling of infection probability function. Closures and lockdowns are simulated with the help of selectively removing specific venues through the CitiSketch tool [46] prior to simulating human mobility to generate synthetic contact network.

4.10 Simulation algorithm

Before stating the simulation algorithms, we review a list of typical simulation wide configuration parameters in Table 4.4.

4.10.1 SIR simulation loop

At its simplest form, for a SIR model, the simulation pseudo code may be described as follows:

1. Set up initial infectious population and mark the end of day 1.
2. From day 2 till the end of simulation
   (a) Iterate on the list of infectious from previous day
      i. **Propagation step**: "Do they infect any of the susceptible from previous day?"
   (b) **Evolution step**: "Transition any non-susceptible from previous day evolve to their next state"
   (c) This concludes the final states for this day, record and repeat

Table 4.4: Typical simulation configuration parameters

Configuration parameter	**Notes and example value(s)**
Contact matrices	sequence of contact networks per day to simulate repeating patterns of mobility
Simulation time (no. of days)	30,60, 90, 210 days
Disease model	SIR, Extended SIR
Amplitude (A)	0.001, 0.002, 0.0025, 0.003
Ramp up time ($t_{ramp-up}$)	1200 seconds
Vaccinated proportions, $\{y_0, y_1, y_2, y_3\}$	Initial proportion of population not vaccinated, vaccinated with 1 dose, with 2 dose and 3 dose respectively
Relative infectiousness (μ)	Note, u_1 is designated for Wild variant with a value 1.0
Vaccine efficacy (ϵ_{inf})	Note, these are odds of vaccine efficacy for a potential vaccinated agent getting infected with a particular variant
First day disease import	No. of infectious individuals at the start of simulation.
Testing capacity and strategy	Types and no. of tests available per day and test disbursal strategy.
Contact tracing strategy	Parameterised contact tracing and traced-testing context.
NPI parameters and strategy	Parameterised NPIs and their timing strategies.

4.10.2 Augmented simulation loop

With the introduction of the extensions to the base model the SIR simulation loop is extended to the following pseudo code:

1. Set up initial infectious population and mark the end of day 1.
2. From day 2 till the end of simulation
   (a) Iterate on the list of susceptible from previous day
      i. **Vaccination step**: "Did anyone get vaccinated yesterday?"
   (b) Iterate on the list of test-seekers from previous day (excluding the newly vaccinated)
      i. **Voluntary Test Request step**: "Did anyone request test yesterday?"
   (c) Iterate on the result-seekers from previous day (including same day test seekers)
      i. **Test Result Generation step**: "Did anyone receive test result yesterday?"
         A. Iterate over agents receiving positive test result
         B. **Contact Tracing step**: Gather all the traced contacts of positive result receiving agent
         C. If: there is a contact tracing delay of more than a day, schedule tests
         D. Else: **Same day traced-testing step**: Generate test result for all the traced contacts except for the ones who already requested and received test result yesterday.
         E. Contact trace the positive results from the previous step.

F. If: maximum same-day-tracing-depth has not been reached exclude the ones who already requested and received test result yesterday and repeat from step D.

G. Else: Schedule testing for contact traced agents from step E for next day (with or without isolating these agents from propagation).

(d) Iterate over remaining infectious agents from previous day

i. **Propagation step**: "Do they infect any of the susceptible from previous day?"

(e) **Evolution step**: "Transition any non-susceptible from previous day evolve to their next state"

(f) This concludes the contagion propagation for this day and finalises the states for the previous day, record and repeat

4.10.2.1 Naive simulation algorithm

The simplest possible loop to simulate disease propagation is a nested **for** structure, with the outer loop iterating over number of simulation days and the inner loop iterating over each individual agent as shown in algorithm 3.

4.10.2.2 Optimised simulation algorithm

Considerable speedup is achieved with the following changes to the naive approach:

- Iterate over infectious agents and compute possible transmissions to suceptible neighbours instead of iterating over susceptibles.
- Evolve the non-susceptible agents in batch as they are mutually independent.
- At the end of a each day find agents in terminal state, i.e. states that cannot evolve and remove from computation.

Algorithm 3 Disease Spread - naive approach

```
Set simulation configuration
Construct disease model with appropriate parameters and convert to a transition matrix T_{states·states}
Load first contact matrix: M_p.
Initialise first day as per vaccinated proportions and first day disease import as S_0 = {0, 0, 0, ...N − 1}, where N is the total number of agent and S_k is a vector of the state of each agent 0 to N for simulation day k
for day 2 to k do
    for each agent i ∈ {0, 1, ...N − 1} do
        if i is susceptible then find Inf_i                ▷ set of infectious neighbours
            for each j ∈ Inf_i do compute probability for j to infect i
            end for
        else Evolve i's current state as per transition matrix T
        end if
    end for
    Load the next contact matrix M_{p+1}
    Record the state vector for this day as S_k
end for
Output the state vectors S_0, S_1, S_2...S_k for aggregation and visualisation.
```

- At the end of each day check, if there are no more infectious agents and rest of the agents are either in susceptible or terminal state, preemptively terminate simulation.

The disease spread algorithm with these changes is listed as algorithm 4.

Algorithm 4 Disease Spread - optimised approach

```
Same as Step 1,2,3 and 4 of algorithm 3
Construct, set of susceptible for first day = Y_1
Construct, set of non-susceptible for first day = Y'_1
Construct, set of Infectious for first day = Inf_1
for day 2 to k do
    Construct, temporary set of new non-susceptible for today = newY'_day = ∅
    Construct, temporary set of susceptible for today = tempY_day = Y_day
    for each agent i ∈ Inf_day−1 do                          ▷ Iterate infectious
        find Y_i_day from tempY_day                          ▷ Susceptible neighbours of i
        for each j ∈ Y_i_day do
            compute probability for i to infect j
            if j is infected then

                remove j from tempY_day
                add j to newY'_day
            end if
        end for
    end for
    for each agent i ∈ Y_day−1 − newY'_day do                ▷ Propagate susceptibles who weren't infected
        Copy i's current state as today's state
    end for
                                                             ▷ Evolve non-susceptible
    for each agent i ∈ Y'_day−1 do
        Evolve i's current state as per transition matrix T
    end for
    Find set of agents moved to terminals for this day = Term_day   ▷ Remove terminals
    for each agent i ∈ Term_day do
        Fill i's current state till end of simulation day
        Remove i from Y`_day
    end for
    Setup Inf_day, Y_day and Y`_day (for next iteration)
    Load the next contact matrix M_p+1
    Record the state vector for this day as S_k
                                                             ▷ Preempt simulation for stable state
    if |Inf_day| = 0 and all agents are in Y_day or Term_day then
        Copy S_day from S_day+1 to S_k
        break                                                ▷ Break from day-wise iterator
    end if
end for
Output the state vectors S_0, S_1, S_2...S_k for aggregation and visualisation.
```

4.10.2.3 Functional simulation algorithm

A SIR simulation loop may be composed as mutually recursive function calls of infection propagation and removal from infectious state as follows:

```
function SimulationWrapper(simulation_configuration){
    // Initialise simulation
    propagate_infection(simulation_configuration,
                        simulation_day_index,
                        accumulator,
                        removal_from_infected)
}

function propagate_infection(simulation_configuration,
                             simulation_day_index,
                             accumulator,
                             callback_function){
    // End of recursion clause - do nothing and return
    if(simulation_day_index > end_day)
        return accumulator

    // Mutate agent states

    callback_function(simulation_configuration,
                        simulation_day_index,
                        accumulator,
                        propagate_infection)
}
```

```
function removal_from_infected(simulation_configuration,
                               simulation_day_index,
                               accumulator,
                               callback_function){
    // Mutate agent states


    callback_function(simulation_configuration,
                          simulation_day_index + 1,
                          accumulator,
                          removal_from_infected)
}
```

Similarly, a SIR model with isolation, testing and contact tracing, may be composed of chain of recursions of the following functions:

1. $SimulationWrapper \rightarrow queue_tests$
2. $queue_tests \rightarrow get_test_results_and_trace$
3. $get_test_results_and_trace \rightarrow get_sameday_traced_test_results$
4. $get_sameday_traced_test_results \rightarrow schedule_overflow$
5. $schedule_overflow \rightarrow propagate_infection$ (with increment in simulation day index)
6. $propagate_infection \rightarrow removal_from_infected$
7. $removal_from_infected \rightarrow restore_from_isolation$
8. $restore_from_isolation \rightarrow queue_tests$ (end of a chain)

4.11 Implementation

4.11.1 Complete simulator: R

The full simulation workflow has been implemented in declarative paradigm in R programming language [80] version 3.6.3 on R Studio IDE [84] with extensive usage of igraph package [21] for construction and manipulation of compartmental models and ggplot2 package [92] for visualisation. Furthermore, the packages dplyr [94] , iterators [2], jsonlite [72], Matrix [7], tidyverse [93] and reshape [91] have been invaluable for data IO and manipulation.

4.11.2 Simulation loop: Julia

As an exercise in optimisation and cross-validation, the simulation loops for select compartmental models have been implemented in purely functional paradigm in Julia programming language [9] version 1.4.1 on Juno IDE plugin version 0.12.6 for Atom text editor version 1.60.0. The following packages were utilised CSV, DataFrames, Plots, StatsBase, BenchmarkTools, LoopVectorization, JLD2, FileIO, Measures and Glob.

4.11.3 Experimental setup

All of experiments were conducted on an Intel(R) Core(TM) i7-10875H CPU @ 2.3 GHz running Ubuntu 20.04.2 LTS x86_64-bit OS with 32 GB RAM.

4.12 Results

Listed in the Table 4.5 are comparison of mean execution time for a single simulation run for the full framework implementation in R and functional simulation loop in Julia with the following configuration parameters:

- **Contact matrix**: generated from mobility of 4602 individuals from Campbellton community of New Brunswick, Canada, provided via CitiSketch.
- **Viral strains and amplitude**: single strain with 0.003.
- **Initial no. of infectious individuals**: 4, randomly assigned.
- **Total simulation days** : 210.
- **Contact tracing strategy**: Same day traced testing (Mobile app based)
- **Vaccination rates**: Constant.

Table 4.5: Comparing simulation time varying disease model and implementation

Disease model	**No. of compartments**	**R full framework (s)**	**Julia simulation loop (s)**
SIR	3	23.0969	1.4252282
SIR with Vaccination	9	26.5943	Not implemented
SIR with Testing and Isolation	7	22.8668	4.9491476
SIR with Testing, Isolation and Contact Tracing	9	26.5436	8.5852787
SIR with Hospitalisation, Testing and Isolation and Contact Tracing	12	35.4862	Not implemented

The hand-crafted simulation loop for Julia is obviously much faster than the full-framework based implementation in R, however the latter remains more stable with the increase in (non infecting) transitions or increase in compartments of a disease model. Parameterised disease model for SIR and SIR with testing and isolation is listed in Appendix A, Section A.1 as Table A.1 and A.2 respectively.

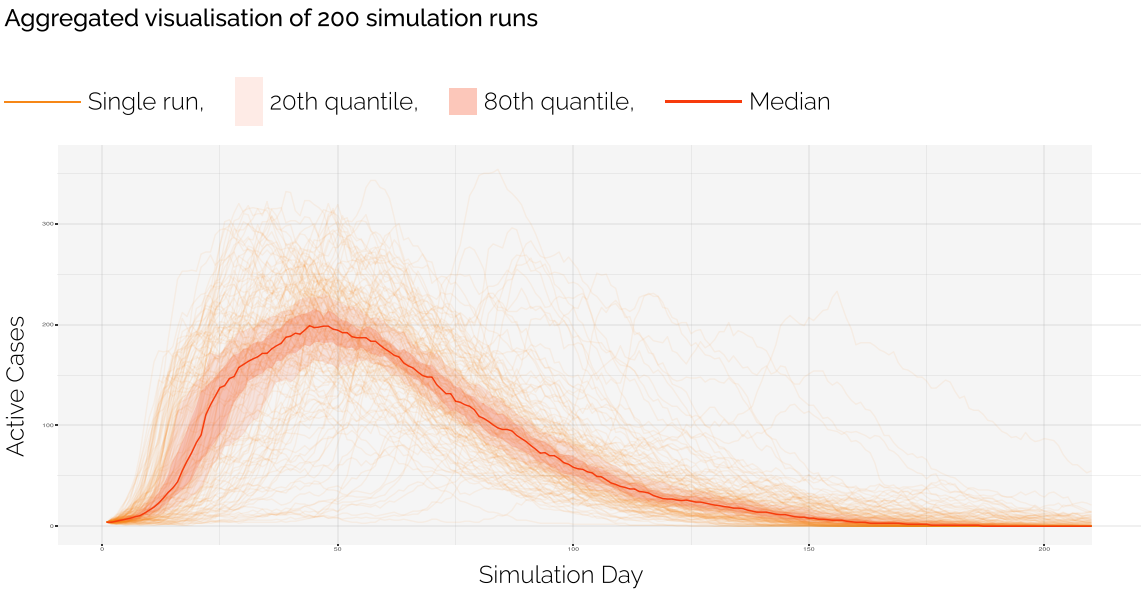


Figure 4.10: Daily active cases for 200 runs

4.12.1 Visualising infection risks

Due to the stochastic nature of the simulator when visualising a metric of interest, say active cases per day the outputs are visualised with aggregated statistics to get a sense of most probable outcomes as shown in Figure 4.10.

4.12.1.1 Multiple instances of single scenario

For a visualising a single scenario as in Figure 4.10, the daily count of active cases per day of each run is a line and day wise range of 20th and 80th quantile respectively are ribbons or areas. Finally, the median count active cases is plot in a darker colour. This composite plot depicts the most probable outcome regions as the median line and the surrounding bands at a glance. The individual line plots are essential to communicate that the worst case scenarios may be significantly worse than aggregated medians. For example, in Figure 4.10, the worst scenario has about 150 more active cases of the simulated infection. Depending on the public health context, this may be a critical piece of information in strategic planning and must be preserved

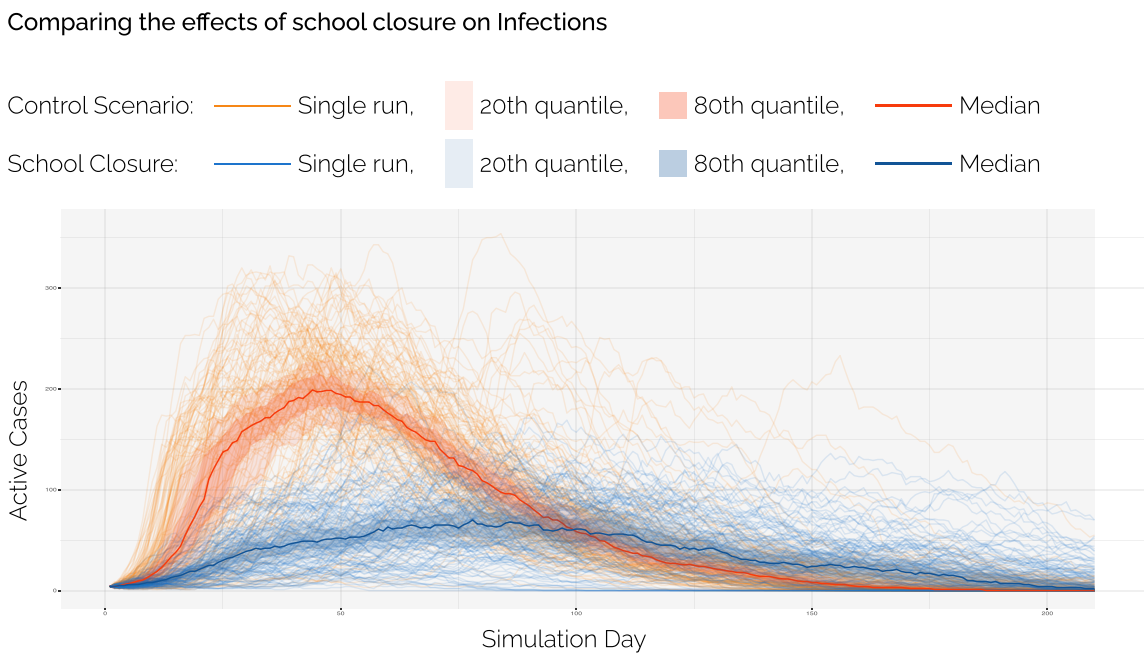


Figure 4.11: Comparing school closure as a strategy to mitigate infection risk

as noted by Juul et al. [48]. Another of observation of interest to public health is the possibility of delayed onset of exponential growth phase of an infection given the same initial simulation scenario parameters.

That being said, counterfactual modelling is effective in answering comparative "what-if" questions and therefore outputs need to visualise multiple scenarios as discussed in the next Subsection 4.12.1.2.

