Spatial Modelling of Non-Pharmaceutical Interventions against COVID-19: taking the UK and China as Examples

by

Shitian Zhang

September 19, 2023

Supervisor: Alan Penn

A Dissertation submitted in part fulfilment of the

Degree of Master of Science (MSc) Built Environment

Space Syntax: Architecture and Cities

Bartlett School of Architecture

University College London

UCL FACULTY OF THE BUILT ENVIRONMENT
BARTLETT SCHOOL OF ARCHITECTURE

MSc SPACE SYNTAX: ARCHITECTURE AND CITIES
Coursework Submission Form

1.1.1. Student's Name (IN CAPS): SHITIAN ZHANG Word Count: 9758

1.1.2. Dissertation Title: Spatial Modelling of Non-Pharmaceutical Interventions against COVID-19: taking UK and China as Examples

1.2.Deadline for Final Submission: Sep 12, 2023

1.3.Actual Date of Submission (if different): Sep 19, 2023

(If different please state whether this has been agreed with the module leader and the appropriate documentation provided)

Extenuating Circumstance approved with self-certification. Documentation is on the next page. EC Reference Code: EC-SZHAQ98-001.

*1.4.*Student Statement

- **I confirm I understand the meaning of plagiarism and that I may be penalised for submitting work that has been plagiarised.**
- **I confirm the work submitted is my own unaided work and that all sources used in its preparation and all quotations are clearly cited.**
- **I confirm that all work will also be submitted electronically and that this can be checked using the JISC detection service, Turnitin.**
- **I confirm that I understand that the work cannot be assessed unless I have handed in both hard copy and electronic versions.**

1.4.1. Student Signature: Shitian Zhang

Your work will be returned unmarked if you fail to sign here

Abstract

The research presents a spatial model based on space syntax for simulating the transmission of COVID-19 as an implementation of SIER model and quantified the impact of several non-pharmaceutical interventions. The aim of the research is to provide a spatial perspective in modelling the pandemic, so that the effect of spatially specified interventions can be clearly simulated and measured. Taking Inner London and Metropolitan Area of Beijing as case areas, daily movements and contacts of the whole population were simulated on the street segment maps. The effects of public policies were quantified on their performances, namely the speed of transmission, and costs, namely the proportion of people affected by the policies. Based on this model, two policy sets abstracted from policies in the UK and China were tested with several sets of time lags of execution. The research found that the policy set based on Chinese policies performs better in controlling the transmission under the condition of short responding time towards the pandemic. When responding time rises beyond a threshold specific to each city, the speed of transmission rapidly increases along with the side effects of interventions. Through the modelling of spatial usage and movements, the perspective of space is recalled in epidemiological analyses. With more detailed modelling of human behaviours in the future, this model will be more helpful for providing reference for public medical emergencies.

Key words

space syntax; COVID-19; non-pharmaceutical intervention; simulation; China; the UK

Table of Contents

Lists of Illustrations and Tables

Acknowledgements

Firstly, I want to present my special gratitude for Professor Alan Penn, who gives me lots of support with broad scope of relevant knowledge and deep insight into the space syntax, and Professor Kayvan Karimi, who presents us with a clear structure of the whole programme of Space Syntax and cares about our academic growth and wellbeing throughout the year.

Then, I want to thank Doctor Tasos Varoudis and Senior IT Officer of UCL, Donald Ivan-Dixon for their support in providing essential hardware for the running of simulation model, as well as Space Syntax Limited and Doctor Tao Yang for their datasets of segment maps. I also need to thank all of our tutors, PGTAs and staffs, including but not limited to Drew Pessoa, Sophia Psarra, Sam Griffiths, Kerstin Sailer, Rosica Pachilova, Laura Vaughan, Kimon Krenz, Po Nien Chen, Sepehr Zhand, Merve Okkali Alsavada and Chenyang Li.

In this year, I am also supported by my friends and classmates. I want to specially thank my friend, Yongjiang Wu. We always support each other and go through every hard moment together throughout this year. I must not forget my friend since childhood, Zhiyuan Zhao, without whom I might not have come to UCL and gained such a memorable year. I am also thankful for Sonia Sasi Stephen, Luisa Amann, Youssef Elassaly, Alessandra Di Cerbo, Fukun Wang, Huixin Lyu, Neha Bhalerao, Tamanna Sethi, Tanya Aggarwal, and all other friends that I encountered this year.

Last but the most importantly, I want to thank my dear mum, dad, and grandparents. Forever love for all of you. And I sincerely hope that my grandmother on my mother’s side can rest in peace in heaven.

Sep 19th, 2023

1 Introduction

1.1 Relevance Statement

The pandemic of COVID-19 has caused immense damage to the wellbeing of people worldwide. Although the hardest time has passed, we need to learn from this disaster. Looking back on public policies against the pandemic worldwide, various approaches have been used according to the changing perspectives towards the pandemic. For example, China is known for exerting collective power of the whole nation and executing rather strict spatial intervention to prevent COVID-19 from spreading. The UK, on the other hand, prefers gentler ways of regulation. So, it is natural to question whether there exists an optimal solution against the pandemic, which both has a controlling effect over the pandemic and leaves little negative impact on daily lives.

With this background, this paper focuses on the impact of various policies. In this paper, the word *policy* specifically refers to non-pharmaceutical interventions (NPIs) aiming at reducing the transmission of the virus. It seems obvious that space plays an important role in the transmission of a pandemic. However, the relationship between the spatial configuration and the transmission has not been clarified. Therefore, this study is trying to build a simulation model and quantify the impact of different policies based on spatial interventions. Based on this, a comparative analysis is carried out to check the effect if the policy of one nation was applied to another. In short, this study is trying to present a new way of checking the effect of policies on the perspective of space and deepen the understanding towards the pandemic.

1.2 Research question

(a) How to quantify the performance of a non-pharmaceutical intervention against COVID-19?

(b) How do different factors such as the combination of policies, time lag of execution and the spatial configuration of cities affect the performance of interventions in controlling the spread of COVID-19?

1.3 Structure of the study

Firstly, the analysis is based on a model to simulate the spread and transmission of Coronavirus. The aim of the model is to introduce the element of the space into the modelling, where the space is often ignored or simplified into abstract *containers* of people rather than a real stage of movement and encounter. Therefore, the simulation model will be introduced after the literature review, as the fundamental method of analysis.

Secondly, the main policies applied in study areas against the pandemic will be quantified and introduced into the model. By affecting the movement patterns of individuals, or changing certain key factor, the speed and range of pandemic transmission may vary among policies. In this paper, different combinations of policies will be analysed in both study areas to check to what extent the policies could control the spread of pandemic, and whether the policies are suitable for the area in the perspective of space. Since there are two study areas in this analysis, this paper will try exchanging the policies between these two areas and check the results. Besides, time also matters in policy application, so the time to trigger policies and the duration of policies will also be considered in the model.

2 Literature Review

2.1 Development of policies in the UK

Since the outbreak of pandemic, the understanding of UK government towards the pandemic has been gradually changing and deepening. The House of Commons published a report named *Coronavirus: lessons learned to date* on October 12th, 2021 (Hunt et al., 2021). This report is to review the governments' response to the pandemic and to reflect on the lessons learned from the disaster.

In the early period of the pandemic, the attitude of the government tended to be gentler. Although the government did not support the prevalent idea of herd-immunity openly, which indicates the state that a large proportion of the population has been infected and immune to a certain disease, they did not take immediate action to prevent the pandemic from spreading. The initial aim of the policy was flattening the peak of infection rather than stop the transmission. The earliest compulsory policy against the pandemic was announced on 12 March 2020. This is about a seven days' self-isolation for anyone with a new continuous cough or a fever (Hunt et al., 2021).

Afterwards, there were in total three national lockdowns during the pandemic (Table 1). During the lockdown, several laws were published to guarantee the restriction. There were three types of restrictions, namely gathering, movement and business restrictions. Gathering restrictions means that most of the social gatherings were banned during national lockdowns. Specific rules include restricting the number of people gathering, and other rules ranging from household mixing to certain events. Movement restrictions means that leaving home without a 'reasonable excuse' was prohibited during the national lockdowns. Travelling was also prohibited. Business restrictions means that certain businesses were required to close during the national lockdowns (Barber et al., 2022).