4.12.1.2 Comparing scenarios

In Figure 4.11, we plot daily active cases for 200 runs with its associated aggregated plots to effectively communicate effectiveness of public health intervention in question, which in this case is closing of schools. Hence we have simulation output of a "control" scenario without school closure.

When producing visualisation to compare more than two scenarios it is prudent to plot only the aggregated statistics. For example, in Figure 4.12 a pair of metrics

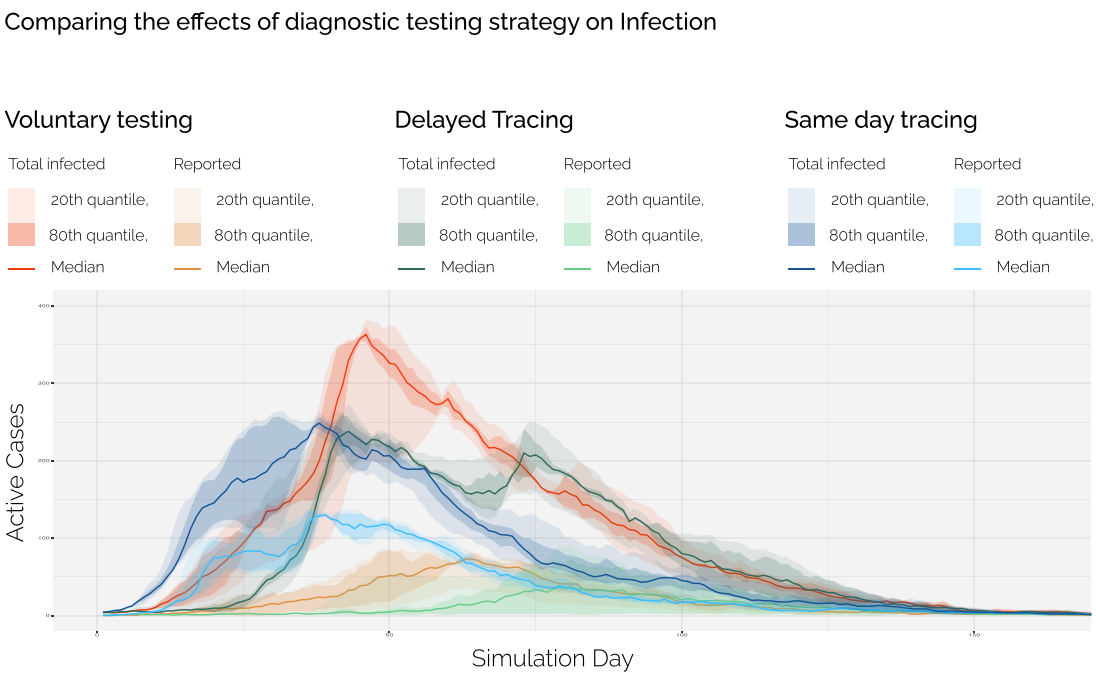


Figure 4.12: Comparing the effects of diagnostic testing strategy on infection risk

namely, observed and total active cases are plotted for each of the three diagnostic testing strategy.

In this comparison, we have two metrics of interest, the observed and the total active infectious cases per day. The contact tracing and testing strategy not only decreases the gap between observed and actual cases of infections, which are only observable by diagnostic testing, they also bring the peak cases of infection down. Furthermore, this reproduces the observation made by Kretzschmar et al. [55] about same day tracing being necessary for effective mitigation of risk.

Furthermore, conditional NPI strategies may be visualised as seen in Figure 4.13. This is a visualisation of three scenarios of reactive closures, one where shopping centres are closed, second where schools are closed and third where both shopping centres and schools are closed once observed active cases reaches a threshold of 50.

Another perspective on infection risk may be gained by comparing relative infection risk for venue types given the makeup of specific community as a categorical time

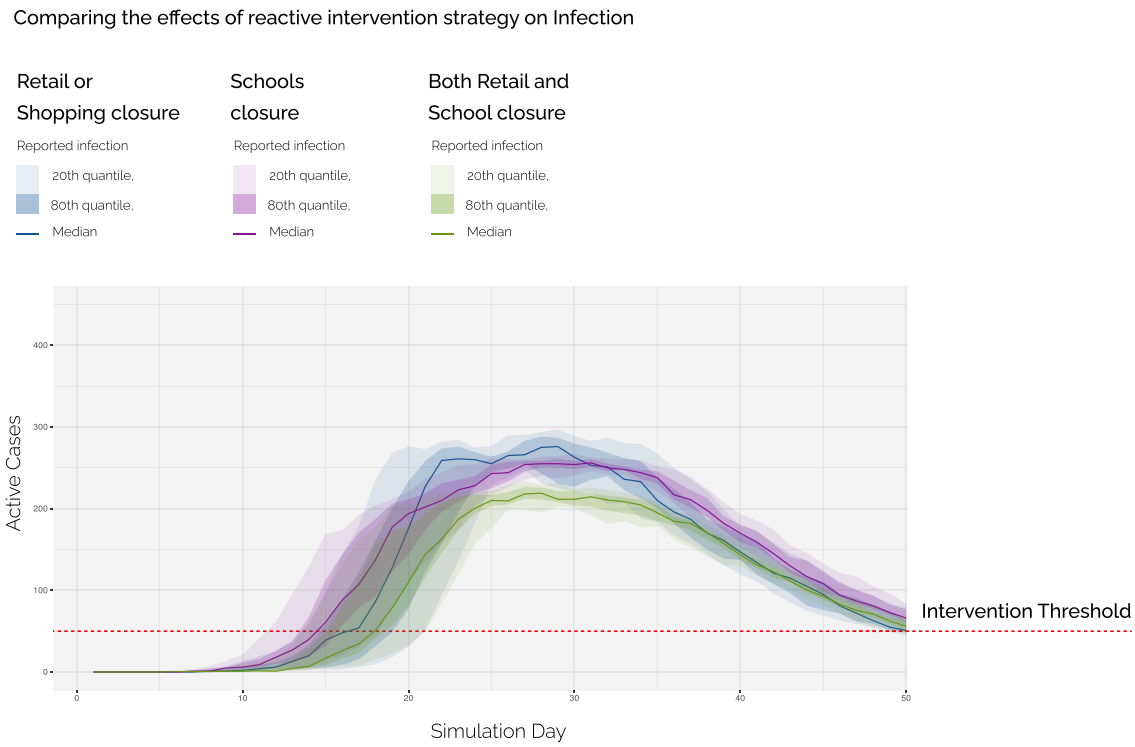


Figure 4.13: Comparing the effects of reactive venue closures on infection risk

series plot of aggregated new infections per day as shown in Figure 4.14.

Plots similar to this figure, provide a jumping-off point for simulating subsequent counterfactual scenarios of venue specific NPI planning and execution and subsequent resource allocation.

4.13 Summary

In this chapter, we describe a simulation framework built upon directed graph based encoding of disease progression model for individuals. We describe how simpler models of disease progression like SIR may be augmented and multiple such models may be composed to simulate realistic infection spread with sufficient detail. We also describe how vaccinations and diagnostic testing as examples of pharmaceutical and non-pharmaceutical contagion spread mitigation may be incorporated as part

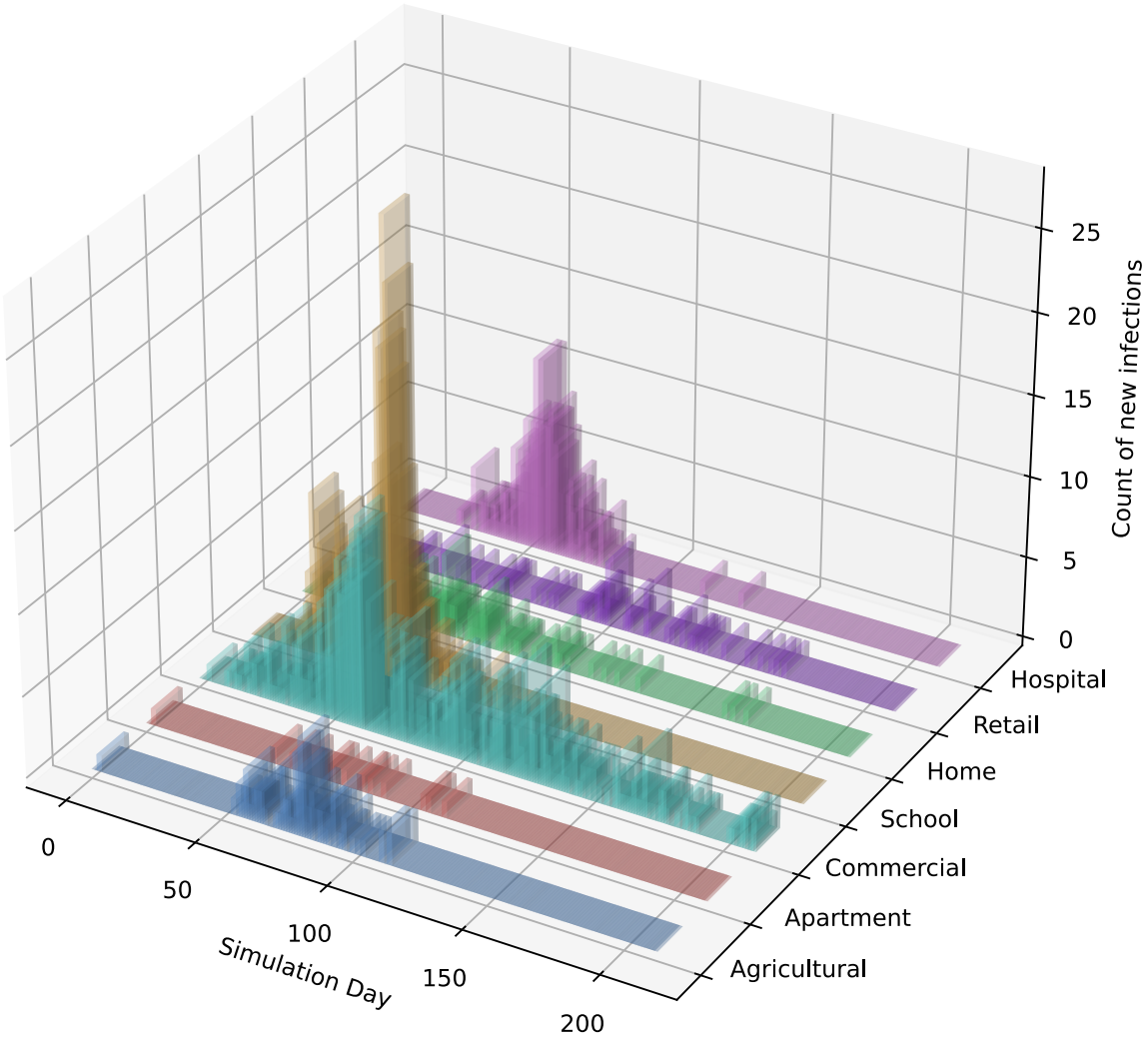


Figure 4.14: Comparing relative infection risks for venue types

of the directed graph encoding with real-world parameters. Next we describe the core simulation loop as pseudo-code and optimised algorithm. Finally, we review a few of example outputs of the simulator visualised as composite time-series plots that offers public health insights. Till now we have considered synthetic contact network of individuals simulated by urban human mobility model, whereas in the next Chapter 5, we propose an efficient spatio-temporal index that may facilitate the generation of individual level contact network from real-world mobility data.

Chapter 5

Spatio-Temporal Indexing

In the previous chapter we detailed the entire workflow of the proposed simulation framework with synthetic contact network generated with a mobility model for humans in an urban area. In this chapter we propose a technique of efficient uncertain trajectory encoding and indexing strategies facilitates generation of contact networks among individuals from real-world mobility data. While synthetic contact networks are useful for geographies with limited to no mobility information and exploring "what-if" scenarios nonetheless, generation of contact notwork from real-world mobility data is expected to increase the accuracy of the contagion model and useful for parameter estimation from historical data.

Based on the surveyed literature and related work, the challenges of this index are due to uncertainty and volume of spatio-temporal observations. In the Chapter 2, Section 2.11 we reviewed GNAT as a viable candidate for spatio-temporal indexing, while in this chapter we utilise GNAT and propose extensions of GNAT for efficient indexing of high volume of uncertain trajectories.

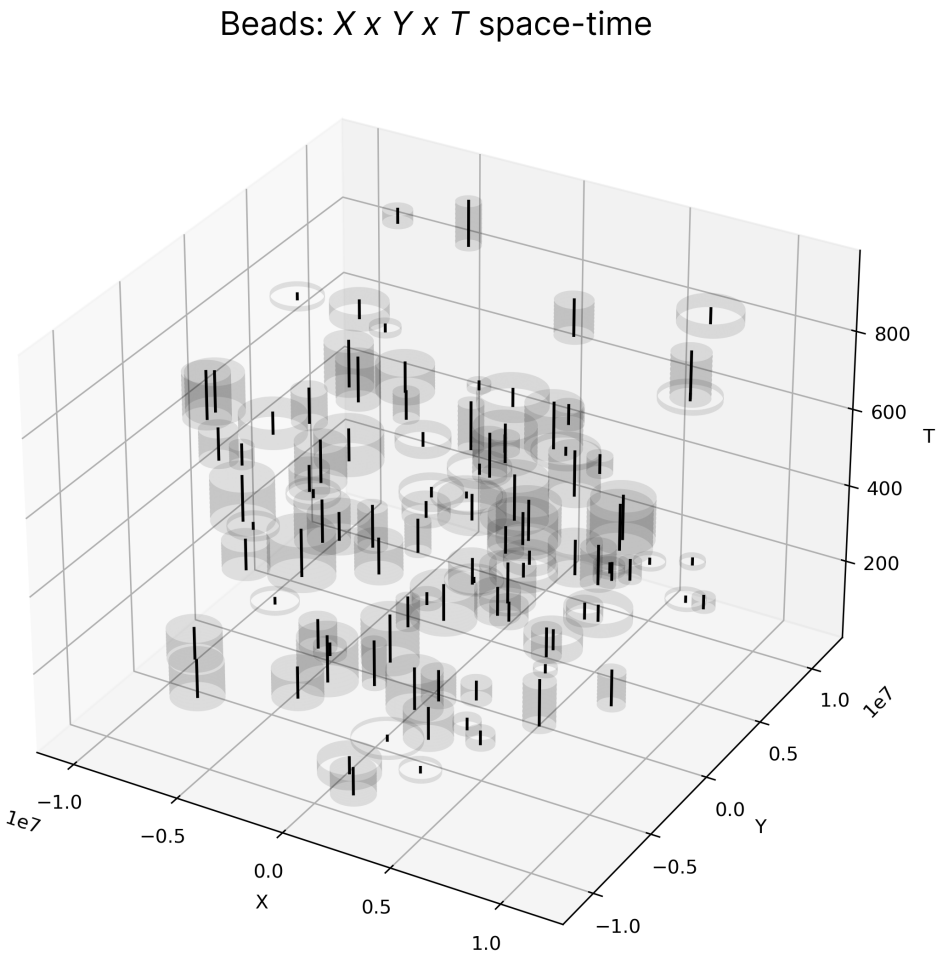


Figure 5.1: Beads in $X \times Y \times T$

5.1 GNAT index for non-metric data

Bead is a shape based modelling technique for uncertain trajectories. Beads are constructed by utilising pairwise sequential observations from a trajectory. For moving objects in real-world 2D space with a maximum possible velocity, beads are elliptical boundary of all possible spatial point the moving object may have occupied in between the two reported observations. When approximated with the semi-major axis, the ellipses are transformed into circles in 2D space. Along with an assumption that the moving object is equally likely to have inhabited at any point of the enclosed circle within the time between the two observations, they may be visualised as space-cylinders in a 2D space with an orthogonal time axis for the purposes of overlap detection. Thus, approximated bead is our choice of shape to model uncertain trajectories, and unlike metric points, they are cylinders on $X \times Y \times T$ space as shown in Figure 5.1, with a symmetrical range on XY plane and an orthogonal temporal range. One strategy to utilise GNAT as is to find overlapping Beads with a virtual Landmark as the space-time query is as follows:

Step 1 For each Bead B_i, construct GNAT with the Bead centre (x_i, y_i) as a 2D metric point.

Step 2 Augment the specified query range q_{range} with the maximum radius r^i_{max} found during Bead construction from trajectories and get list of candidate beads that may or may not overlap with the specified space-time query.

Step 3 Check for space-time overlap with each candidate Bead and the query bounds.

This process is visually demonstrated in Figure 5.2

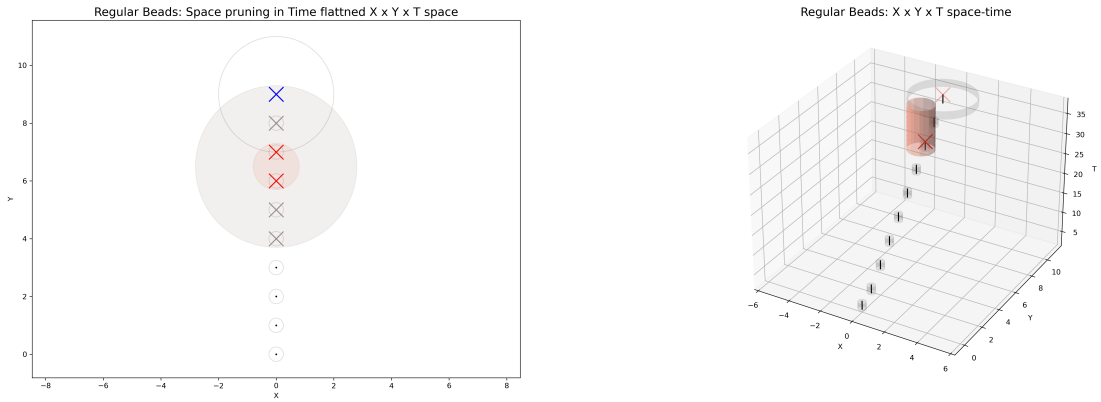


Figure 5.2: On the time-flattened figure on the left, the initial range of the space time query is coloured red. The gray circle is the query range augment by r^i_{max} which produces 6 candidate Beads to check for overlap to find the actual result. The figure on the right is visualising the final result in 3D space for clarity.

5.2 Extending GNAT for uncertain trajectory

Zhang et al. [98] extended a metric-index for M-tree [18] by including the radius of a bead as part of the data point and including it in the metric distance function for space based pruning and including a range-tree, proposed by Bentley et al. [8] for time based pruning in a single traversal of the resultant tree. Taking inspiration from how they extended a metric-point indexing structure to account for both symmetric spatial range and an orthogonal time range, we propose the following extension to classic GNAT as Uncertain Trajectory GNAT (UT-GNAT).

5.2.1 UT-GNAT node structure

The following parameters are sufficient for tree construction and pruning both spatial and temporal ranges.

- **Bead** : Contains full information of an Uncertain Trajectory segment:
  - m_{id} : moving object id
  - x, y, $radius$: spatial range of $radius$, centred on (x, y)

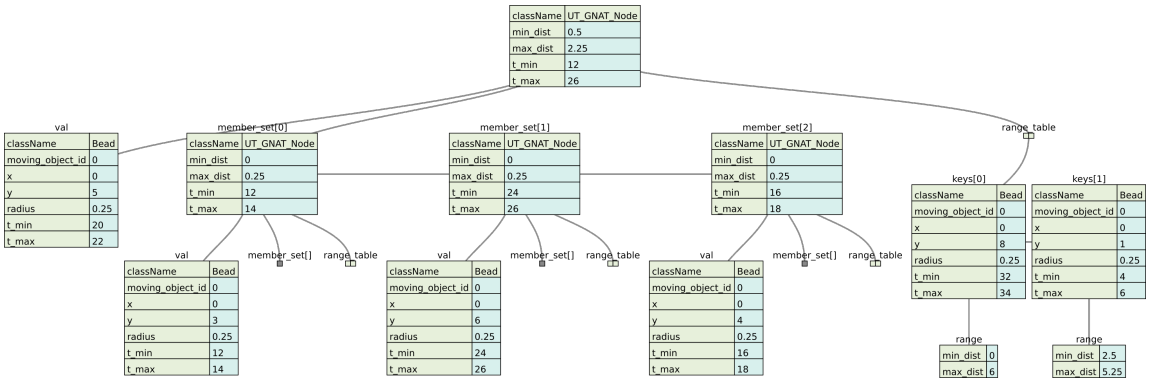


Figure 5.3: Single UT-GNAT node visualised with JSON tree visualiser [47]

  - t_{start}, t_{end} : temporal range

- min_dist : minimum distance to its member set
- max_dist : maximum distance to its member set
- *t_max : maximum of all t_ends of its member set
- *t_min : minimum of all t_starts of its member set
- $member_set$: set of member UT-GNAT nodes
- $range_table$: set of member UT-GNAT nodes

The fields marked with asterisk are introduced like that of a UTM-tree and, are used to maintain a temporal range list for time based pruning. The membership of Beads and UT-GNAT node, and their attributes is shown in Figure 5.3.

5.2.2 Distance function for symmetric range in metric space

Since we are dealing with distances between symmetric ranges of space, the distances between to Beads, B_1 and B_2 may be defined as equations 5.1, 5.2 and 5.3 below and illustrated in Figure 5.4:

$$c2c_dist(B_1, B_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5.1}$$

$$min_dist(B_1, B_2) = max(0, c2c_dist(B_1, B2) - r_1 - r_2) \tag{5.2}$$

$$max_dist(B_1, B_2) = c2c_dist(B_1, B2) + r_2 \tag{5.3}$$

Here, *min_dist* is a stand-in for overlap detection and therefore has the "symmetry" property, i.e.:

$min_dist(B_1, B_2) = min_dist(B_2, B_1)$

But it, does not follow "triangle inequality" for all B_1, B_2.

Therefore, we need *max_dist* which does follow "triangle-inequality":

$max_dist(B_1, B_2) \geq max_dist(B_1, B_3) + max_dist(B_2, B_3)$.

This is true by construction as, $r_2, r_3 \geq 0$ and $c2c_dist(B_1, B_2)$ is Euclidean distance between (x_1, y_1) and (x_2, y_2).

Please note, *min_dist* does not strictly have the "identity" property, since any two overlapping pair of *Beads* yields a 0 and neither does *max_dist* as, $max_dist(B_1, B_2) > 0$ for all $r_2 > 0$.

For split point selection and membership assignment, we utilise the *min_dist* between UT-GNAT nodes and construct the UT-GNAT. And, utilise *max_dist* for relative range based pruning and as a stand-in for maintaining minimum bounding circle for *member_set* of a *Bead*. Thereby making sure during each operation, only one kind of distance calculation is performed.

5.2.3 Doubly ranged search query and space-time pruning

For each space pruning step, we add the corresponding temporal pruning for standard nodes and during relative spatial range pruning using range tables as detailed in algorithm 5

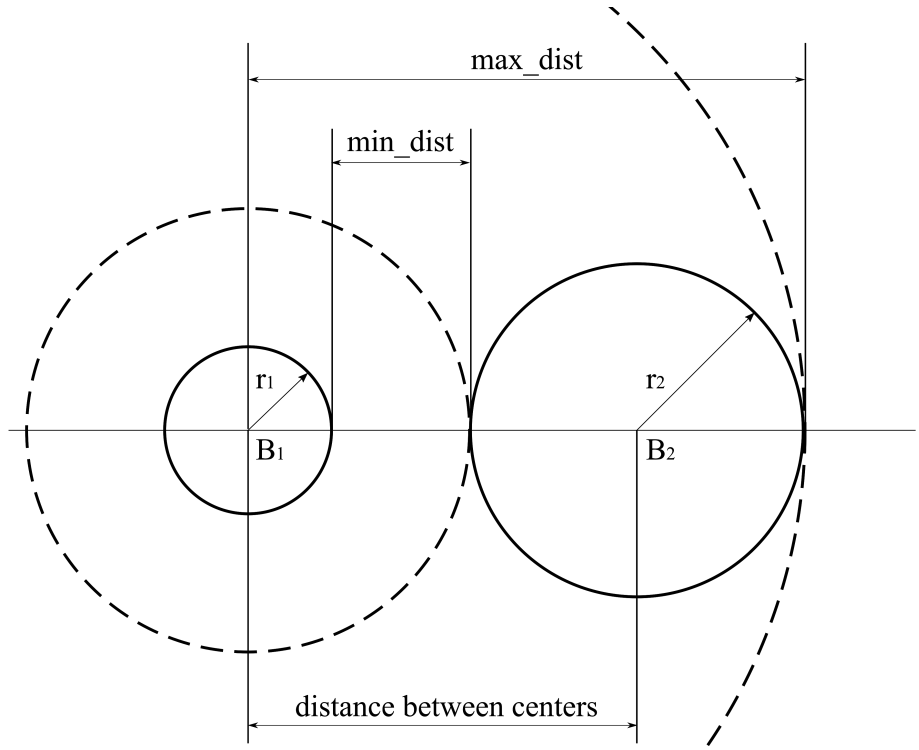


Figure 5.4: Max and min spatial distances between two Beads

Algorithm 5 Recursive Range Search - by Brin [12], extended to Double Range Search

Input: $P, query$
Output: R : set of overlapping Beads

```
Let P be the list of root level UT-GNAT nodes
query contains both spatial and temporal bound to search
Randomly select a split point p ∈ P
Check if p spatio-temporally overlaps with query, then add to R
d ← dist(query, p)
r ← query.spatial_range
relative_range ← [d − r, d + r]
t ← query.temporal_range
for q ∈ P − p do
    if relative_range ∩ range(p, D_q) = φ then
        Remove q from P                                  ▷ Space Pruning
    end if
    if t ∩ [q.t_min, q.t_max] = φ then
        Remove q from P                                  ▷ Time Pruning
    end if
end for
Double Range Search for each D_p for p left in P
```

5.3 Encoding GNAT on regular dataframe as data-store

As a precursor step to encode UT-GNAT on distributed data-frame, we first implement it on pandas [74] dataframe which allows for complex column types.

5.3.1 Column encoding for pandas dataframe

Each of the fields UT-GNAT as listed in Section 5.2.1 are assigned a column build from basic data types except for the column equialents for $member_set$ and $range_table$, and are defined as pandas equivalent of Python's [89] set and Python's dictionary respectively.

5.4 Encoding GNAT on distributed dataframe as data-store

Petersohn et al. [76] provides a drop-in distributed dataframe system as a replacement for Pandas dataframe named Modin. Although their contributions includes formalising dataframe decomposition operations, providing multi-operator optimisation, decoupling logical and physical orderings to name a few. For our context, we utilise their distributed dataframe as a data-store and PySpark [25] as the first step to build towards a fully distributed UT-GNAT.

5.4.1 Unpacking complex data type to columns for Modin dataframe

Due to message passing data shape constraints inherent to Modin system, we had to manually flatten the the column equivalent of $range_table$ from a dictionary to $3(k-1)$

columns for storing *split_sibling_row_id*, *max_dist*, *min_dist* for a maximum possible $k-1$ *split_siblings*, where k is the maximum degree or arity of the UT-GNAT.

5.4.2 Equivalence with Python object based implementation

The correctness of both dataframe based implementations was evaluated by creating a corresponding base python UT-GNAT from the same underlying uncertain trajectory data and comparing results of the same query executed on all three indexes.

5.5 Experimental evaluation

All four UT-GNAT and a GNAT data-structure were written for Python 3 standard and executed on an Intel(R) Core(TM) i7-10875H CPU @ 2.3 GHz running Ubuntu 20.04.2 LTS x86_64-bit OS with 32 GB RAM. Python libraries utilised for dataframe systems are pandas 1.4.3 and modin-dask 0.15.2. For visualising results, Python libraries matplotlib 3.5.1 [45] and cartopy 0.18.0 [64] were utilised as well. A synthetic store of uncertain trajectory was generated for testing increasing data volumes from 1K to 1M Beads. A full list of experiment parameters are listed in Table 5.1.

Table 5.1: Parameter list for initial UT-GNAT experiments

Parameter	**Values**
Type of UTGNAT implementation	base python, row encoded in Pandas, row encoded in Modin, row referenced in Pandas, row referenced in Modin, row referenced in PySpark
Degree of tree	5, 12, 100
Number of uncertain trajectory segments	1131, 5000, 20483, 248728, 1000000
Number of moving objects	10, 4065
Query selectivity	$\sim$ 65%

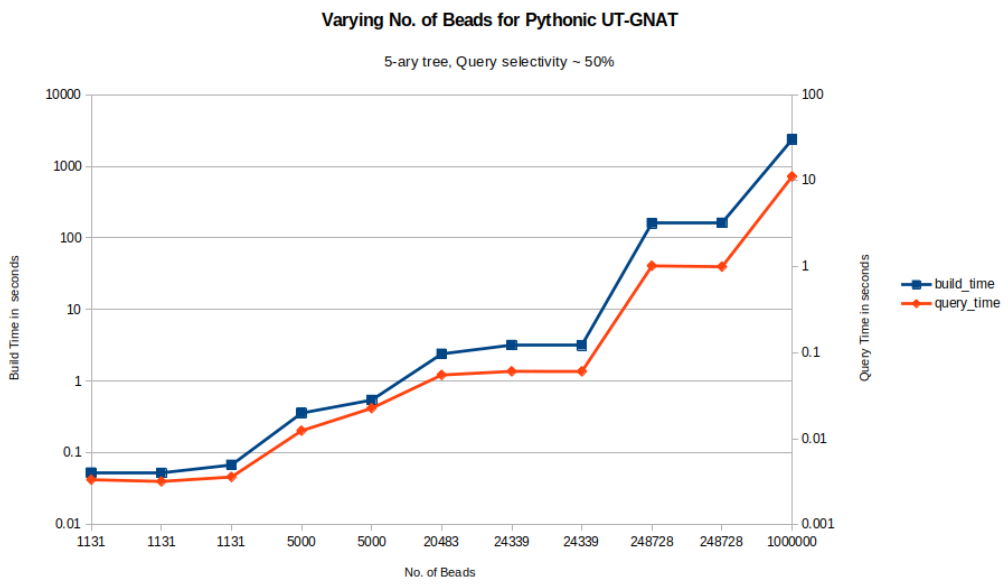


Figure 5.5: Time complexity of (Pythonic) UT-GNAT Please note that there are separate logarithmic time axes for building and querying on the left and right side respectively

5.5.1 Single UT-GNAT index

A UT-GNAT for 1M Beads takes about 2404.8 seconds ($\sim$ 40 minutes) to create and a range query of $\sim$ 50% selectivity takes 11.176 seconds to return all the overlapping Beads. Keeping the degree 5 and query selectivity $\sim$ 50%, the build time and query time follows a polynomial trend as expected, illustrated in Figure 5.5.

5.5.2 Performance loss due to inefficient encoding

When comparing performance for the same building condition, the present pandas dataframe implementation is about 50$\times$ slower and the distributed dataframe is 1600$\times$ slower than base Python UT-GNAT. There is a minor improvement in query execution time of about 25x and 3.3x for singular and distributed dataframe as datastore respectively, with the index data structure without mutating the dataframe. This clearly proves that the naive columnar encoded or referenced UT-GNAT nodes is although equivalent but is not a valid strategy for implementing an efficient index.

This is a result of the number of individual access or collects of data-points during both tree creation and pruning instead of batch accesses.

5.6 Efficiency gain by parallel and distributed indices

Next we compare efficiency gain by splitting up the set of points into multiple indices and execute queries in parallel. The following three splitting strategies were compared:

1. **Clustered**: Trajectory segments closest to a split point were assigned to the same index. Similar to member set generation for GNAT.

2. **De-clustered**: Trajectory segment that would have been part of the same partition in the clustered strategy are allocated across all available partitions in round-robin fashion.

3. **Random**: Random allocation of trajectories for comparison.

5.6.1 Parallel: Implementation and evaluation

Python implementation of UT-GNAT was extended for processing of query in parallel utilising multiprocessing package. Synthetic trajectory segments were generated, with a uniform random distribution of equal spatial bounds from from $\sim$ 1K segments to 1M segments. Two cases of query type, a high selective query of $\sim$ 60% and another low selective query of $\sim$ 0.1% were executed varying the no. of parallel indices from 2 to 16 on an Intel(R) Core(TM) i7-10875H CPU @ 2.3 GHz with 16 logical cores available for parallelisation running Ubuntu 20.04.2 LTS x86_64-bit OS with 32 GB RAM.