Besides, between the first two lockdowns, a three-tier system of local restrictions were announced on 12 October 2020 to manage the varying restrictions according to local states. Each area was categorised into one of three tiers including 'medium', 'high' and 'very high'. Different levels of restrictions were applied in areas of different tiers (Brown and Kirk-Wade, 2-21).

According to the Hose of Commons, the pandemic could have been controlled better if the restrictions were applied earlier. They also concluded several lessons. Firstly, although the *Scientific Advisory Group for Emergencies* (SAGE) has done well in providing scientific advice in public policies, especially non-pharmaceutical interventions, the House of Commons thinks that the government have taken too long waiting for the scientific evidence and missed the chance to call an immediate lockdown to stop the virus from spreading. There is also concern about lack of data to support simulation models, and whether the scientific advice has learned from international experiences. As for the localised policies, the House of Commons thinks that the three-tier system was short of clear and unified standard for categorisation, and the diverse local policies lack of scientific support.

Table 1 Duration of National Lockdowns in UK

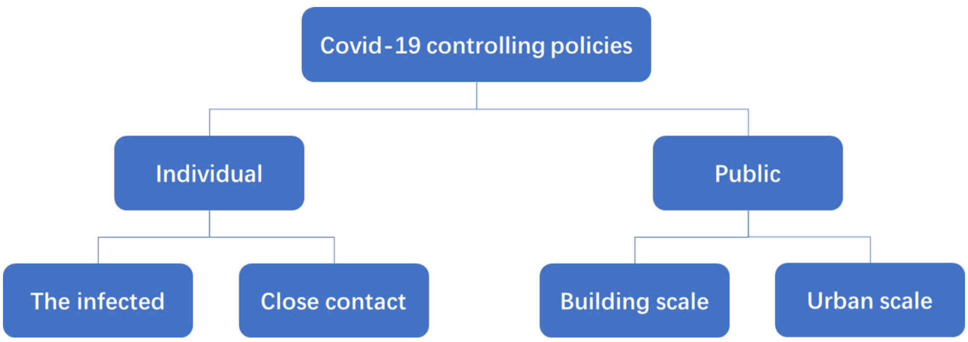
(Source: House of Commons, United Kingdom)

	Start	**End**	**Number of weeks**
Lockdown one	27 Mar 2020	1 Jun 2020	10
Lockdown two	5 Nov 2020	2 Dec 2020	4
Lockdown three	6 Jan 2021	17 Mar 2021	13

2.2 Development of policies in China

After the first case of COVID-19 was discovered in Wuhan on Dec 27th, 2019, local and national authorities made a quick response. On Dec 31st, which was four days later, the National Health Commission of China sent experts to Wuhan for guidance and investigation. People in Wuhan was noticed to wear face masks and avoid gathering on the same day. Afterwards, National Health Commission initially determined the pathogen as a new kind of coronavirus and reported to World Health Organisation on Jan 9th, 2023. On Jan 19th, National Health Commission confirmed that the coronavirus could be transmitted among humans, and Wuhan was locked down immediately on Jan 20th (China, 2020). After months of fight against the virus, the transmission was primarily controlled, and the work of controlling the pandemic becomes normalised since May 2023. Then, a series of policies were published and will be briefly introduced below.

Since the outbreak of COVID-19 in Dec 2019, medical support and segregation have been executed on various scales to prevent the pandemic from spreading. This work is guided by the national government through a set of policy documents, the most important of which is *Covid-19 Prevention and Control Guideline.* Policies have covered many aspects, ranging from the ones specific to the infected individuals, as well as broader ones toward the public (Graph 1).

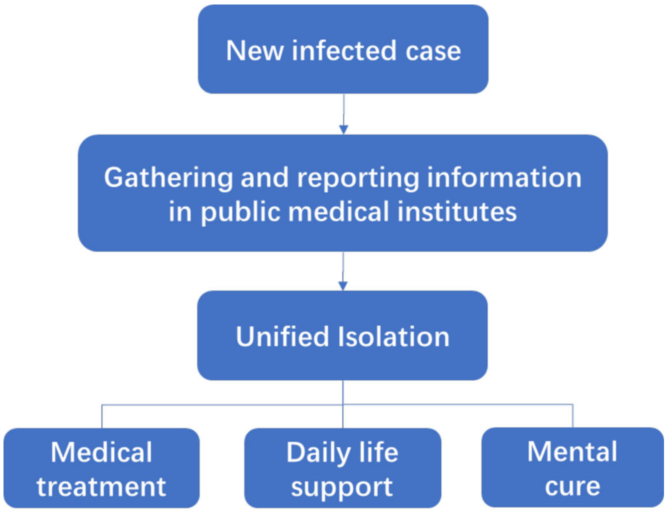


Graph 1 Structure of COVID-19 Controlling Policies in China

(Source: the author)

The individual intervention differs between the infected cases and the close contact.

Firstly, newly infected cases would be reported and isolated at a certain site, where they would receive medical treatment and daily life support (Graph 2). Mental health is also considered important during such isolation. To deal with the rapidly increasing infected cases, many temporary treatment sites have been built during the pandemic (Figure 1).



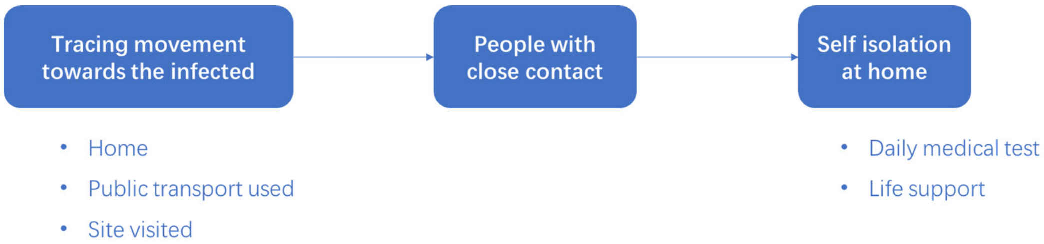
Graph 2 Treatment on Infected Cases

(Source: the author)

Figure 1 Temporary Treatment Site in Wuhan Sports Centre

(Source: http://www.news.cn)

Metadata plays an important role in the control of COVID-19. Based on the metadata and epidemiological investigation, the range of movement of the infected case was analysed, which includes but is not limited to household, transport and site visited. Afterwards, people who might have contacted the infected will be defined as close contact. They will be informed and required for self-isolation at home for a certain period of time, where medical test is executed daily by staffs in the community (Graph 3).



Graph 3 Treatment on Close Contacts

(Source: the author)

To confirm the transparency of the spread of COVID-19 and inform the public of the potential threat, the tracing information is collected anonymously and reported regularly by official health institutes in the official press conferences of COVID-19 in Beijing, which has also been mentioned in Section 3.4. Below is one example of reported infected cases in the 215th Official Press Conference of COVID-19 in Beijing.

> *"…**Case No. 6**, male, 9 years old, currently living in Ronghui Community, Tiangongyuan Sub-District, Daxing District, is the grandson of Case 5. He has visited the confirmed case reported on January 17 many times. On January 18, as a close contact of the confirmed case, he underwent unified isolation for medical observation and nucleic acid testing. On January 19, the testing report was positive. On the same day, he was transferred to Ditan Hospital by ambulance. Based on comprehensive epidemiological history, clinical manifestations, laboratory testing and imaging examination results, he has been confirmed as an infected case on the same day, and the clinical classification was mild…"*

There are also policies of entry permission towards the public. Any visitor to a neighbourhood or indoor public space is required to scan the Health QR Code to make registration at the entrance, and by this way the tracing information will be gathered anonymously by a platform called National Metadata Pass (Figure 2). On checking the National Metadata Pass of the visitor, the administrator can easily learn the current Risky Status of the visitor and Risky Areas visited within 14 days.

Figure 2 National Metadata Pass

(Source: WeChat)

Based on the individual trace, the *Risky Status* of the visitor is judged by National Metadata Pass into four levels. According to the Risky Status of an individual, the National Metadata Pass will be shown in green, yellow, orange, and red (Figure 3). Green status means that the visitor is safe from the pandemic. Yellow status means that the visitor is from risky countries, which are regularly updated by national government. Orange status means that the visitor is close contact explained in Section 4.2. And red status means that the visitor is a confirmed or suspected patients.