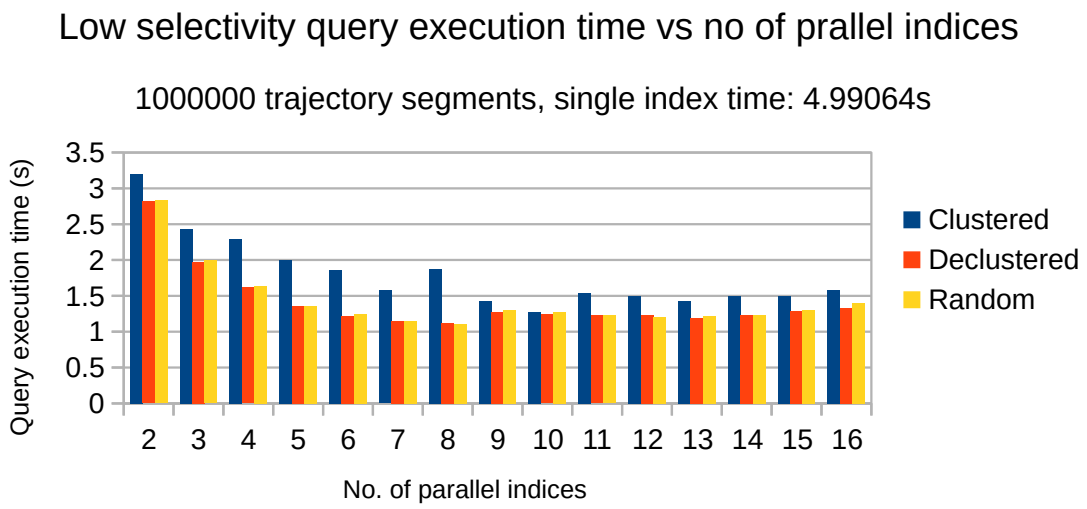


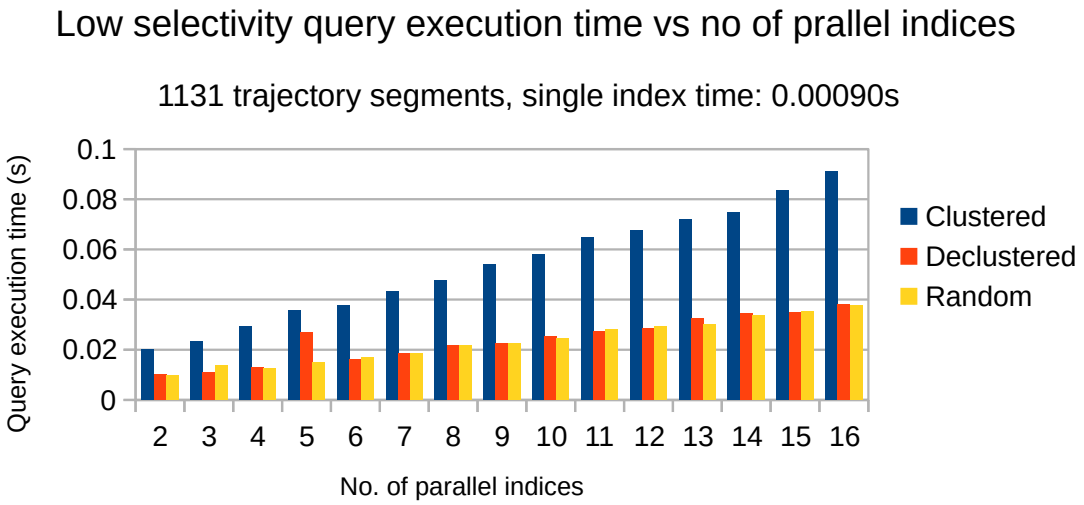
Figure 5.6: Low selectivity, parallel query on 1000000 trajectory segments

5.6.2 Parallel: Results

Efficiency gain outweighs integration cost from 50K trajectory segments and beyond as shown in Figures 5.7 and 5.9. For the full list of results please refer to Appendix B, Section B.1. From Figures 5.6 and 5.8, it is clear that, **clustering** strategy works better for low selectivity query types and **de-clustering** strategy works for both low and high selectivity with a significant improvement over clustering strategy high selectivity query type. For high selectivity query types, de-clustered indices may achieve more than 2× improvement over a single index with just 2 partitions. However, for low selectivity queries, de-clustered indices need at least 3 partitions to achieve ∼2× improvement. The random strategy performs as well as intentional clustering as a result of uniform random test data generation. Similar results may be expected for the clustering strategies for a distributed uncertain trajectory store.

5.6.3 Distributed: Implementation and evaluation

Python implementation of UT-GNAT was further extended for processing of query in a distributed environment utilising Dask [23]. Specifically, the *Actor* API of Dask,



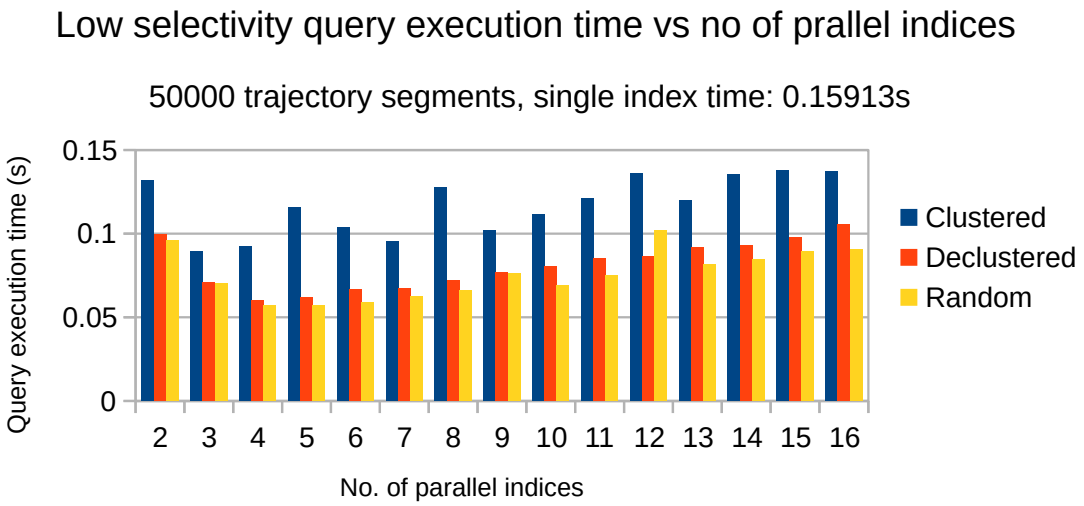


Figure 5.7: Comparing effects of increasing data volume on execution time of parallel query with low selectivity

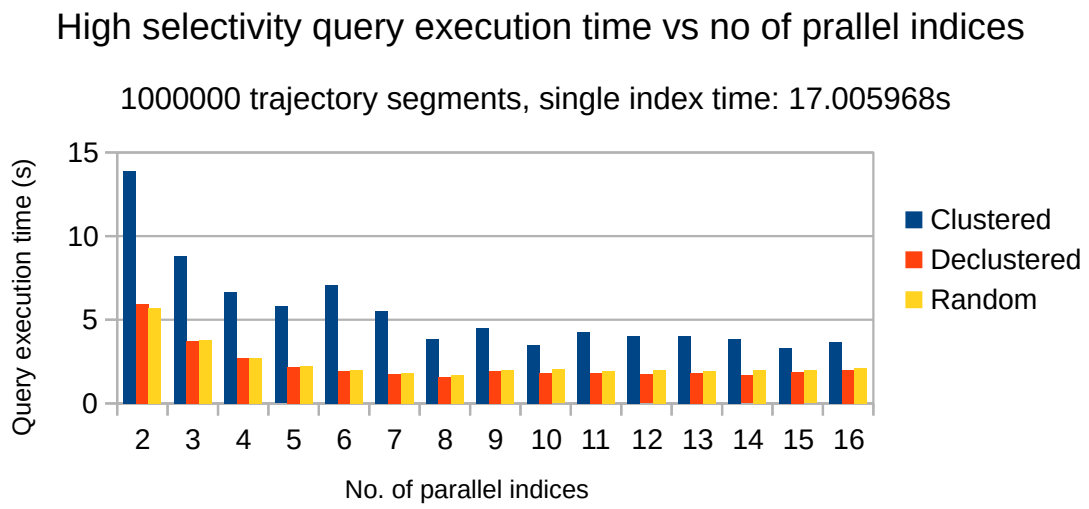


Figure 5.8: High selectivity, parallel query on 1000000 trajectory segments

which enables stateful computation at worker nodes.

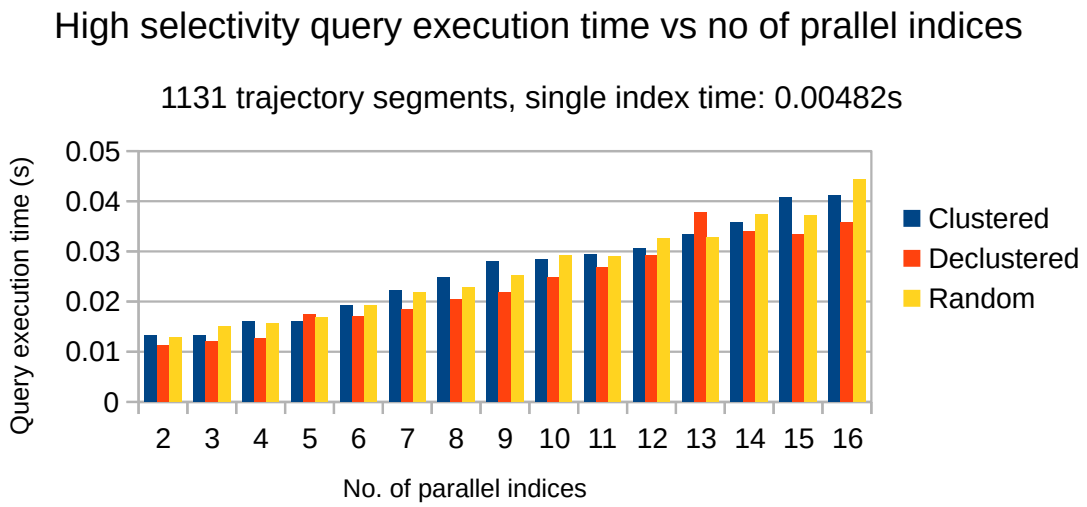
A complete implementation of a UT-GNAT in a distributed context would require the following three steps:

Step 1 Data ingestion of uncertain trajectory segments into a cluster of worker nodes.

Step 2 Partitioning and shuffling of the uncertain trajectories optimally among the worker nodes.

Step 3 Given a cluster of partitioned trajectory segments creation of indices on each worker node for simultaneous querying.

In this experiment, we only focus on Step 3, with worker nodes already set up with their own partitions of the full uncertain trajectory segment store. Similar to previous experiments, the synthetic trajectory segments were generated, with a uniform random distribution of equal spatial bounds of 10K, 20K, 50K and 100K segments and partitioned with the three strategies listed in Section 5.6. Two cases of query type, a high selective query of $\sim 60\%$ and another low selective query of $\sim$



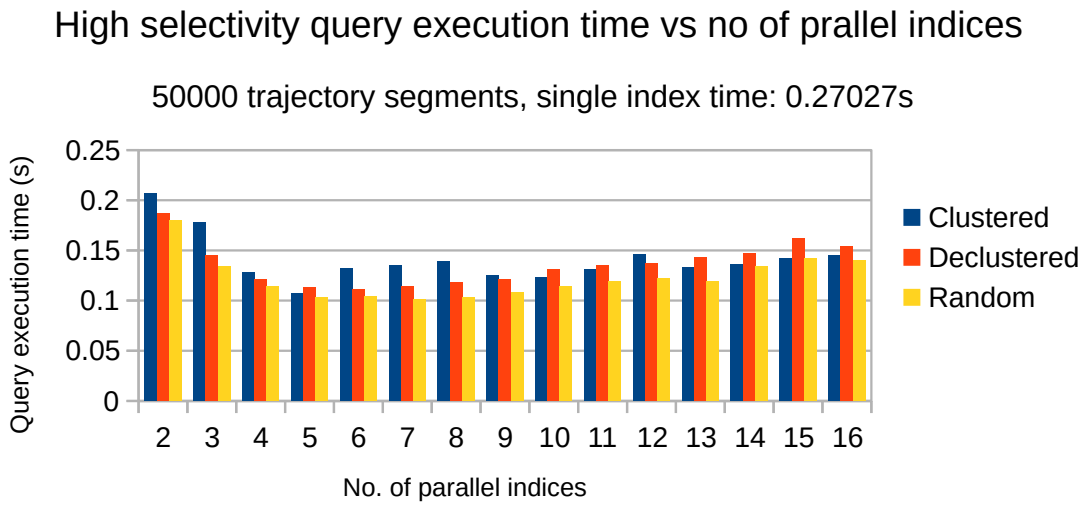


Figure 5.9: Comparing effects of increasing data volume on execution time of parallel query with high selectivity

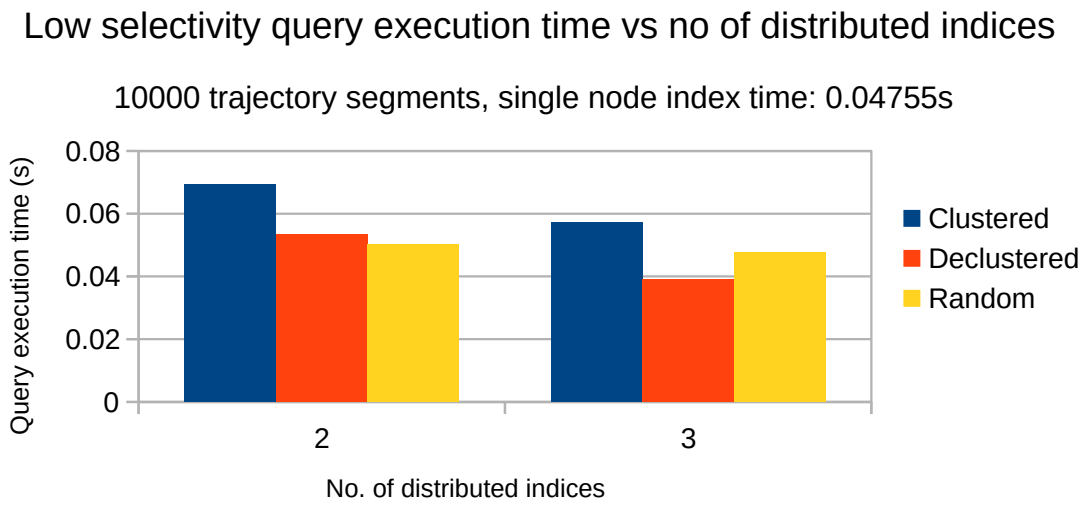
0.1% were executed on a 2 worker node cluster and 3 worker node cluster. Please note, to evaluate worst case time complexity, the network cost of transferring all the points of query results were taken into account while measuring the total cost. Each worker node was created on a machine with an Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz, running Ubuntu 19.04 x86_64-bit OS with 8 GB RAM. The machines were networked via wired Ethernet TP-LINK(R) TL-SG1005D Gigabit Switch.

5.6.4 Distributed: Results

Efficiency gain of distributed partitioning only outweighs network cost from for low selectivity queries as shown in Figures 5.10 and 5.11. For the full list of results please refer to Appendix B, Section B.2. This discrepancy in efficiency gain stems from the worst case evaluation of total query evaluation time that includes both query execution time at worker node and query result transfer time from worker nodes to client node. The query result transfer consumes 93.73% of the total execution time for low selectivity query and 99.2% of the total execution time for high selectivity query.

5.7 Summary

In this chapter we extend a metric-space index GNAT to accommodate non-metric uncertain trajectory indexing named UT-GNAT with a pair of distance functions. We implement the proposed index in Python and run extensive experiments to test single index performance. Next we propose efficient partitioning schemes compatible with this data structure and experiment to confirm the efficiency gain. In the next chapter, we conclude this thesis with discussions and possible future works.