Entering permission is dependent on the Risky Status of the visitor. Proof of health is required for the visitor in yellow status, and visitors in orange or red status are not allowed to enter public spaces.

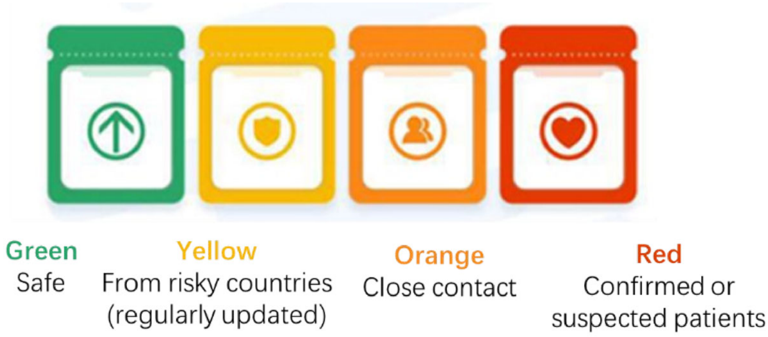


Figure 3 Risky Status on National Metadata Pass

(Source: WeChat)

As has been explained, individual Risky Status is judged by whether one has visited Risky Areas within 14 days. Therefore, to support this judgements, *three levels of Risky Areas* are divided nationwide. The overall standard is dynamically updated by national government through *Covid-19 Prevention and Control Guideline*, and the division and publication are executed by local authorities. Correspondent to Section 3.4, the overall data of Risky Areas published nationwide are collected regularly and announced on multiple platforms. Therefore, it is easy for everyone to check the newly updated Risky Areas throughout China.

Table 2 shows the regulation standard of Risky Areas in Beijing. To balance the need of pandemic control and convenience of citizens, High and Medium Risky Areas are often restricted in the range of neighbourhoods. People in High Risky Areas are restricted at home, while those in Medium Risky Areas can only move within the area. Regular medical test and basic living needs are supported by local authorities. Low Risky Area is usually defined as the rest areas in the sub-district where High or Medium Risky Areas are located. There is no compulsory restriction on the people in Low Risky Areas, so this is just used as a reminder of the potential risk of infection.

Table 2 Definition and Spatial Range of Risky Areas in Beijing

(Source: Beijing Government)

Risky Level	Standard	Restriction	Spatial Range
High	Within 14 days, there are - 5 new confirmed cases or - 2 new clustering infected cases.	Restricted at home.	Community (Neighbourhood)
Medium	Within 14 days, there are - 2 - 5 new confirmed cases or - 1 new clustering infected cases.	Restricted within the area.	Community (Neighbourhood)
Low	Within 14 days, no new cases found.	No compulsory restriction.	Usually the sub-district where High/Medium Risky Areas are located

2.3 Research on the spatial effects of pandemic

There is lots of research on the spread and control of COVID-19 since the outbreak of pandemic. In this topic, space syntax theories and methodologies (Hillier and Hanson, 1984) are widely used in spatial analyses, ranging from national scale to building scale.

Research on national and urban scale mostly focuses on finding and analysing various factors affecting the space. For example, Lima et al. carried out a study on national scale, aiming at testing the significance of certain factors to the spread of COVID-19 in the US (Lima et al., 2021). Taking city as the basic unit of research, they found that there is positive correlation between deaths caused by the pandemic and several factors, such as walkability score, urban population size and density, etc. Similarly, there are also studies on the urban scale. Legeby et al. analysed the impact of the pandemic on the usage of urban green facilities in Stockholm and emphasized the importance of availability of urban green space (Legeby et al., 2022). Another research on Xi'an City checked the change in the spatial configuration with space syntax methods and tested several related factors (Yuan et al., 2022). These studies provided comparison on the change of spatial usage before and after the outbreak of pandemic, and revealed the spatial impact of pandemic from various perspectives. However, they are seldom related to relevant policies, which should be an important factor in reshaping the space in the pandemic era.

Research on the scale of buildings mostly aims at testing the change of spatial usage caused by the pandemic. For example, Büyükşahin used Visibility Graph Analysis to investigate the change in usage patterns of shopping malls during the pandemic (Büyükşahin, 2023). Askarizad et al. analysed the difference in the frequency of people going to healthcare centres before and after the pandemic (Askarizad et al., 2022). Wayfinding in the indoor space of hospitals was also analysed with the combination of space syntax methods and mathematical models. In this topic, Mustafa and Ahmed also did a more detailed analysis on the location of clinics part in hospitals (Mustafa and Ahmed, 2022). They pointed out that although placing the clinic at the location of high integration in a hospital is beneficial for wayfinding, it also increases the possibility of clustering, which may lead to a higher rate of infection. Moreover, there is another research on social distancing in clubhouses with graph analysis (Abdul Nasir et al., 2021). Besides, Chen et al. did a sophisticated emulation towards the spread of COVID-19 (Chen et al., 2022). The model is insightful for the design of indoor space. There are also studies towards outdoor space, testing the availability of several public spaces during the pandemic based on the spatial network (Istiani et al., 2023). In short, studies on the building scale provide deep insight into spatial mechanism of the spread of pandemic and are beneficial for the medical control.

Besides the research on physical space, there are also analyses in social or mental aspects during the pandemic. For example, Park et al. collected data from social media and analysed people's perception towards some airports (Park et al., 2022). In another novel research, spatial network was analysed in combination with social network (Li et al., 2022). It found that in the process of isolation, the infected were deprived of their original roles in the social network, and a new social network was formed within the isolation area based on the spatial proximity. These studies imply various factors which might affect the management of space during the pandemic. The text mining techniques used in the research also inspired this paper.

2.4 Research on pandemic transmission modelling

Since the outbreak of pandemic, various models have been used to predict the trend of infection, medical needs, and effects of policies (Colbourn, 2020). One of the models widely accepted in the UK is the *CovidSim* model derived from classic *SEIR* model (Schneider et al., 2020). In SIER model, the process of infection is divided into several stages such as Susceptible, Exposed, Infected and Removed (Davies et al., 2020). Based on this, *CovidSim* model develops a more precise method to simulate the process of contact and infection. The model uses a stochastic way and treats each person individually rather than treating every age group as a whole (Walker et al., 2020). To simulate contact, the model set four scenarios including the household, school, workplace and wider community, where different infectious rates upon contact are set according to the age, place, etc (Ferguson et al., 2020). However, the model is not spatial, since all the scenarios don't have real locations. There are also critical thoughts on the *CovidSim* model. For example, when the model is used for political consultancy, the result of the model might affect the policy and the expectation towards the pandemic. This may in turn affect the pandemic in reality (Van Basshuysen et al., 2021).

3 Study Area, Data and Methodology

3.1 Study areas

The research areas selected in this paper are the central part of London and Beijing. Compared to the outer regions, the selected areas have higher population density and more social and economic activities, and so are more fragile towards the pandemic. Therefore, study focusing on these areas should be more instructive.

Inner London was chosen as the first study case, denoted as Case London in later sections. This area consists of the City of London and 12 London boroughs (Figure 4). The area and population of each component of this area are listed in Table 3.

Figure 4 Case London: Inner London

(Source: the author)

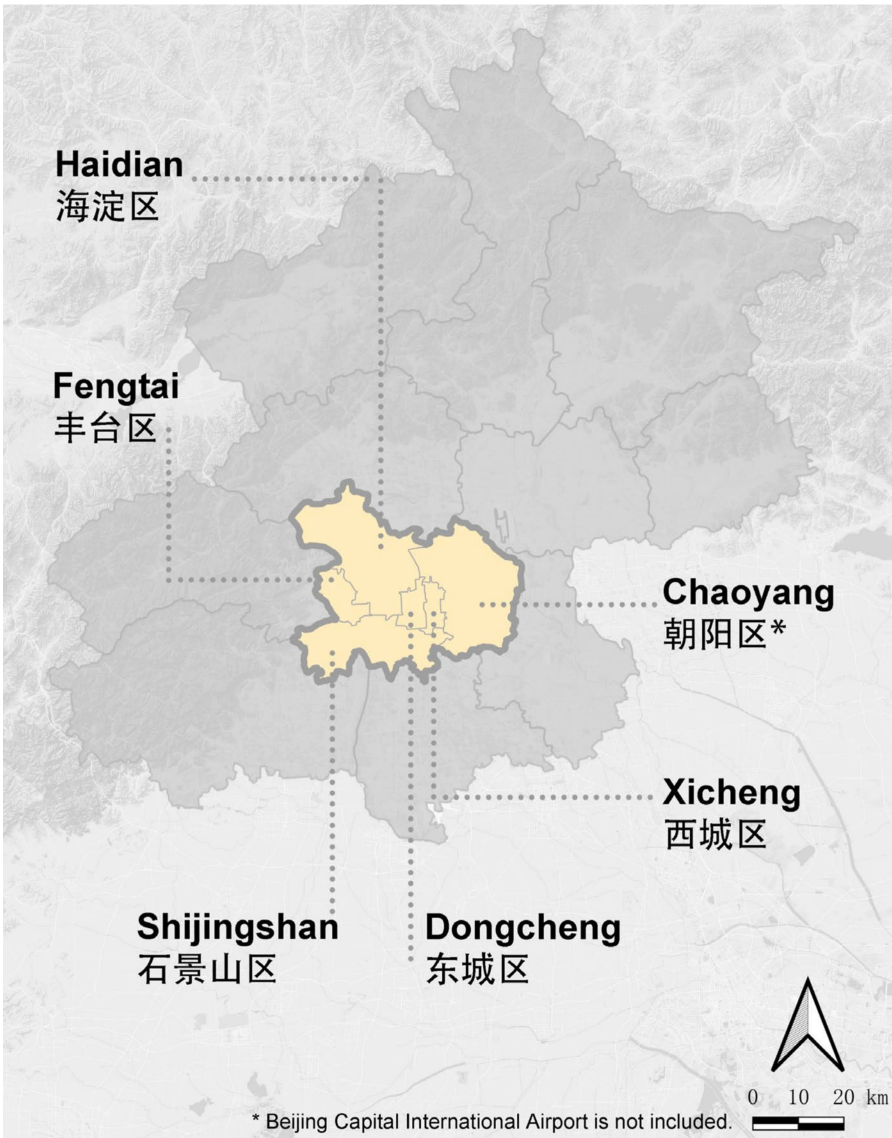


Figure 5 Case Beijing: Metropolitan Area of Beijing

(Source: the author)

The second study area is the Metropolitan Area of Beijing, denoted as Case Beijing in later sections. This area consists of six districts, which are shown in Figure 5. District is an administrative unit subject to cities. It should be noted that the area of Beijing Capital International Airport is an enclave belonging to Chaoyang District but is not included in the study area of Case Beijing. The area and population of each component of this area are listed in Table 4.

Table 3 Details of Case London

(Source: Office for National Statistics, United Kingdom)

	Area (km^2)	**Population in 2021** (thousands)
Camden	21.79	280
City of London	3.148	10.9
City of Westminster	22.05	270
Greenwich	50.46	289
Hackney	19.05	281
Hammersmith and Fulham	17.16	184
Islington	14.86	248
Kensington and Chelsea	12.39	157
Lambeth	27.25	321
Lewisham	35.33	305
Southwark	29.93	320
Tower Hamlets	21.58	332
Wandsworth	35.23	330
Total	310.2	3328

Table 4 Details of Case Beijing

(Source: Beijing Municipal Bureau of Statistics, China)

	Area (km^2)	**Population in 2021** (thousands)
Chaoyang*	455.5	3449
Dongcheng	41.93	708
Fengtai	306.0	2015
Haidian	428.9	3130
Shijingshan	84.35	566
Xicheng	50.35	1104
Total	1367	10972

* Beijing Capital International Airport is not included

3.2 Data

Two types of data were used in this paper. Firstly, the spatial data constitute the base of the simulation. The street segment map of London in 2018[1] and the street segment map of Beijing in 2017[2] are used as the base maps for two cases. The author modified part of the maps for the analyses. Based on the space syntax theory, streets are divided into segments at intersections, and each segment acts as the basic spatial unit for containing people and movements. The street segment map (*segment map* for short in texts below) of both case areas are shown in Figures 6 and 7. Other spatial data, including the locations of residential areas and stations for the public transport, are mostly from open sources such as Open Street Map. They will be shown in the sections below. Then, some statistical data are essential for some key parameters in the analysis, for example the data of population among different age groups, household size or the proportion of various means of transport. Similarly, their sources will be mentioned when used. Besides, some parameters are based on certain assumption in the modelling.

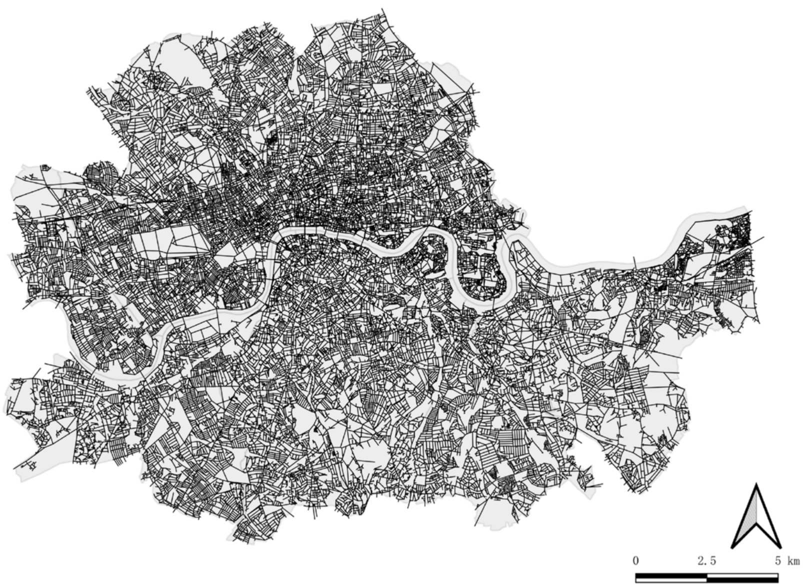


Figure 6 Segment Map of Case London

(Source: Space Syntax Ltd., modified by the author)

[1] Accessed from Space Syntax Limited.

[2] Accessed from Dr Tao Yang, currently working in China Academy of Urban Planning and Design.

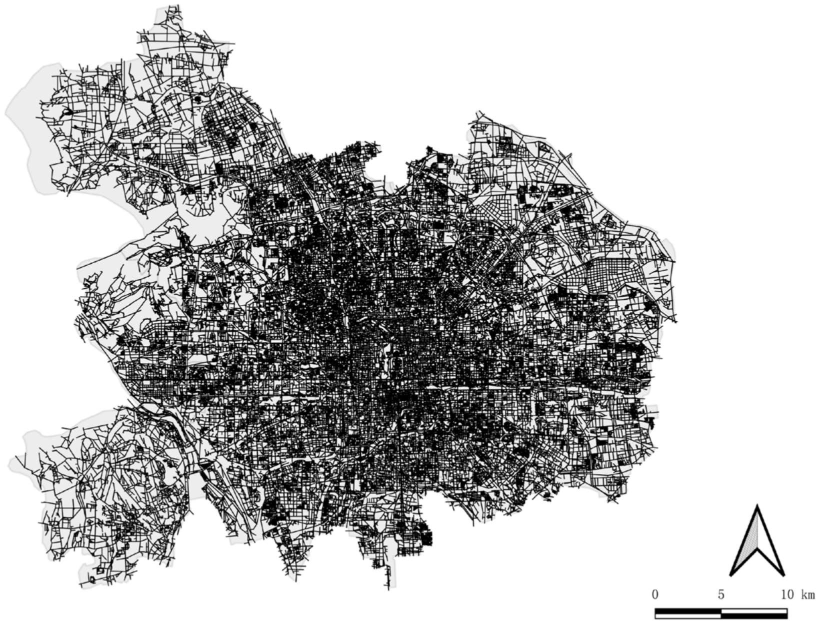


Figure 7 Segment Map of Case Beijing

(Source: Tao Yang, modified by the author)

3.3 Methodology

This paper proposed a new spatial-based model to track and simulate the movements happening in a city and track the transmission of the COVID-19. In this process, several main policies used in the UK and China will be realised in the model, so that they could be tested and quantified. Then, a comparative analysis will be carried out to examine how the pandemic would be affected under different sets of policies. In this process, a set of indices will be proposed to quantify the performance of policies in controlling the pandemic and also the side effects on normal daily lives.