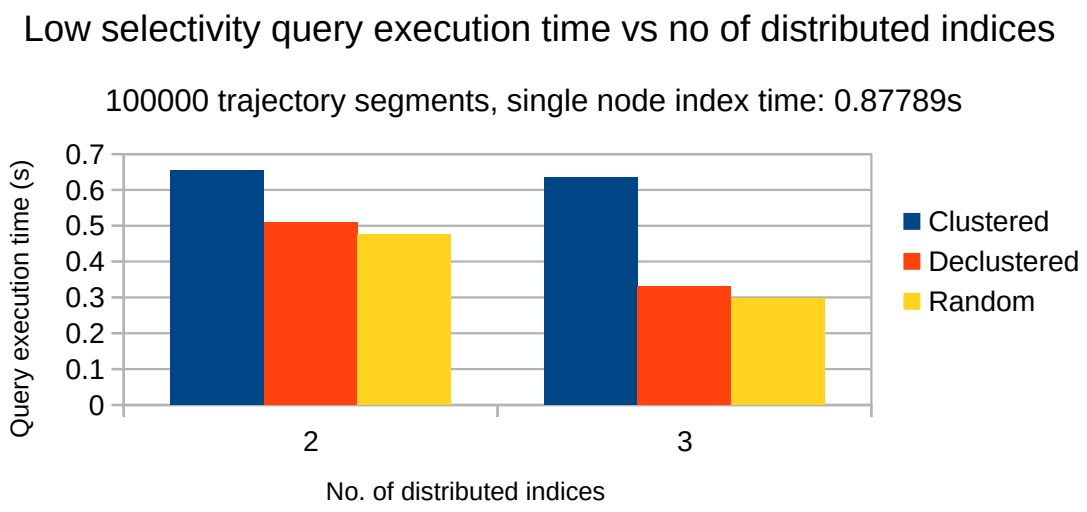
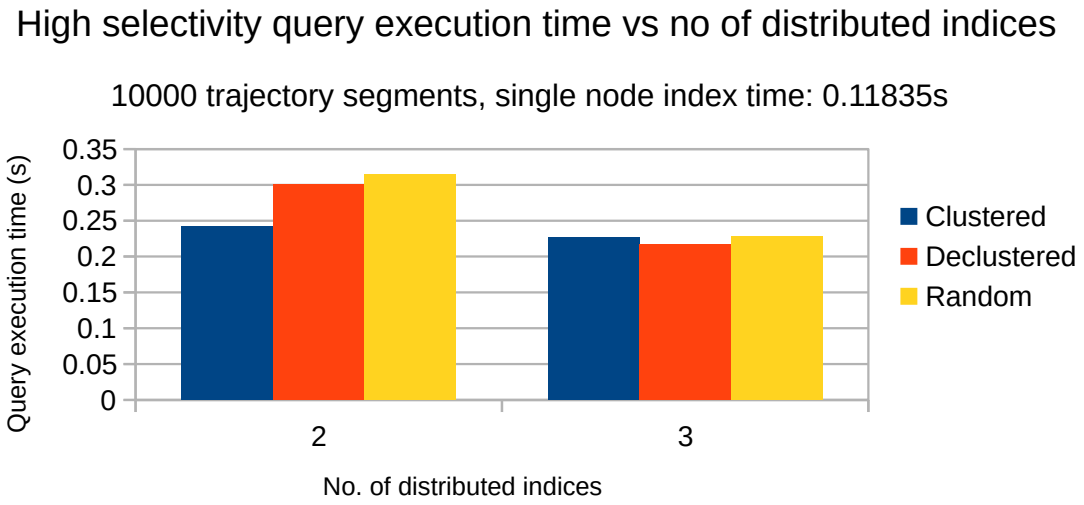


Figure 5.10: Comparing effects of increasing data volume on execution time of distributed query with low selectivity



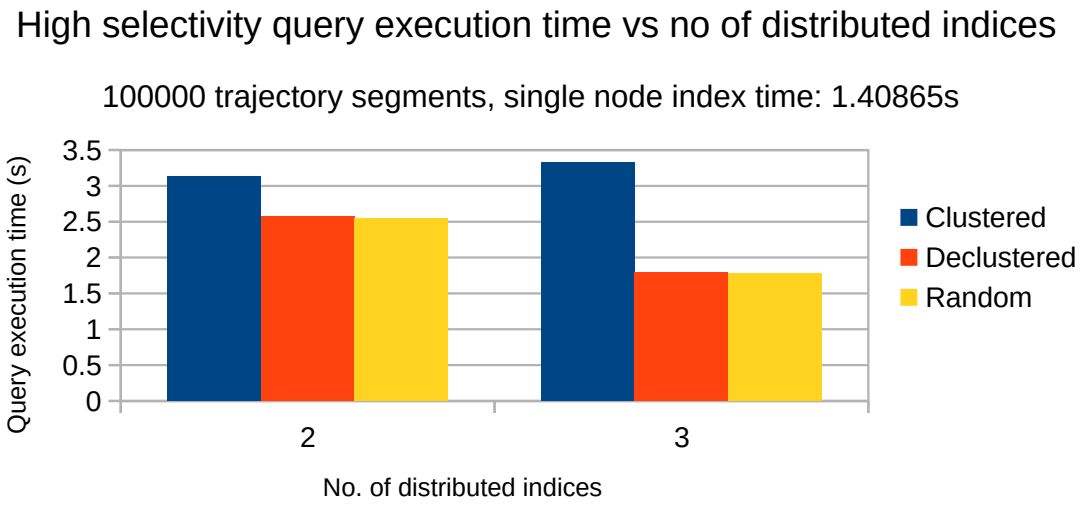


Figure 5.11: Comparing effects of increasing data volume on execution time of distributed query with high selectivity

Chapter 6

Conclusion and Discussion

In this chapter we discuss the advantages and limitations of the of the two techniques implemented as part of this thesis namely, the individual level contagion simulation (framework) and efficient indexing of uncertain trajectories for contact network generation. We also mention several future research directions.

6.1 Individual level contagion simulation

The approach of modelling contagion spread at an individual level ties mechanistic viral load transfer models with human mobility over a geography, preserving all observable heterogeneity of age, activity, preexisting health risk factors, disposable income and so on. Therefore, this has potential to provides public health policymakers with granular control over forecasting changes as well as counterfactual scenario simulations for answering "what-if" questions, like "How would closing shopping centres for a period of time change the hospitalisation rate?", "How would testing strategy change affect overall disease progression?", "How might we distribute vaccine in a community to minimise casualties?".

The framework takes the approach of merging multiple finite-state-machines into a single directed graph and its transition matrix that offers a simple single data

structure to manage multiple sub-models involved in simulating contagion spread, its detection and mitigation.

At the time of writing this thesis, a similar network based stochastic contagion spread modelling framework and implementation, that enables plugging in different disease progression models with little to no adjustment to simulation code with parameterised sub-models for vaccination, diagnostic testing and contact tracing was not found while searching through relevant literature. As many contagions propagate through transfer of viral load among individuals of a population, we believe that individual simulation approaches like that of this thesis are uniquely positioned to accurately describe the disease dynamics as emergent properties of spatio-temporal context, health factors and behaviours of individuals.

6.1.1 Limitations and possible extensions

- The contagion simulation framework in this thesis treats the participating agents as vertices in a contact network graph and hence the disease progression model does not affect mobility of an agent. Although, minimal rule based behaviour is simulated, a full feedback loop of status of disease affecting mobility decisions would be an extension worth investigating.

- The simulation geography and demography have been limited to small communities of less than 10K individuals. Multiple avenues of scaling-up remains to be investigated including parallel processing, GPU based optimisation and distributed computation.

- The directed graph based approach merges both fixed time and stochastic time based intervals. This results in challenges re-constructing an ODE equivalent for the implemented model.

- The synthetic contact network graphs generated in this simulator framework

provides granularity of a pair of individuals time-overlap at a point of of interest and the activities they were engaged in. However, it does not provide mobility information within a point of interest.

- The underlying mobility model utilised for generating synthetic contact network for small communities in New Brunswick province have only been validated for Greater Toronto Area, Canada.

- The focus of this framework have been modelling contagion spreading from person to person. This may be extended for zoonotic contagions by augmenting spatial-context of infection events with the (disease carrying) vector population densities or similar proxy metric and utilisation of relevant infection probability function. For example mosquito density for diseases like malaria or dengue.

6.2 Efficient uncertain trajectory indexing

UT-GNAT proposed in this thesis is a viable candidate for indexing high volume of uncertain trajectory data as an alternative to UTM-tree. UT-GNAT combined with the data-set splitting strategies provide efficient parallel processing of selection queries and shows promise as shuffling algorithm for distributed context. At the time of writing this thesis, similar extensions to GNAT to accommodate for uncertain trajectory indexing or subsequent partitioning strategy have not been found.

6.2.1 Limitations and possible extensions

- UT-GNAT is based on GNAT, which is one of many Generic Hyperplane Tree family of data structures and there may be other more efficient index candidates.

- The extension of range search proposed and implemented is based on basic

recursive search. This may be extended with alternatives like Approximating Eliminating Search Algorithm (AESA) to begin with.

- The prototype implementation in Python makes it challenging for analysing memory overheads.

6.3 Future work

Beyond the extensions mentioned in Section 6.1.1 and Section 6.2.1, we list the following possibilities for future work:

- Multi-level contagion spread models of smaller communities to hierarchically scale to larger interconnected geographies and demographics.
- Inclusion of non-individual entities as part of the contagion model to accommodate for indirect observations of contagion spread. For example leveraging the waste water based disease surveillance, proposed by Hillary et al. [42].
- Comparison of disease modelling accuracy and efficacy for synthetic and real-world contact network of a given geography.

Bibliography

[1] Matthew Abueg, Robert Hinch, Neo Wu, Luyang Liu, William Probert, Austin Wu, Paul Eastham, Yusef Shafi, Matt Rosencrantz, Michael Dikovsky, and others. 2021. Modeling the effect of exposure notification and non-pharmaceutical interventions on COVID-19 transmission in Washington state. *NPJ digital medicine* 4, 1 (2021), 1–10.

[2] Revolution Analytics and Steve Weston. 2020. iterators: Provides Iterator Construct. (2020). https://CRAN.R-project.org/package=iterators R package version 1.0.13.

[3] Chrysovalantis Anastasiou, Constantinos Costa, Panos K Chrysanthis, Cyrus Shahabi, and Demetrios Zeinalipour-Yazti. 2021. ASTRO: Reducing COVID-19 Exposure through Contact Prediction and Avoidance. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 8, 2 (2021), 1–31.

[4] W Brian Arthur, Steven N Durlauf, and David A Lane. 1997. *The economy as a complex evolving system II*. Vol. 27. Westview Press.

[5] John Baez, Xiaoyan Li, Sophie Libkind, Nathaniel Osgood, and Evan Patterson. 2022. Compositional Modeling with Stock and Flow Diagrams. *arXiv preprint arXiv:2205.08373* (2022).

[6] Raul Bagni, Roberto Berchi, and Pasquale Cariello. 2002. A comparison of simulation models applied to epidemics. *Journal of Artificial Societies and Social Simulation* 5, 3 (2002).

[7] Douglas Bates and Martin Maechler. 2019. Matrix: Sparse and Dense Matrix Classes and Methods. (2019). https://CRAN.R-project.org/package=Matrix R package version 1.2-18.

[8] Jon Louis Bentley and Jerome H Friedman. 1979. Data structures for range searching. *ACM Computing Surveys (CSUR)* 11, 4 (1979), 397–409.

[9] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. Julia: A fresh approach to numerical computing. *SIAM Review* 59, 1 (2017), 65–98. DOI:http://dx.doi.org/10.1137/141000671

[10] Lianlian Bian, Fan Gao, Jialu Zhang, Qian He, Qunying Mao, Miao Xu, and Zhenglun Liang. 2021. Effects of SARS-CoV-2 variants on vaccine efficacy and response strategies. *Expert review of vaccines* 20, 4 (2021), 365–373.

[11] Fred Brauer. 2017. Mathematical epidemiology: Past, present, and future. *Infectious Disease Modelling* 2, 2 (2017), 113–127.

[12] Sergey Brin. 1995. Near neighbor search in large metric spaces. (1995).

[13] Daniel G Brown, Scott E Page, Rick Riolo, and William Rand. 2004. Agent-based and analytical modeling to evaluate the effectiveness of greenbelts. *Environmental Modelling & Software* 19, 12 (2004), 1097–1109.

[14] Joseph D Challenger, Cher Y Foo, Yue Wu, Ada WC Yan, Mahdi Moradi Marjaneh, Felicity Liew, Ryan S Thwaites, Lucy C Okell, and Aubrey J Cunnington. 2022. Modelling upper respiratory viral load dynamics of SARS-CoV-2. *BMC medicine* 20, 1 (2022), 1–20.

[15] Lu Chen, Yunjun Gao, Xuan Song, Zheng Li, Yifan Zhu, Xiaoye Miao, and Christian S Jensen. 2020. Indexing metric spaces for exact similarity search. *ACM Computing Surveys (CSUR)* (2020).

[16] Vincent Chi-Chung Cheng, Shuk-Ching Wong, Vivien Wai-Man Chuang, Simon Yung-Chun So, Jonathan Hon-Kwan Chen, Siddharth Sridhar, Kelvin Kai-Wang To, Jasper Fuk-Woo Chan, Ivan Fan-Ngai Hung, Pak-Leung Ho, and others. 2020. The role of community-wide wearing of face mask for control of coronavirus disease 2019 (COVID-19) epidemic due to SARS-CoV-2. *Journal of Infection* 81, 1 (2020), 107–114.

[17] Wen-Hao Chiang, Xueying Liu, and George Mohler. 2022. Hawkes process modeling of COVID-19 with mobility leading indicators and spatial covariates. *International journal of forecasting* 38, 2 (2022), 505–520.

[18] Paolo Ciaccia, Marco Patella, and Pavel Zezula. 1997. M-tree: An efficient access method for similarity search in metric spaces. In *Vldb*, Vol. 97. 426–435.

[19] Fabrizio Cicala, Weicheng Wang, Tianhao Wang, Ninghui Li, Elisa Bertino, Faming Liang, and Yang Yang. 2021. Pure: A framework for analyzing proximity-based contact tracing protocols. *ACM Computing Surveys (CSUR)* 55, 1 (2021), 1–36.

[20] PixiJS contributors. 2013. PixiJS. (2013). https://github.com/pixijs/pixijs Online; accessed 18-Oct-2022;.

[21] Gabor Csardi and Tamas Nepusz. 2006. The igraph software package for complex network research. *InterJournal* Complex Systems (2006), 1695. https://igraph.org

[22] Domenico Cucinotta and Maurizio Vanelli. 2020. WHO declares COVID-19 a pandemic. *Acta Biomed.* 91, 1 (March 2020), 157–160.

[23] Dask Development Team. 2016. *Dask: Library for dynamic task scheduling.* https://dask.org

[24] Andreza Aruska de Souza Santos, Darlan da Silva Candido, William Marciel de Souza, Lewis Buss, Sabrina L Li, Rafael HM Pereira, Chieh-Hsi Wu, Ester C Sabino, and Nuno R Faria. 2021. Dataset on SARS-CoV-2 non-pharmaceutical interventions in Brazilian municipalities. *Scientific data* 8, 1 (2021), 1–6.

[25] Spark Developers. 2016. Pyspark package. (2016). http://spark.apache.org/docs/2.2.0/api/python/pyspark.html

[26] Jacqueline Dinnes, Pawana Sharma, Sarah Berhane, Susanna S van Wyk, Nicholas Nyaaba, Julie Domen, Melissa Taylor, Jane Cunningham, Clare Davenport, Sabine Dittrich, and others. 2022. Rapid, point-of-care antigen tests for diagnosis of SARS-CoV-2 infection. *Cochrane Database of Systematic Reviews* 7 (2022).

[27] Ensheng Dong, Hongru Du, and Lauren Gardner. 2020. An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases* 20, 5 (2020), 533–534. DOI:http://dx.doi.org/https://doi.org/10.1016/S1473-3099(20)30120-1

[28] Ali Dorri, Salil S. Kanhere, and Raja Jurdak. 2018. Multi-Agent Systems: A Survey. *IEEE Access* 6 (2018), 28573–28593. DOI:http://dx.doi.org/10.1109/ACCESS.2018.2831228

[29] Josh A Firth, Joel Hellewell, Petra Klepac, Stephen Kissler, Adam J Kucharski, and Lewis G Spurgin. 2020. Using a real-world network to model localized COVID-19 control strategies. *Nature medicine* 26, 10 (2020), 1616–1622.

[30] Richard G FitzJohn, Edward S Knock, Lilith K Whittles, Pablo N Perez-Guzman, Sangeeta Bhatia, Fernando Guntoro, Oliver J Watson, Charles Whit-

taker, Neil M Ferguson, Anne Cori, and others. 2020. Reproducible parallel inference and simulation of stochastic state space models using odin, dust, and mcstate. *Wellcome Open Research* 5 (2020).

[31] Seth Flaxman, Swapnil Mishra, Axel Gandy, H Juliette T Unwin, Thomas A Mellan, Helen Coupland, Charles Whittaker, Harrison Zhu, Tresnia Berah, Jeffrey W Eaton, and others. 2020. Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe. *Nature* 584, 7820 (2020), 257–261.

[32] Virginia A Folcik, Gary C An, and Charles G Orosz. 2007. The Basic Immune Simulator: an agent-based model to study the interactions between innate and adaptive immunity. *Theoretical Biology and Medical Modelling* 4, 1 (2007), 1–18.

[33] John Geanakoplos, Robert Axtell, J Doyne Farmer, Peter Howitt, Benjamin Conlee, Jonathan Goldstein, Matthew Hendrey, Nathan M Palmer, and Chun-Yi Yang. 2012. Getting at systemic risk via an agent-based model of the housing market. *American Economic Review* 102, 3 (2012), 53–58.

[34] Nigel Gilbert, John C Hawksworth, and Paul A Swinney. 2009. An agent-based model of the English housing market. (2009).

[35] John W Glasser, Zhilan Feng, Saad B Omer, Philip J Smith, and Lance E Rodewald. 2016. The effect of heterogeneity in uptake of the measles, mumps, and rubella vaccine on the potential for outbreaks of measles: a modelling study. *The Lancet Infectious Diseases* 16, 5 (2016), 599–605.

[36] John D Goss-Custard, Niall HK Burton, Nigel A Clark, Peter N Ferns, Selwyn McGrorty, Christopher J Reading, Mark M Rehfisch, Richard A Stillman, Ian Townend, Andrew D West, and others. 2006. Test of a behavior-based

individual-based model: response of shorebird mortality to habitat loss. *Ecological Applications* 16, 6 (2006), 2215–2222.