As the main part of the analysis, the simulation model is written in Python, and the details will be introduced in Section 4. The full code will also be attached in the appendix. DepthmapX and QGIS play an important role in processing spatial data and making space syntax analyses. Besides, statistical analysis also goes along the data processing and is realised through Excel and SPSS.

4 Spatial modelling and measurement

To make quantitative measurement on the spatial impact of policies, a spatial model is introduced to simulate the transmission of COVID-19 and test the effect of various policies from a spatial perspective.

4.1 Spatial simulation model of the pandemic

4.1.1 Sample, time, and infectious stage

The model is built based on the segment map of London and Beijing separately. During a certain time period, the movements of all the population are to be simulated on temporal basis of $\Delta t = 0.25\ day$. Initially, every individual is allocated with a home in a segment according to the population data, and are randomly assigned with several attributes, including age, preferred means of transport, location of school or workspace, etc. To achieve these, all segments are categorised into *Origin Segments* within residential areas or *Destination Segments* within non-residential areas according to the land use data from Open Street Map. Then, the population of each London Borough or District of Beijing is allocated to all Origin Segments within according to their length (Figures 8 and 9), and the locations of workspace or schools are randomly chosen from all Destination Segments. As for the age, people are categorised into three age groups, namely children, adults, and elders. The proportions of each age group for both cases are shown in Table 5. Besides, people are assigned into multiple households. The proportions of household size are the same as official statistical data shown in Tables 6 and 7. Family members may have more chances of contact, which will be explained in Section 4.1.2.

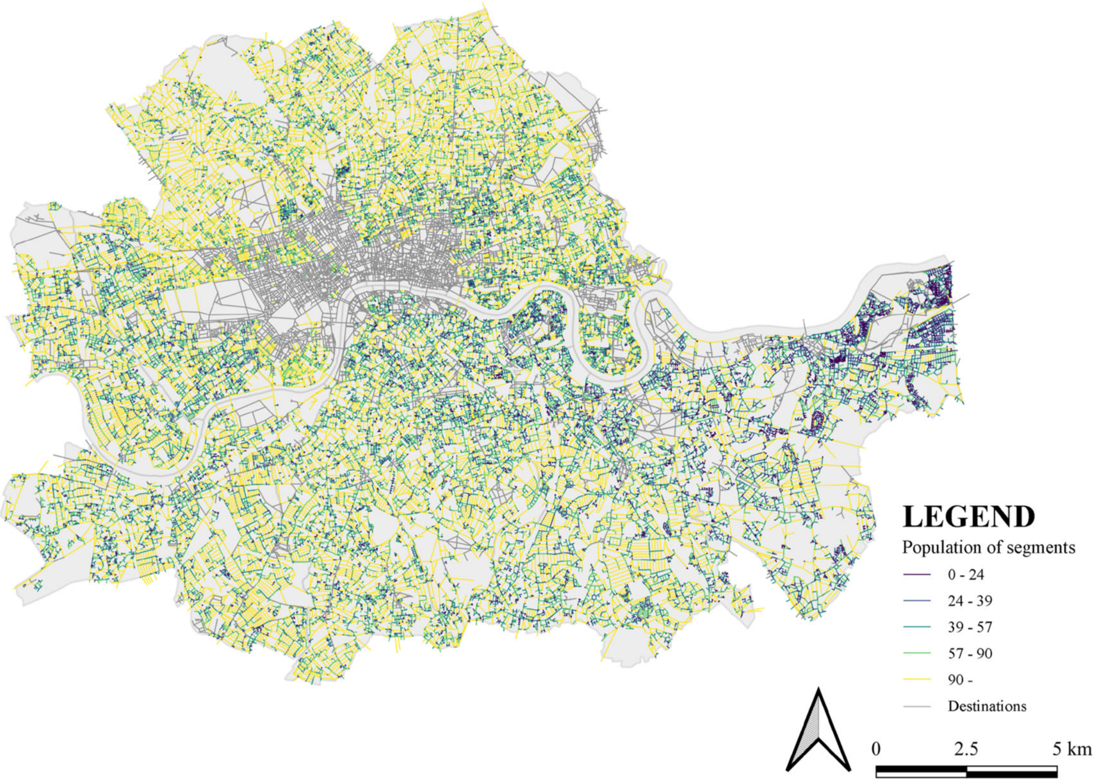


Figure 8 Population of Segments in Case London

(Source: the author)

Figure 9 Population of Segments in Case Beijing

(Source: the author)

Table 5 Proportions of Population by Age Groups in 2021

(Source: Office for National Statistics, United Kingdom; Beijing Municipal Bureau of Statistics, China)

Category	Age	Case London	Case Beijing
Children	0-15	16.1%	7.74%
Adult	16-64	74.59%	45.63%
Elders	65+	9.31%	46.63%

Table 6 Household Size of London in 2021

(Source: Office for National Statistics, United Kingdom)

Household Size	Number of Households (thousands)	Proportion	Cumulative Proportion
1	920	25.8%	25.8%
2	1105	30.9%	56.7%
3	629	17.6%	74.3%
4	623	17.4%	91.8%
5	213	6.0%	97.7%
6+	81	2.3%	100%
All	3571	100%	100%

Table 7 Household Size of Beijing in 2019

(Source: Beijing Municipal Bureau of Statistics, China)

Household Size	Proportion	Cumulative Proportion
1	23.6%	23.6%
2	31.6%	55.2%
3	25.1%	80.3%
4	10.5%	90.8%
5+	9.2%	100%
All	100%	100%

The process of infection for each individual is divided into several stages according to SIER model (Davies et al., 2020), which is shown in Table 8. At first, everyone is susceptible (S) for infection except for 2 infected individuals for seeding. Then an individual enters Exposed state (E) for 4 days upon effective contact with infected people. The individual is not infectious until the end of the stage. Afterwards, one either enters the preclinical stage (I_P) for 1.5 days and clinical stage (I_C) for 3.5 days with the probability of y_i, where there are clinical symptoms, or enters subclinical stage (I_P) with the probability of $(1 - y_i)$, where one is assumed to be only 50% infectious. Then, the infected individual is removed (R) from the spatial transmission model out of either recovery or isolation while waiting medical treatment.

Table 8 Process of Infection

(Source: the author)

	Period (days)	**Possible to be Infected**	**Infectious**
Susceptible (S)	-	Yes	No
Exposed (E)	4	No	No
Preclinical (I_P)	1.5	No	Yes
Clinical (I_C)	3.5	No	Yes
Subclinical (I_S)	5	No	Yes
Removed (R)	-	No	No

4.1.2 Activities and contacts

In every Δt, every individual is to make movement following several patterns. There are 3 ways of movement in the model. The first one is *stay* in the current location. The second one is *targeted movement*, where one moves to a target (which might be very far) corresponding to global movements. The third one is *free movement*, where one moves around within street segments adjacent to the current location. The movement choice differs according to time and age groups, which is shown in Table 9. To simulate the process of household transmission, everyone is to stay at home in the first time period of every day and go back home when the day ends. As for the two time periods in the middle of the day, every child or adult is to apply a targeted movement towards school or workspace during the second time period during weekdays, and stay there in the following period, while an elder will conduct free movement twice. During weekend, everyone is to make independent random choices twice between free movement near home and targeted movement towards a random non-residential destination.

Table 9 Daily Activities in the Model

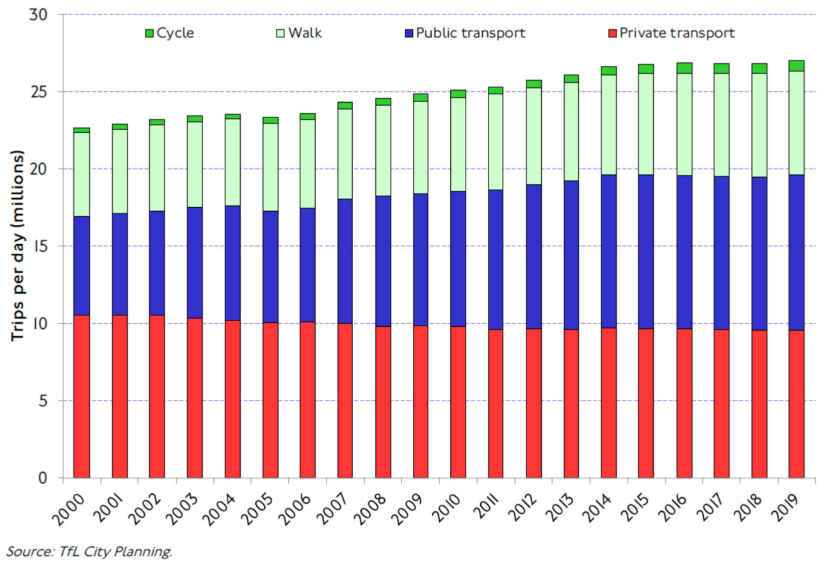
(Source: the author)

	Children or Adults	**Elders**	**Category of Activities**
Weekday 0:00-6:00	Staying at home.	Staying at home.	Household
Weekday 6:00-12:00	Targeted moving to workspace or school. (Fixed every weekday.)	Free moving near home.	Outdoor
Weekday 12:00-18:00	Staying at work / school.	Staying at home.	Indoor
Weekday 18:00-24:00	Targeted moving home.	Free moving near home.	Outdoor
Weekend 0:00-6:00	Staying at home.	Staying at home.	Household
Weekend 6:00-18:00	50% targeted moving to random segment 50% free moving	50% targeted moving to random segment 50% free moving	Outdoor
Weekend 18:00-24:00	Targeted moving home.	Targeted moving home.	Outdoor

In each time period, the numbers of *potential contacts* for each individual in Susceptible stage with infectious individuals including Preclinical, Clinical and Subclinical cases are counted. For each contact, a Susceptible individual is possible to be infected and enters Exposed stage. During the stay at home from 0:00-6:00, only contacts between family members are counted. For all other activities, two individuals are regarded to having contact once they pass by or stay in the same segment in each time period.

However, this may overestimate the number of contacts. Therefore, an extra parameter is introduced to proportionally scale down outdoor contacts, and the rate of infection used in the model is a statistical parameter rather than the actual rate of infection for each contact. These two parameters will be discussed in Section 4.1.3 below.

In targeted movement, the individual is often to travel a long distance. If such kind of movement is treated the same as wandering around, then the individual is actually *walking* through the whole city. This is not realistic, and such a long travel will affect too many segments, causing bias in simulation. Therefore, various means of transport are introduced into this model. According to Transport for London (TfL, 2020), the frequency of private transport is almost the same with that of public transport (Graph 4).



Graph 4 Estimated Daily Average Trips by Main Mode, 7-day week, 2000-2019

(Source: Transports for London, the United Kingdom)

So, in this model, everyone is randomly assigned into two groups of transport preferences with equal probabilities. The first group of individuals always use private transport for targeted movements, and the second group of individuals use public transport. Since people in private cars do not have effective contact with other drivers or pedestrians, they cannot infect others or be infected along the targeted movement.

Therefore, targeted movements with private transport only make effective contact at origins and destinations. The routes themselves are neglected. As for public transport, movement traces from origins and destinations to their nearest public transport stations are treated effective, while the routes taken by the public transport are not counted when calculating contacts (Figure 10). This is because people taking public transport do not infect or be infected by pedestrians during their travel by the public transport. However, this does not underestimate the transmission within the public transport, because the contacts among passengers have been counted twice when they gather near the stations waiting for the public transport or get off.

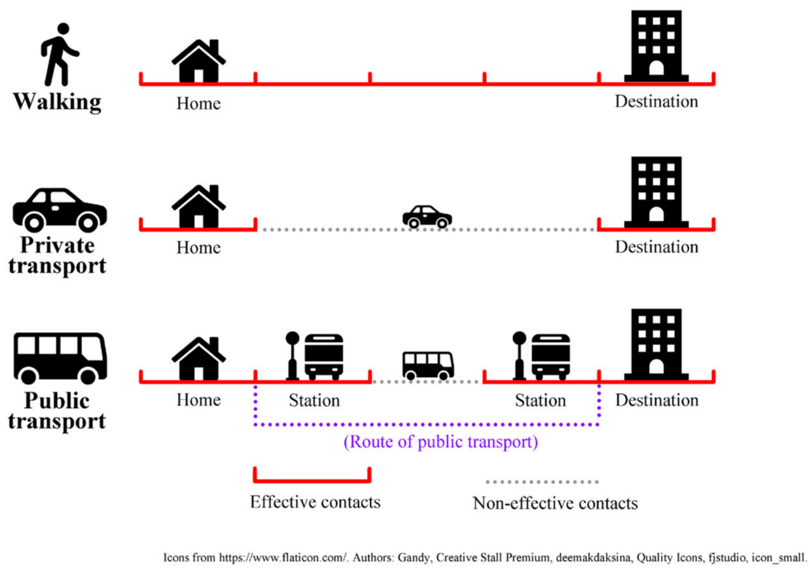


Figure 10 Means of Transport in the Model

(Source: the author)

4.1.3 Infection probability

After each period, the contacts between susceptible and infected individuals are checked. The probability of one individual gets infected is

$$p_I = \begin{cases} 1-(1-a)^{x} \; for\ indoor\ contacts \\ 1-(1-a)^{\frac{x}{b}} \; for\ outdoor\ contacts \end{cases},$$

where a indicates the basic infectious rate upon each potential contact with an infected person, and x indicates the number of infected person that one met in a certain Δt.

The distinction between indoor and outdoor contacts has been shown in Section 4.1.2. Since the model is built on the segment map of study areas, the precision of location is limited to street level. However, that two individuals passed by the same street does not mean that they are bound to meet or get close enough so that the virus is able to transmit. Specifically, this would lead to overestimation of the probability of effective outdoor contacts. Therefore, in this model, the number of outdoor encounters was scaled down to a reasonable level and so the index b is introduced to the model. The parameter b was adjusted to a certain level where the proportion of indoor infection is near 50%.

Since the value of these two factors, a and b, affects the probability of infection, they have a direct effect on the speed of the transmission. Therefore, to determine the values, the appropriate duration of pandemic should be figured out. R value means that each infected person is expected to infect R individuals during the infectious period. If there are infinite number of susceptible people, the number of the newly infected should be an exponential function of time. Let T denotes the rounds of infectious period and assume that there is one infected individual initially. Then the number of newly infected people at time T is

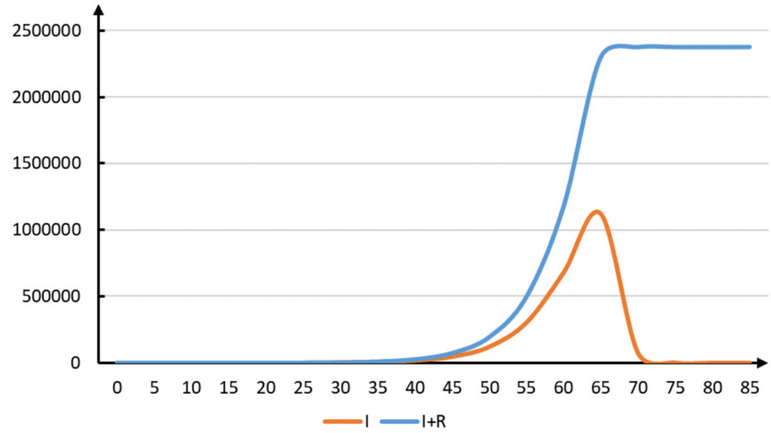
$$I_T = \begin{cases} 1\, for\ T = 0 \\ RI_{T-1}\, for\ T = 1, 2, \cdots \end{cases}.$$

However, in actual cases, the proportion of the susceptible is declining as the virus spreads. Let R_0 be the initial R value, the formula above could be amended as

$$I_T = \begin{cases} 1\, for\ T = 0 \\ R_0 I_{T-1}\left(1 - \frac{S_{T-1}}{N}\right)\, for\ T = 1, 2, \cdots \end{cases},$$

where S_T denote the number of the susceptible at T and N denote the number of the whole population. According to Davies et al., the mean value of R_0 is 2.68 (Davies et al., 2020). Based on this, the ideal graph of infection could be drawn as follows (Graph 5).

In this graph, the curve of I indicates the number of the newly infected, and the curve of $I + R$ indicates the sum of the infected and recovered people, or the cumulative



Graph 5 Logistic Curve of Infection

(Source: the author)

number of infections. It shows that given the total population of Case London and the corresponding R_0, the infection lasts for approximately 85 days. This duration constitutes the second criteria for choosing appropriate values of a and b.