[37] Canada Government of New Brunswick. 2022. Covid-19 testing. (Apr 2022). https://www2.gnb.ca/content/gnb/en/corporate/promo/covid-19/about-covid-19/testing-tracing.html [Online; accessed 4-July-2022].

[38] Transportation Government of Ontario. 2020. Transportation Tomorrow historical survey data. (Jan 2020). https://open.canada.ca/data/en/dataset/a58c577e-deb8-4091-9ee7-cd5fe790e16f Online; accessed 18-Oct-2022;.

[39] Xi He, Eric HY Lau, Peng Wu, Xilong Deng, Jian Wang, Xinxin Hao, Yiu Chung Lau, Jessica Y Wong, Yujuan Guan, Xinghua Tan, and others. 2020. Temporal dynamics in viral shedding and transmissibility of COVID-19. *Nature medicine* 26, 5 (2020), 672–675.

[40] Niel Hens, Pascal Vranck, and Geert Molenberghs. 2020. The COVID-19 epidemic, its mortality, and the role of non-pharmaceutical interventions. *European Heart Journal: Acute Cardiovascular Care* 9, 3 (2020), 204–208.

[41] R Raymond Hill, RG Carl, and LE Champagne. 2006. Using agent simulation models to examine and investigate search theory against a historical case study. *J Simul* 1, 1 (2006), 29–38.

[42] Luke S Hillary, Shelagh K Malham, James E McDonald, and David L Jones. 2020. Wastewater and public health: the potential of wastewater surveillance for monitoring COVID-19. *Current Opinion in Environmental Science & Health* 17 (2020), 14–20.

[43] Andreas Horni, Kai Nagel, and Kay W Axhausen. 2016. *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, London, GBR.

[44] Elizabeth Hunter, Brian Mac Namee, and John Kelleher. 2018. An open-data-driven agent-based model to simulate infectious disease outbreaks. *PloS one* 13, 12 (2018), e0208775.

[45] J. D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, 3 (2007), 90–95. DOI:http://dx.doi.org/10.1109/MCSE.2007.55

[46] The Black Arcs Inc. 2018. Citisketch. (2018). https://blackarcs.org/#slide2 Online; accessed 18-Oct-2022;.

[47] ivan ivanov. 2014. JSON to Tree Diagram Converter. (2014). https://github.com/ivan111/vtree Online; accessed 10-Nov-2022;.

[48] Jonas L Juul, Kaare Græsbøll, Lasse Engbo Christiansen, and Sune Lehmann. 2021. Fixed-time descriptive statistics underestimate extremes of epidemic curve ensembles. *Nature physics* 17, 1 (2021), 5–8.

[49] Matt J Keeling and Ken TD Eames. 2005. Networks and epidemic models. *Journal of the royal society interface* 2, 4 (2005), 295–307.

[50] Ben Killingley, Alex J Mann, Mariya Kalinova, Alison Boyers, Niluka Goonawardane, Jie Zhou, Kate Lindsell, Samanjit S Hare, Jonathan Brown, Rebecca Frise, and others. 2022. Safety, tolerability and viral kinetics during SARS-CoV-2 human challenge in young adults. *Nature Medicine* 28, 5 (2022), 1031–1041.

[51] Stephen M Kissler, Petra Klepac, Maria Tang, Andrew JK Conlan, and Julia R Gog. 2020. Sparking "The BBC Four Pandemic": Leveraging citizen science and mobile phones to model the spread of disease. *BioRxiv* (2020), 479154.

[52] Petra Klepac, Stephen Kissler, and Julia Gog. 2018. Contagion! the bbc four pandemic–the model behind the documentary. *Epidemics* 24 (2018), 49–59.

[53] Nattawat Klomjit, Mariam Priya Alexander, Fernando C Fervenza, Ziad Zoghby, Arvind Garg, Marie C Hogan, Samih H Nasr, Marwan Abu Minshar, and Ladan Zand. 2021. COVID-19 vaccination and glomerulonephritis. *Kidney international reports* 6, 12 (2021), 2969–2978.

[54] Timothy A Kohler, George J Gumerman, and Robert G Reynolds. 2005. Simulating ancient societies. *Scientific American* 293, 1 (2005), 76–84.

[55] Mirjam E Kretzschmar, Ganna Rozhnova, Martin CJ Bootsma, Michiel van Boven, Janneke HHM van de Wijgert, and Marc JM Bonten. 2020. Impact of delays on effectiveness of contact tracing strategies for COVID-19: a modelling study. *The Lancet Public Health* 5, 8 (2020), e452–e459.

[56] Md Abdul Kuddus and Azizur Rahman. 2021. Analysis of COVID-19 using a modified SLIR model with nonlinear incidence. *Results in Physics* 27 (2021), 104478.

[57] Shengjie Lai, Nick W Ruktanonchai, Liangcai Zhou, Olivia Prosper, Wei Luo, Jessica R Floyd, Amy Wesolowski, Mauricio Santillana, Chi Zhang, Xiangjun Du, and others. 2020. Effect of non-pharmaceutical interventions to contain COVID-19 in China. *nature* 585, 7825 (2020), 410–413.

[58] Ralph Lange, Harald Weinschrott, Lars Geiger, André Blessing, Frank Dürr, Kurt Rothermel, and Hinrich Schütze. 2009. On a generic uncertainty model for position information. In *International Workshop on Quality of Context*. Springer, 76–87.

[59] Maylis Layan, Mayan Gilboa, Tal Gonen, Miki Goldenfeld, Lilac Meltzer, Alessio Andronico, Nathanaël Hozé, Simon Cauchemez, and Gili Regev-Yochay. 2022. Impact of BNT162b2 vaccination and isolation on SARS-CoV-2 transmis-

sion in Israeli households: an observational study. *American journal of epidemiology* 191, 7 (2022), 1224–1234.

[60] C Macal, D Sallach, and M North. 2004. Emergent structures from trust relationships in supply chains. In *Proc. Agent 2004: Conf. on Social Dynamics*. 7–9.

[61] C M Macal and M J North. 2010. Tutorial on agent-based modelling and simulation. *Journal of Simulation* 4, 3 (2010), 151–162. DOI:http://dx.doi.org/10.1057/jos.2010.3

[62] Monia Makhoul, Houssein H Ayoub, Hiam Chemaitelly, Shaheen Seedat, Ghina R Mumtaz, Sarah Al-Omari, and Laith J Abu-Raddad. 2020. Epidemiological impact of SARS-CoV-2 vaccination: Mathematical modeling analyses. *Vaccines* 8, 4 (2020), 668.

[63] Ouassim Manout and Francesco Ciari. 2021. Assessing the role of daily activities and mobility in the spread of COVID-19 in Montreal with an agent-based approach. *Frontiers in Built Environment* (2021), 101.

[64] Met Office. 2010 - 2015. Cartopy: a cartographic python library with a Matplotlib interface. (2010 - 2015). https://scitools.org.uk/cartopy

[65] Eric J Miller and Matthew J Roorda. 2003. Prototype model of household activity-travel scheduling. *Transportation Research Record* 1831, 1 (2003), 114–121.

[66] James Moffat, Josephine Smith, and Susan Witty. 2006. Emergent behaviour: theory and experimentation using the MANA model. *Journal of Applied Mathematics and Decision Sciences* 2006 (2006).

[67] MyHealthNB. 2022. Covid-19 results. (2022). https://myhealth.gnb.ca/faq#covid19pcrtestresults [Online; accessed 4-July-2022].

[68] Michael J North, Charles M Macal, James St Aubin, Prakash Thimmapuram, Mark Bragen, June Hahn, James Karr, Nancy Brigham, Mark E Lacy, and Delaine Hampton. 2010. Multiscale agent-based consumer market modeling. *Complexity* 15, 5 (2010), 37–47.

[69] Beata Nowok, Gillian M. Raab, and Chris Dibben. 2016. synthpop: Bespoke Creation of Synthetic Data in R. *Journal of Statistical Software* 74, 11 (2016), 1–26. DOI:http://dx.doi.org/10.18637/jss.v074.i11

[70] Government of Canada. 2022. statistics canada: canada's national statistical agency. (2022). https://www.statcan.gc.ca/en/start Online; accessed 18-Oct-2022;.

[71] Daniela Olivera Mesa, Alexandra B Hogan, Oliver J Watson, Giovanni D Charles, Katharina Hauck, Azra C Ghani, and Peter Winskill. 2022. Modelling the impact of vaccine hesitancy in prolonging the need for Non-Pharmaceutical Interventions to control the COVID-19 pandemic. *Communications Medicine* 2, 1 (2022), 1–8.

[72] Jeroen Ooms. 2014. The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects. *arXiv:1403.2805 [stat.CO]* (2014). https://arxiv.org/abs/1403.2805

[73] OpenStreetMap contributors. 2017. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org. (2017).

[74] The pandas development team. 2020. pandas-dev/pandas: Pandas. (Feb. 2020). DOI:http://dx.doi.org/10.5281/zenodo.3509134

[75] Shayn M Peirce, Eric J Van Gieson, and Thomas C Skalak. 2004. Multicellular simulation predicts microvascular patterning and in silico tissue assembly. *The FASEB journal* 18, 6 (2004), 731–733.

[76] Devin Petersohn, Dixin Tang, Rehan Durrani, Areg Melik-Adamyan, Joseph E. Gonzalez, Anthony D. Joseph, and Aditya G. Parameswaran. 2021. Flexible Rule-Based Decomposition and Metadata Independence in Modin: A Parallel Dataframe System. *Proc. VLDB Endow.* 15, 3 (nov 2021), 739–751. DOI: http://dx.doi.org/10.14778/3494124.3494152

[77] Dieter Pfoser, Christian S Jensen, Yannis Theodoridis, and others. 2000. Novel approaches to the indexing of moving object trajectories.. In *VLDB*. 395–406.

[78] Emma Pritchard, Philippa C Matthews, Nicole Stoesser, David W Eyre, Owen Gethings, Karina-Doris Vihta, Joel Jones, Thomas House, Harper VanSteen-House, Iain Bell, and others. 2021. Impact of vaccination on new SARS-CoV-2 infections in the United Kingdom. *Nature medicine* 27, 8 (2021), 1370–1378.

[79] Olha Puhach, Kenneth Adea, Nicolas Hulo, Pascale Sattonnet, Camille Genecand, Anne Iten, Frédérique Jacquérioz, Laurent Kaiser, Pauline Vetter, Isabella Eckerle, and others. 2022. Infectious viral load in unvaccinated and vaccinated individuals infected with ancestral, Delta or Omicron SARS-CoV-2. *Nature Medicine* (2022), 1–10.

[80] R Core Team. 2020. R: A Language and Environment for Statistical Computing. (2020). https://www.R-project.org/

[81] Steven F Railsback and Volker Grimm. 2019. *Agent-based and individual-based modeling: a practical introduction*. Princeton university press.

[82] Sirisha Rambhatla, Sepanta Zeighami, Kameron Shahabi, Cyrus Shahabi, and Yan Liu. 2022. Toward accurate spatiotemporal covid-19 risk scores using high-resolution real-world mobility data. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 8, 2 (2022), 1–30.

[83] Matthew J Roorda, Eric J Miller, and Khandker MN Habib. 2008. Validation of TASHA: A 24-h activity scheduling microsimulation model. *Transportation Research Part A: Policy and Practice* 42, 2 (2008), 360–375.

[84] RStudio Team. 2020. RStudio: Integrated Development Environment for R. (2020). http://www.rstudio.com/

[85] Raphael Sonabend, Lilith K Whittles, Natsuko Imai, Pablo N Perez-Guzman, Edward S Knock, Thomas Rawson, Katy AM Gaythorpe, Bimandra A Djaafara, Wes Hinsley, Richard G FitzJohn, and others. 2021. Non-pharmaceutical interventions, vaccination, and the SARS-CoV-2 delta variant in England: a mathematical modelling study. *The Lancet* 398, 10313 (2021), 1825–1835.

[86] Phillip Stroud, Sara Del Valle, Stephen Sydoriak, Jane Riese, and Susan Mniszewski. 2007. Spatial Dynamics of Pandemic Influenza in a Massive Artificial Society. *Journal of Artificial Societies and Social Simulation* 10, 4 (2007), 9. https://www.jasss.org/10/4/9.html

[87] Goce Trajcevski. 2011. Uncertainty in spatial trajectories. In *Computing with spatial trajectories*. Springer, 63–107.

[88] Robert Trevethan. 2017. Sensitivity, specificity, and predictive values: foundations, pliabilities, and pitfalls in research and practice. *Frontiers in public health* 5 (2017), 307.

[89] Guido Van Rossum and Fred L Drake Jr. 1995. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

[90] OJ Watson, Patrick Walker, Charlie Whittaker, Peter Winskill, and Giovanni Charles. 2020. Squire - SEIR transmission model of COVID-19. (May 2020). https://mrc-ide.github.io/squire/ [Online; accessed 07-November-2022].

[91] Hadley Wickham. 2007. Reshaping data with the reshape package. *Journal of Statistical Software* 21, 12 (2007). https://www.jstatsoft.org/v21/i12/

[92] Hadley Wickham. 2016. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org

[93] Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. 2019. Welcome to the tidyverse. *Journal of Open Source Software* 4, 43 (2019), 1686. DOI:http://dx.doi.org/10.21105/joss.01686

[94] Hadley Wickham, Romain François, Lionel Henry, and Kirill Müller. 2021. dplyr: A Grammar of Data Manipulation. (2021). https://CRAN.R-project.org/package=dplyr R package version 1.0.7.

[95] Uri Wilensky. 1999. *NetLogo*. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo/

[96] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. 2011. Driving with knowledge from the physical world. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 316–324.

[97] Sepanta Zeighami, Cyrus Shahabi, and John Krumm. 2020. Estimating spread of contact-based contagions in a population through sub-sampling. *arXiv preprint arXiv:2012.06987* (2020).

[98] Xichen Zhang, Suprio Ray, Farzaneh Shoeleh, and Rongxing Lu. 2021. Efficient Contact Similarity Query over Uncertain Trajectories.. In *EDBT*. 403–408.

Appendix A

A.1 Disease models as transition matrix

Table A.1: SIR disease model - transition matrix

	susceptible	infectious	removed
susceptible		$f(contact_time)$	
infectious			0.1
removed			

Table A.2: SIR with Testing and Isolation disease model - transition matrix

	S	I	S_{iso}	$S_{iso,+ve}$	I_{iso}	R	$I_{iso,+ve}$
S		$f(contact_time)$	Test				
I					Test	0.1	
S_{iso}	TNR			FPR			
$S_{iso,+ve}$	1/12						
I_{iso}		FNR				0.1	TPR
R							
$I_{iso,+ve}$						0.1	

Appendix B

B.1 Parallel query execution experiment results

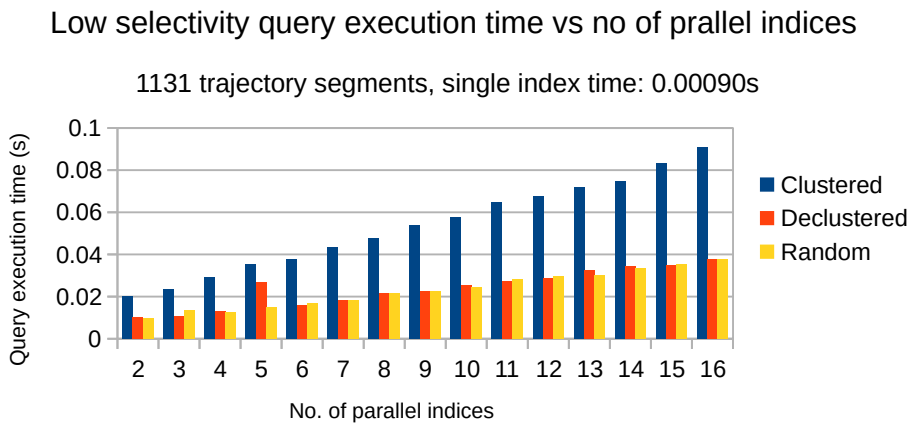


Figure B.1: Low selectivity, parallel query on 1131 trajectory segments

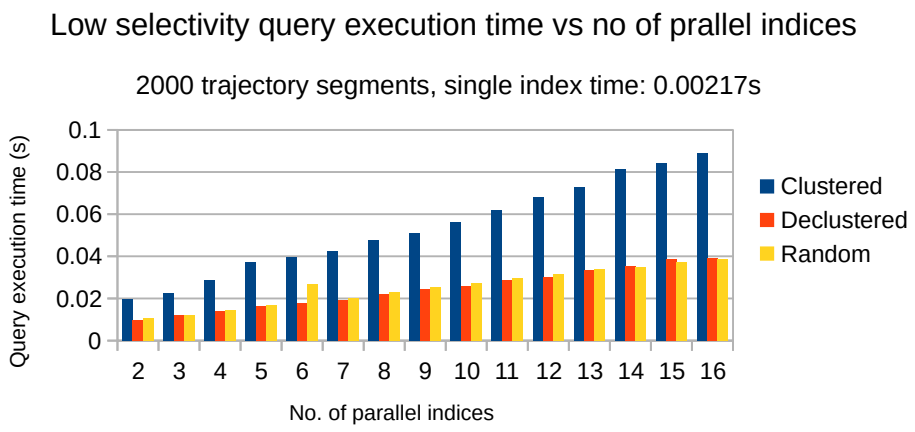


Figure B.2: Low selectivity, parallel query on 2000 trajectory segments

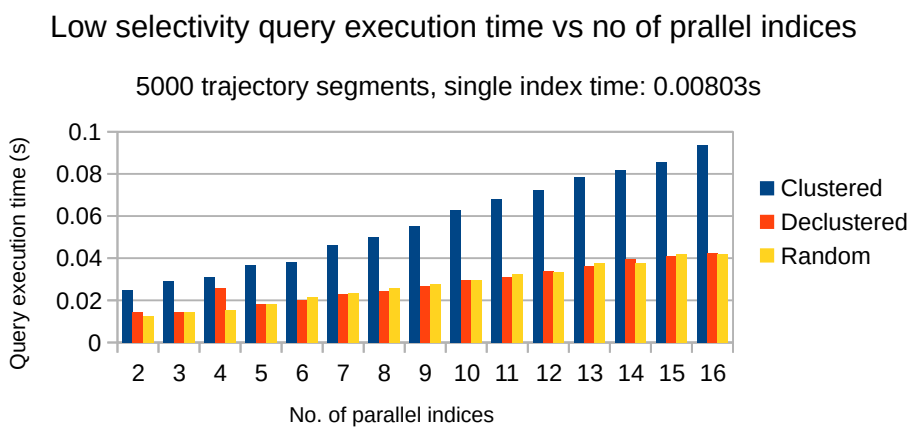


Figure B.3: Low selectivity, parallel query on 5000 trajectory segments

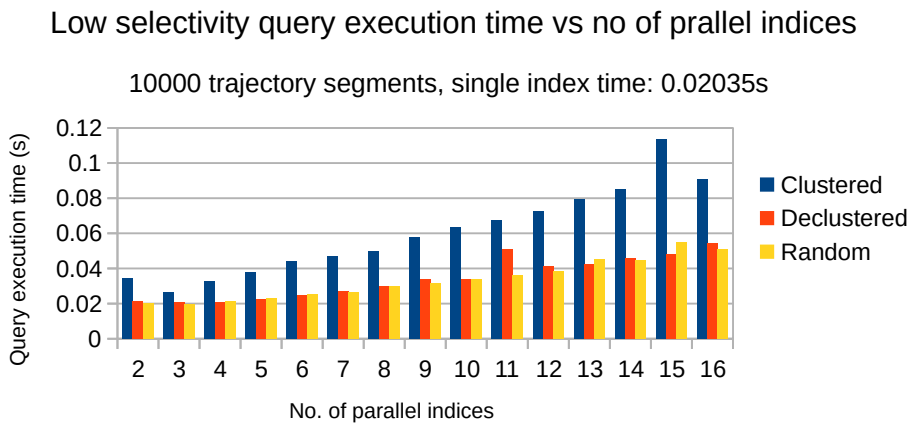


Figure B.4: Low selectivity, parallel query on 10000 trajectory segments

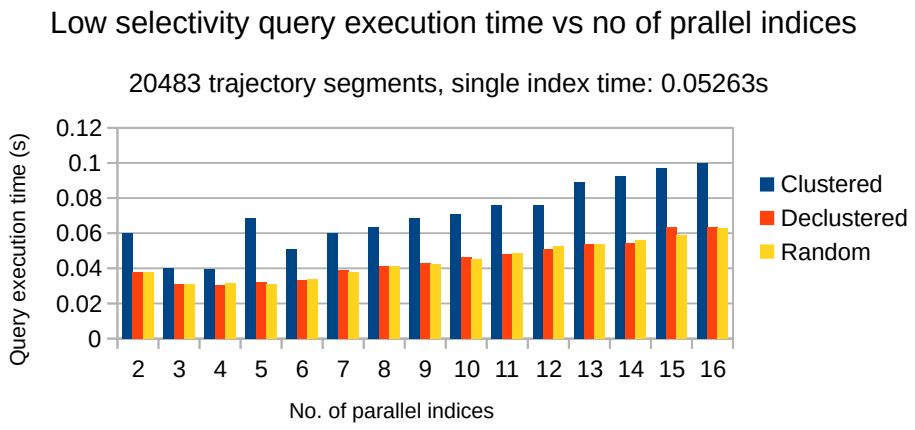


Figure B.5: Low selectivity, parallel query on 20483 trajectory segments

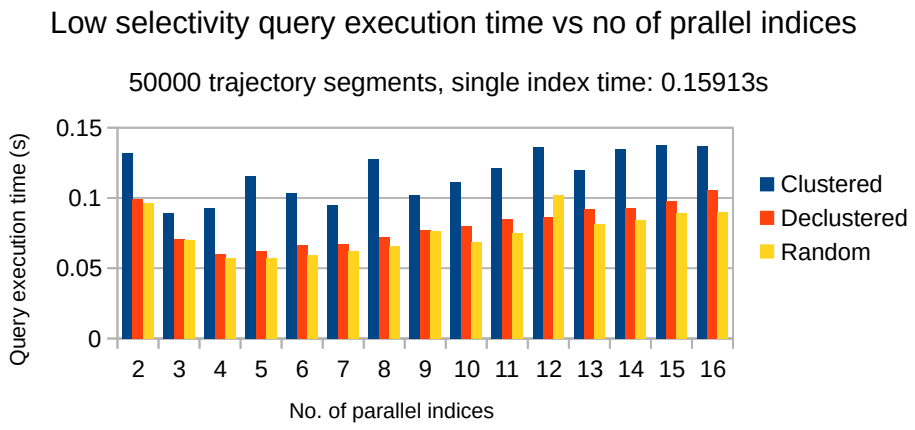


Figure B.6: Low selectivity, parallel query on 50000 trajectory segments

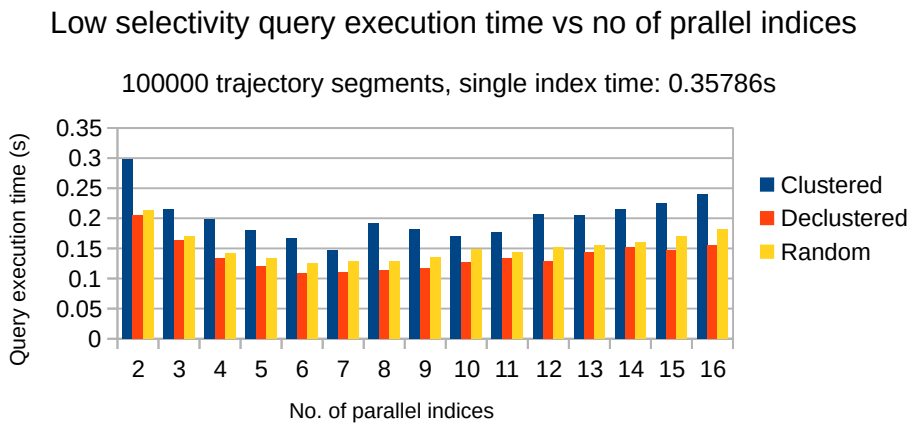


Figure B.7: Low selectivity, parallel query on 100000 trajectory segments

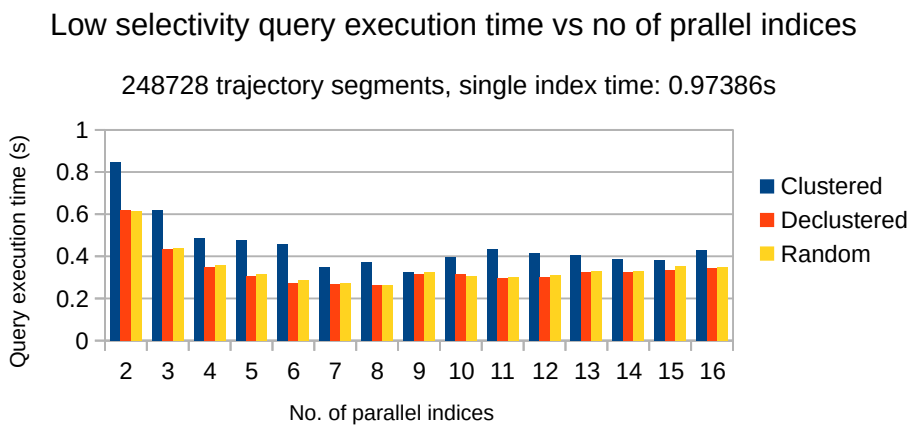


Figure B.8: Low selectivity, parallel query on 248728 trajectory segments

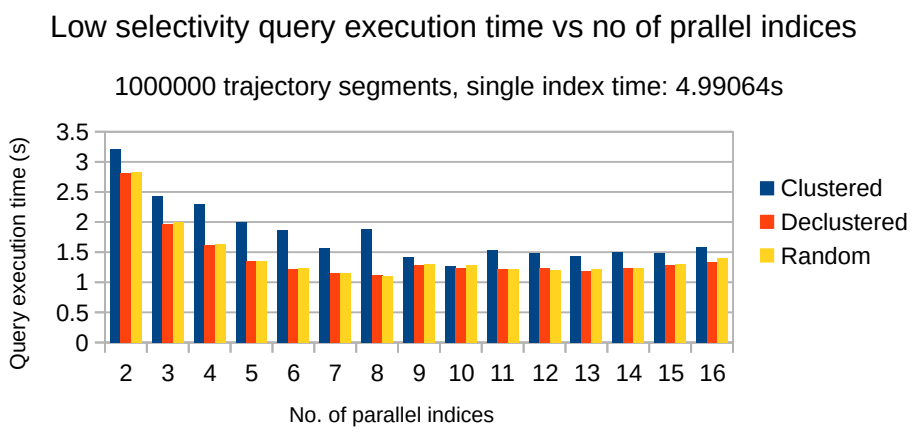


Figure B.9: Low selectivity, parallel query on 1000000 trajectory segments

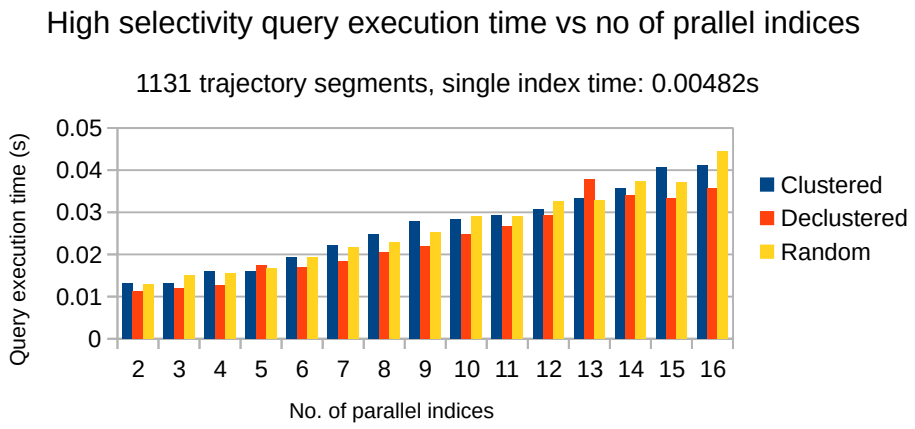


Figure B.10: High selectivity, parallel query on 1131 trajectory segments

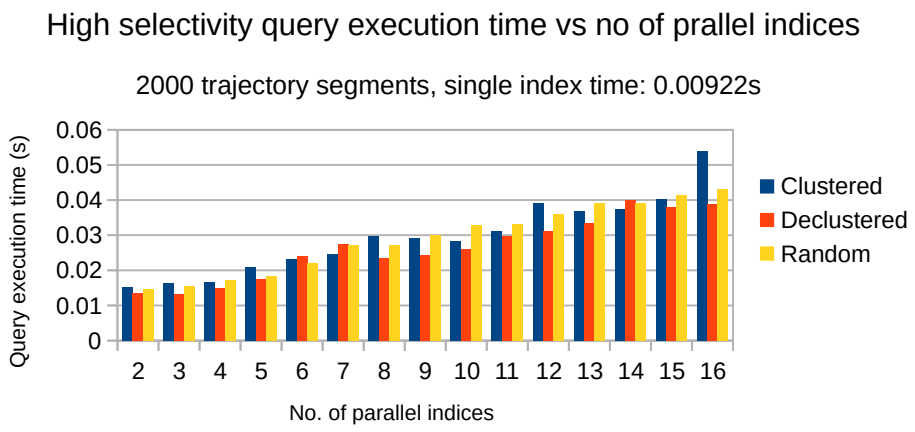


Figure B.11: High selectivity, parallel query on 2000 trajectory segments

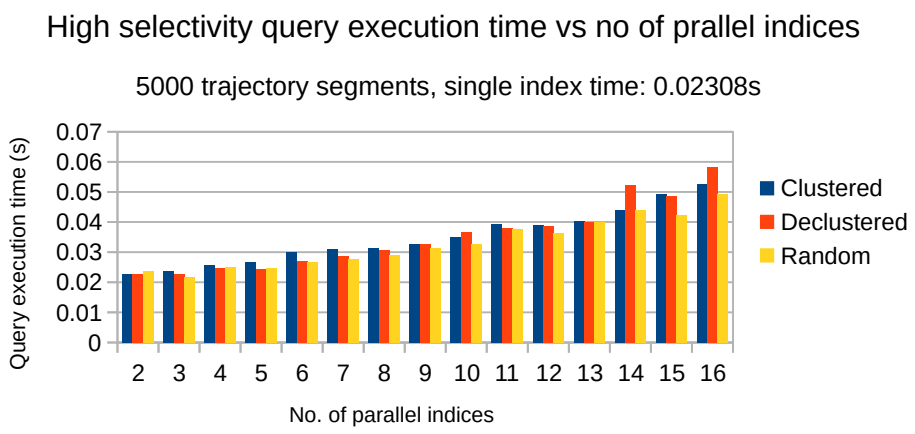


Figure B.12: High selectivity, parallel query on 5000 trajectory segments

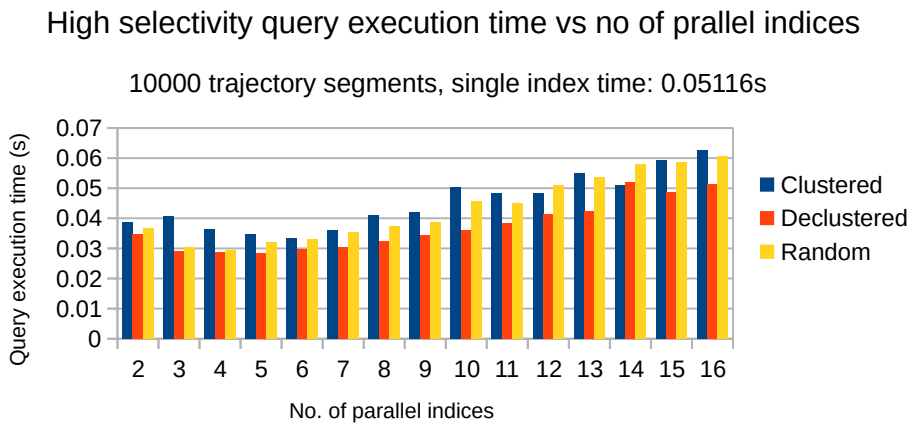


Figure B.13: High selectivity, parallel query on 10000 trajectory segments

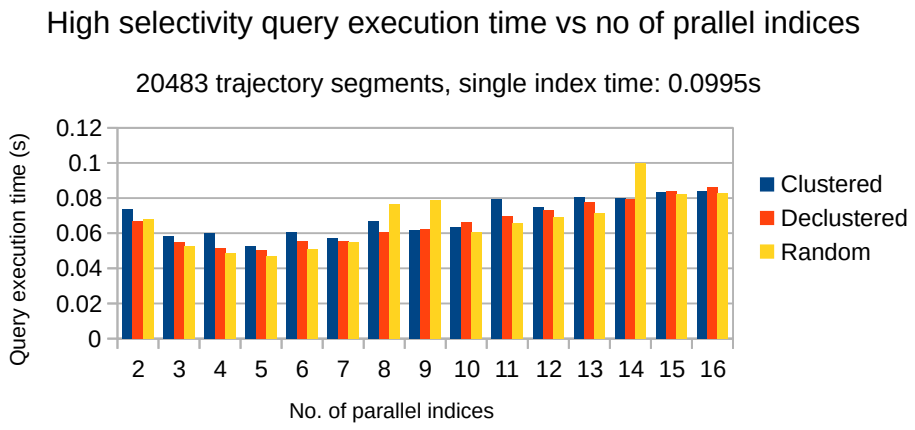


Figure B.14: High selectivity, parallel query on 20483 trajectory segments

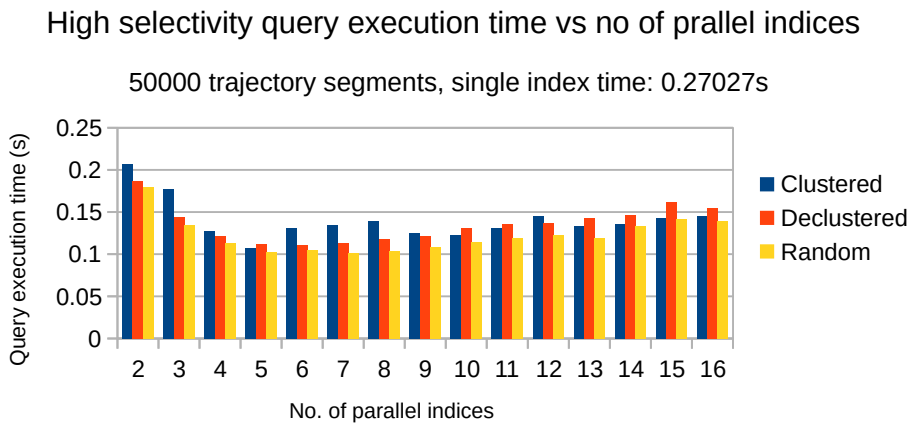