With the proportion of indoor infection assumed as 50% and the duration of pandemic set around 85 days, the appropriate values of a and b can be settled. Firstly, for any value of a between 0 and 1, there exists a corresponding value of b, so that the average duration of the pandemic is near 85 days. It is obvious that the values of a and b are positively correlated. When the value of a rises, the value of b also increases, leading to the rise of the proportion of indoor infection, because the indoor infection rate is not affected by b. Then, there exists a couple of a and b, whose corresponding proportion of indoor infection is the closest to 50%. The precision depends on the set of values tested and are limited to random errors. Finally, the probability of one individual gets infected is

$$p_I = \begin{cases} 1-(1-0.02)^x \; for\ indoor\ contacts \\ 1-(1-0.02)^{\frac{x}{22}} \; for\ outdoor\ contacts \end{cases},$$

In the model above, every movement and contact within a city could be randomly simulated. Compared to the statistical models, the advantage of this spatial model is that it is easier to check the transmission not only over time but space. More relationships between the urban spatial configuration and pandemic could be revealed, and the spatial effects of various policies could be tested.

4.1.4 Randomness

Full randomness is introduced into the process of simulation. For each simulation, all individuals are randomly assigned with ages, home addresses, workspaces or schools, households, and preferred means of transport. Their activities on weekends are also random. For each scenario tested in Section 5, the simulation result is the average of several independent random simulations. Besides, to ensure that the results of simulation in multiple scenarios are comparable, a set of random seeds are controlled in the initialisation process, so that the simulation of each scenario starts from a fixed set of initial conditions.

4.2 Measurements of Simulation Results

In Section 5 below, multiple scenarios with one or more policies will be tested in the simulation, and the impact of policies will be measured in multiple ways compared with the basic model where no policy is applied.
Firstly, graphs will be used to show the trend of transmission over time. The impact of policies can be directly shown through the changes in the shapes of curves. The data of graphs come from two indices, namely the numbers of newly infected cases and current clinical cases per day. The number of newly infected cases shows the speed of transmission, while the number of clinical cases is linked to the burden of medical system.

Secondly, some indices are used for further analyses on the performance of policies. Each public policy has its own benefit and cost. While protecting public health and promoting social wellbeing in the long term, policies against the pandemic restricts movements and activities, leaving negative impact on the economy and social welfare. Therefore, policies cannot be evaluated only by its performance against the pandemic. However, since this paper is not a comprehensive study, it cannot cover multiple subjects such as economy or psychology but can only focus on the pandemic itself. For the benefits of policies, the performance of each policy will be measured by multiple indices including:

(1) The proportion of people having not been infected at the end of the model, indicating the long-term effects of the policy;

(2) The peak number of clinical cases, corresponding to the burden of medical system, and

(3) The reproduction number (R value) mentioned in Section 4.1.3, indicating the speed of transmission.

On the other hand, to measure the cost of each policy, the Coverage Index is introduced in the model. It uses the cumulative time of everyone affected by a certain policy divided by the total population and duration of the model to show the average proportion of time for each person being affected. The formula of Coverage Index is

$$CI_p = \frac{\sum_{i=0}^{N} t_i}{NT}$$

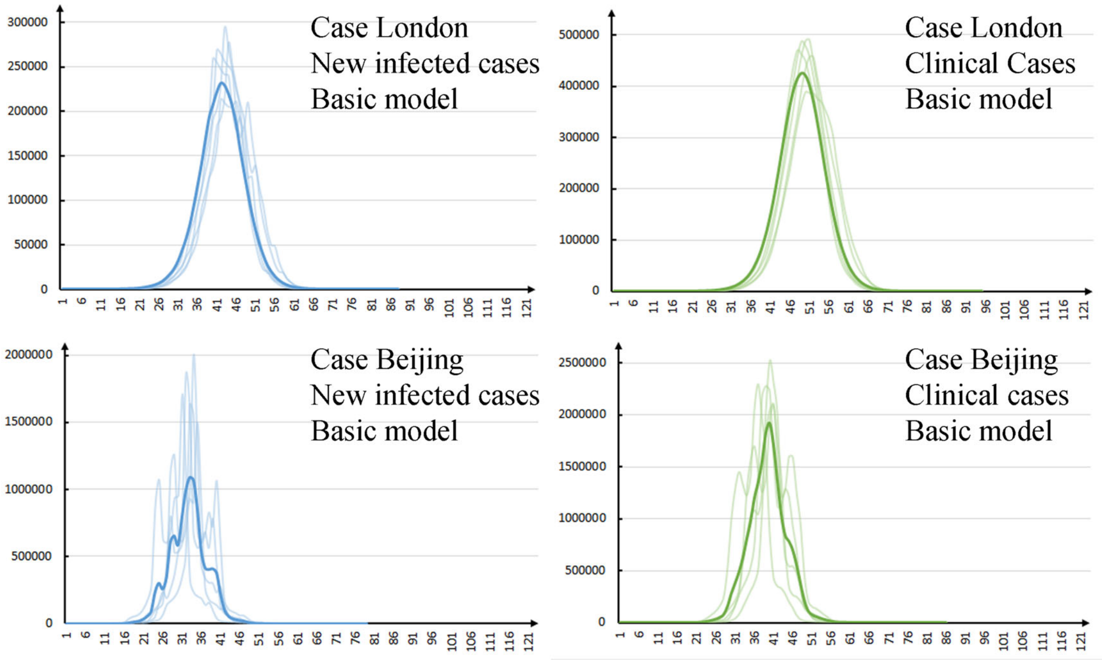
where CI_p indicates the Coverage index of policy set p. The variable t_i means the time of the ith individual being affected by the policy set. N indicates the total population and T means the duration of simulation model. Besides, some specific policies deserve special attention. For example, in some cases the government requires that the infected cases move to a unified isolation area, or total lockdown is applied to some areas. In these cases, additional data of people affected by the policies will be specifically recorded.

Finally, the spatial distribution of the infection over time will also be tracked, and the link between the spatial configuration and the spread of pandemic will be primarily analysed.

5 Results of Simulation

5.1 Basic simulation

Firstly, the basic model described in Section 4.1 was run to show the trend of the pandemic without any intervention. At the beginning of each model, 10 individuals at the same segment were set into Exposed stage as the initial infected cases. The models of both study areas were run 100 times independently. The trend of newly infected cases and clinical cases for both study areas are shown in Graph 6. The trend of newly infected cases is measured with the number of people who enters Exposed stage each day. The thick lines in the graphs show the average of all 100 simulations, and other lines are the results of some samples. It is clear that in Case London, the infection quickly reaches the peak after approximately 1.5 months. The spatial distribution of the infected in the end of 5^{th}, 6^{th}, $7^{th,}$ and 8^{th} week of the first simulation of Case London are shown in Figure 11. Each segment is marked with the proportion of people having been infected. The location of the initially infected cases is also marked with points. Due to the daily commute, the spatial spread of the pandemic does not follow spatial proximity. The virus quickly spread into most of the communities in the first month and caused citywide outbreak later. The infection got even worse through local movement and household contacts. After the first two months, most people have been infected.