Figure B.15: High selectivity, parallel query on 50000 trajectory segments

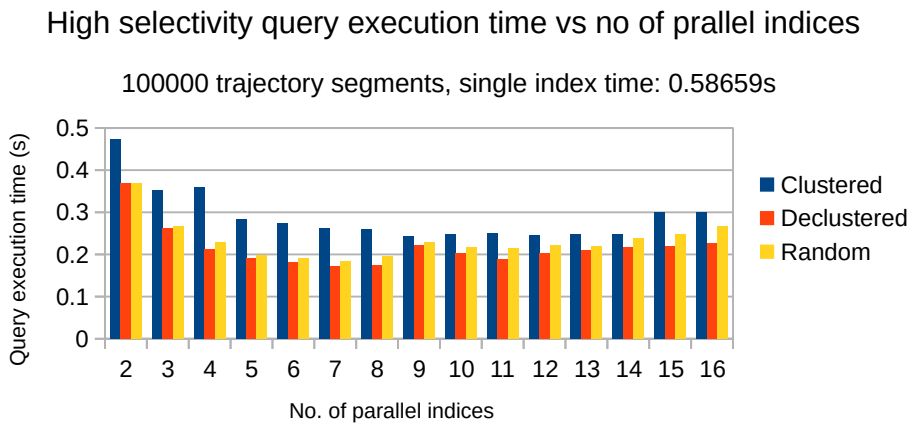


Figure B.16: High selectivity, parallel query on 100000 trajectory segments

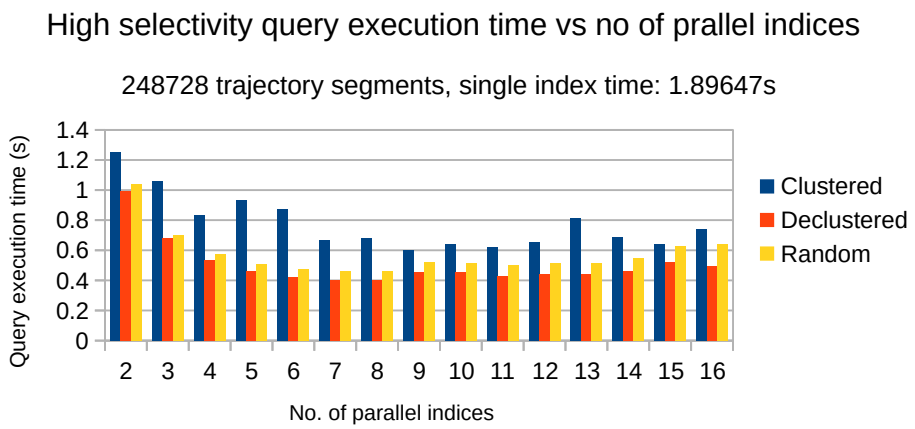


Figure B.17: High selectivity, parallel query on 248728 trajectory segments

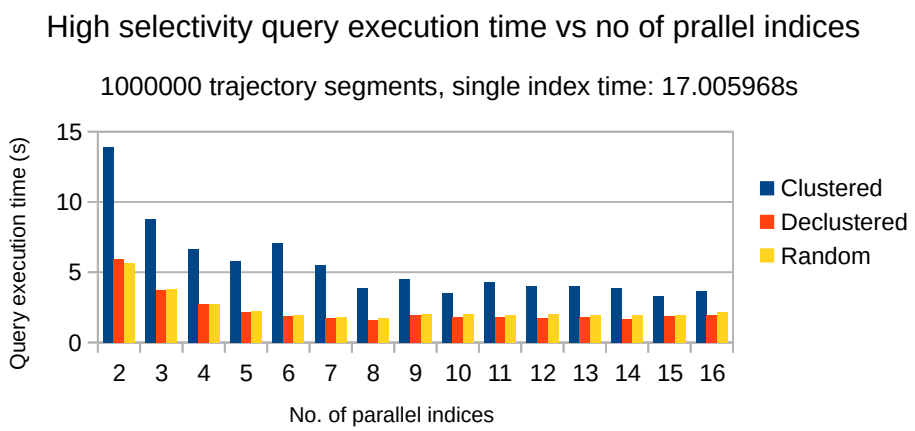


Figure B.18: High selectivity, parallel query on 1000000 trajectory segments

B.2 Distributed query execution experiment results

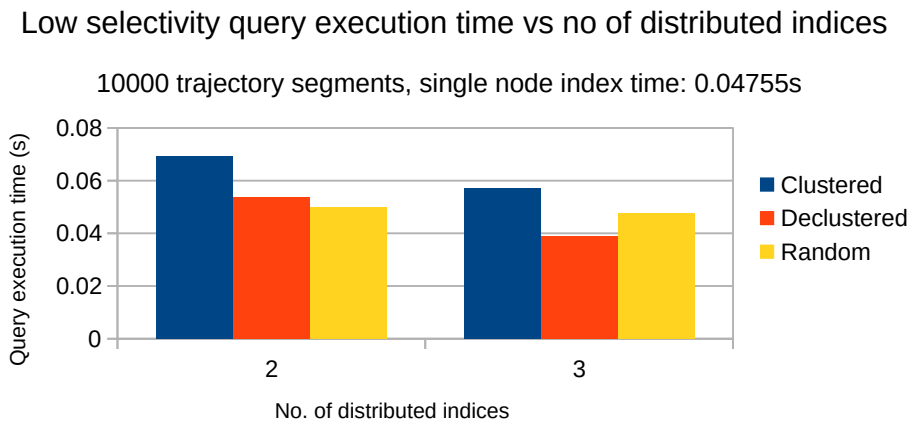


Figure B.19: Low selectivity, distributed query on 10000 trajectory segments

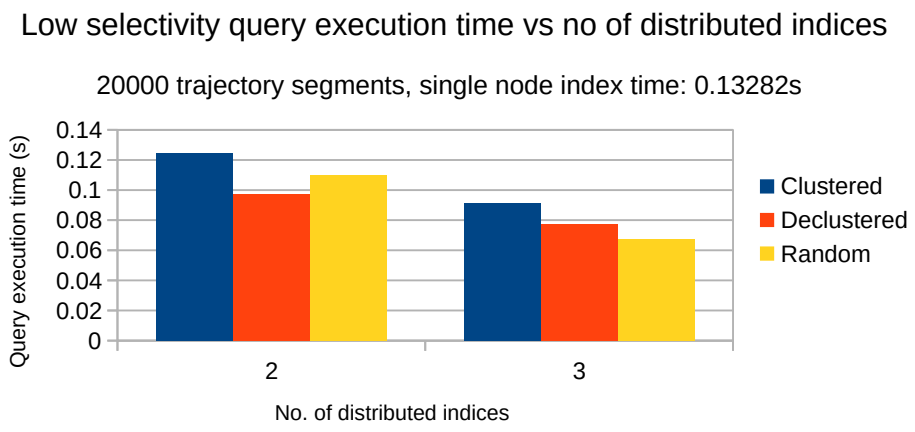


Figure B.20: Low selectivity, distributed query on 20000 trajectory segments

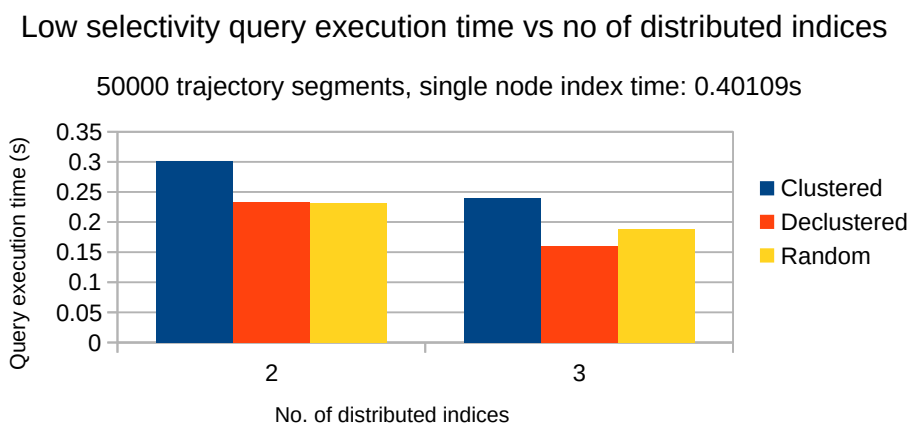


Figure B.21: Low selectivity, distributed query on 50000 trajectory segments

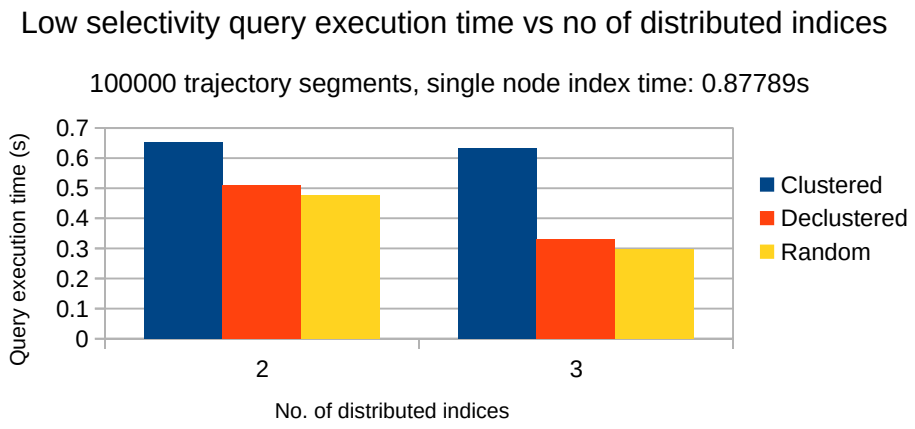


Figure B.22: Low selectivity, distributed query on 100000 trajectory segments

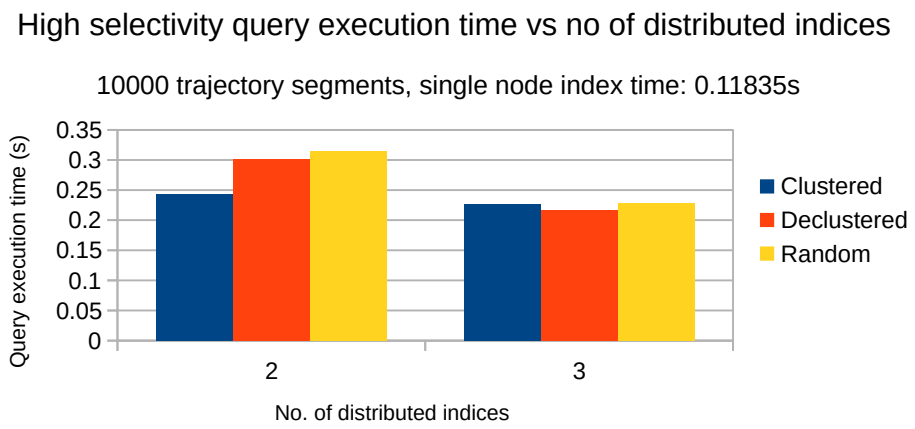


Figure B.23: High selectivity, distributed query on 10000 trajectory segments

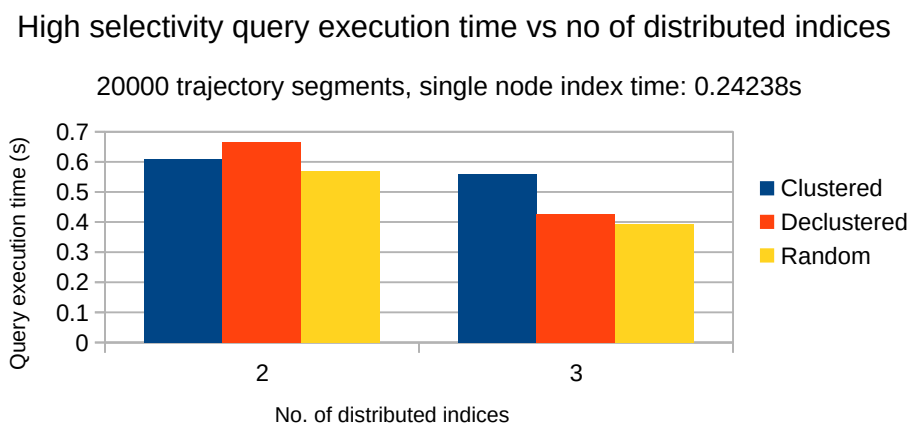


Figure B.24: High selectivity, distributed query on 20000 trajectory segments

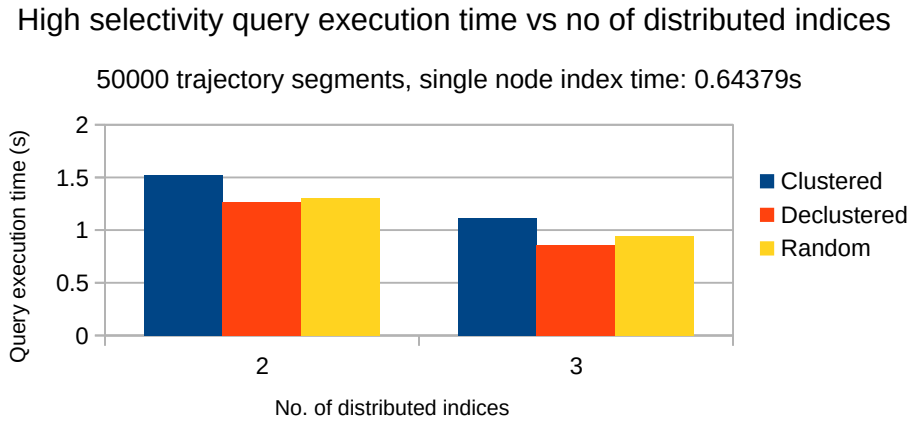


Figure B.25: High selectivity, distributed query on 50000 trajectory segments

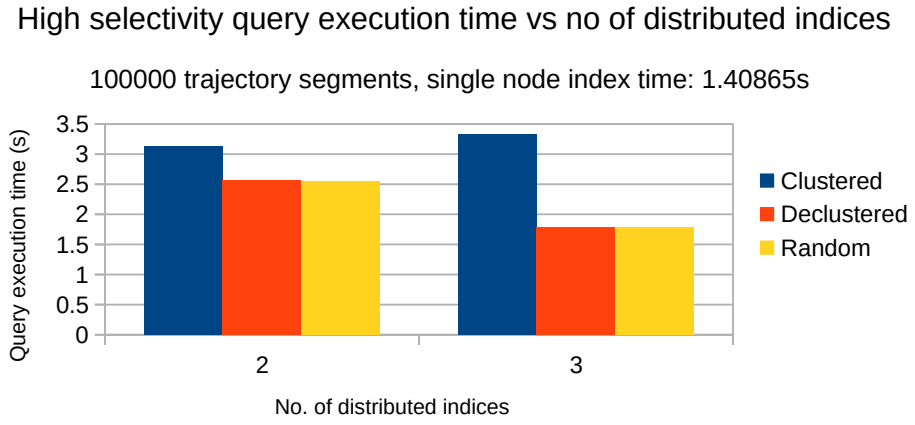


Figure B.26: High selectivity, distributed query on 100000 trajectory segments

Vita

Candidate's full name: Avinaba Mistry
University attended (with dates and degrees obtained):
2010-2014: Bachelor of Computer Science and Engineering, Jadavpur University, India
Publications: N/A
Conference Presentations: 3 minute short talk - MfPhest 22, The Fields Institute, Toronto