* Results for Beijing and London are presented at different scales

Graph 6 Basic Model, Case London, and Case Beijing

(Source: the author)

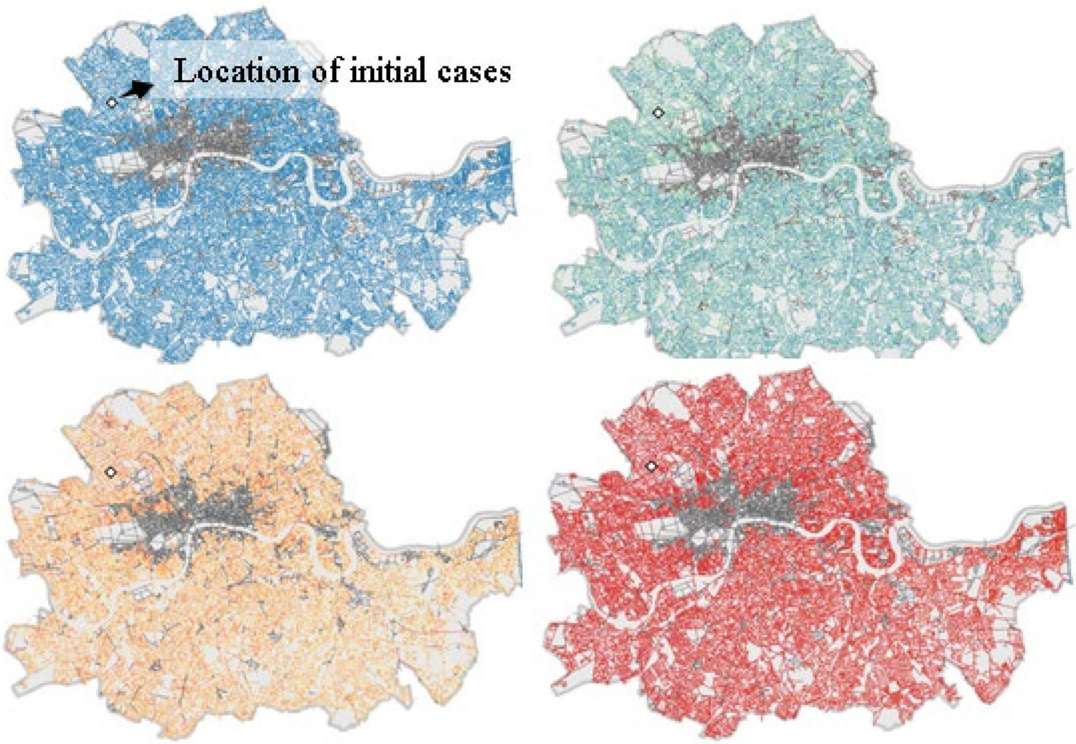


Figure 11 Schematic Figure of the Spatial Process of Transmission

(Source: the author)

5.2 Simulation of single citywide policies

To illustrate the policies simulated in this paper, the effects of several policies were tested for Case London in this Section. The policies tested in the model are shown in Table 10. In each simulation, a single policy was applied to the whole area. The first set of policies is about isolation. From self-isolation and centralised isolation to regular nuclear test, the degree of intervention increased step by step. Similarly, two scenarios of wearing face masks were simulated. In one scenario, people wear face masks only when they are outdoors, while in another scenario, they also wear face masks during indoor activities. Other two policies from UK were also tested. The rule of six indicates a limitation to gatherings, and the regional restriction is correspondent to the movement restrictions applied during national lockdown of UK.

Simultaneously, the effect of time lag was also tested. Here, four sets of time for triggering regulations are tested. Firstly, after the outbreak of the pandemic, it often takes time for the government to assess specific circumstance and make decisions. Even if people are alarmed in advance by the pandemic overseas, it is not reasonable that any compulsory policy takes place before the pandemic actually occurs. In this case, it could only be expected that non-mandatory recommendations such as wearing face masks or keeping social distance may be announced in advance. However, since the obedience cannot be guaranteed, these guidance or recommendations cannot form effective protection for the country. Therefore, it is still necessary to test the impact of time lag in executing policies. In this set of simulation, the time lag was regarded as the interval between the time when the first clinical case was found to the announcement of compulsory policies. As are shown in Table 11, several cases of time lag were tested, where the policies were triggered immediately when the first clinical infection case was found, or one week, two weeks or four weeks later.

Table 10 Policies Tested Individually in the Model

(Source: the author)

Policy	**Description**
1-1 Self-isolation	Any individuals found with clinical symptoms are isolated at home until recovery.
1-2 Unified isolation	Clinical cases are moved to segregated isolation areas and receive treatment immediately.
1-3 Centralised isolation + regular nuclear test	The above policy plus regular nuclear test every three days.
2-1 Wearing face masks in outdoor scenarios	The infection rate is reduced by 50% with the utilisation of face masks. This applies with outdoor contacts.
2-2 Wearing face masks in all scenarios.	The protection of face masks applies with both indoor and outdoor contacts.
3 Rule of six	The maximum number of contacts per Δt is limited to 5.
4 Lockdown.	People are restricted at home for two months.

Table 11 Time Lags Tested in the Model

(Source: the author)

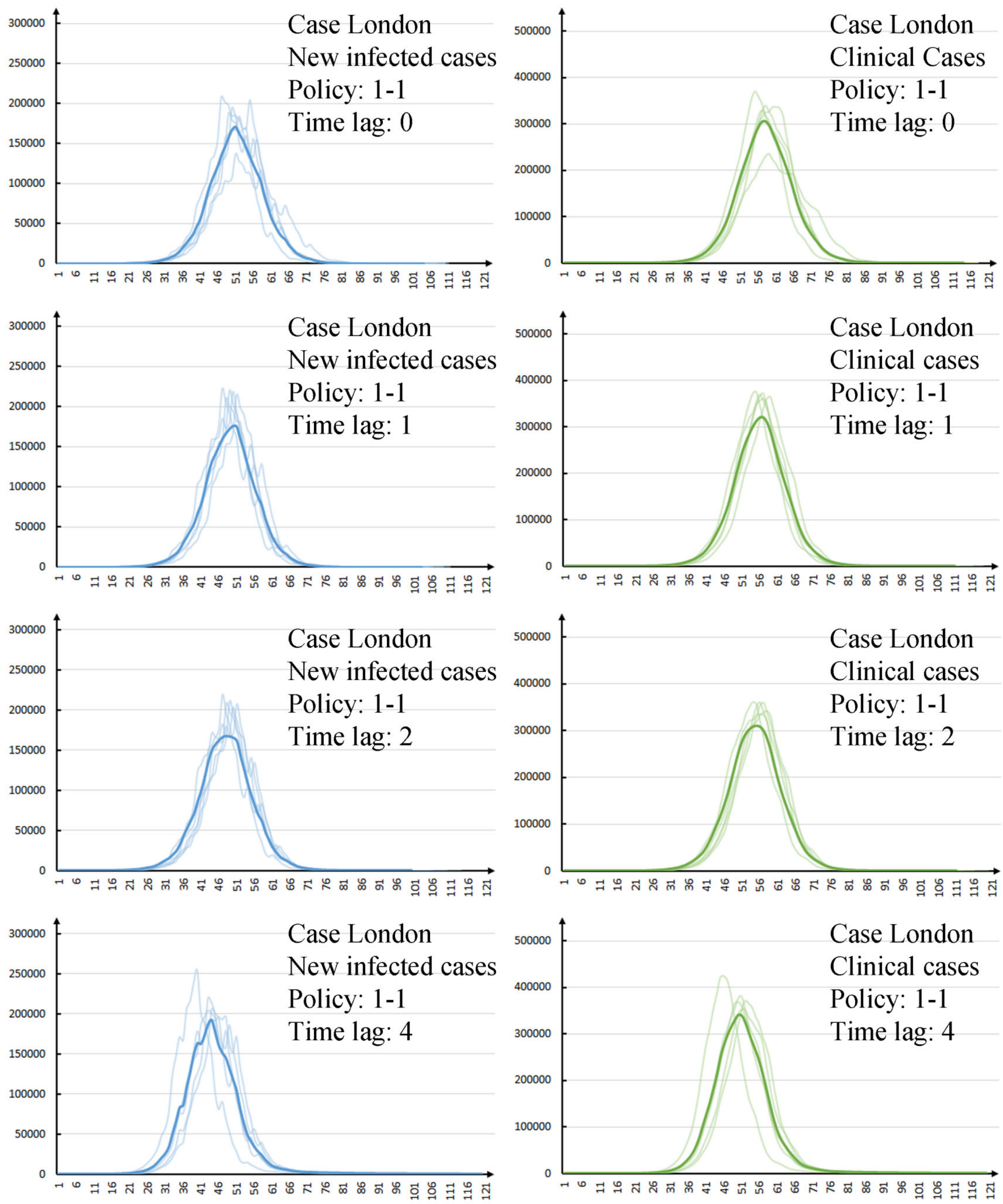
Time Lag	**Description**
0	Policies are triggered once the first clinical case is found.
1	Policies are triggered one week after the first clinical case is found.
2	Policies are triggered two weeks after the first clinical case is found.
4	Policies are triggered four weeks after the first clinical case is found.

5.2.1 Isolation policies

Firstly, the effect of self-Isolation, denoted as Policy 1-1, was tested. This is regarded as a general policy, which indicates that any individuals found with clinical symptoms are required to self-isolate at home until recovery. This is based on the policy which the UK government announced on 12 March 2021 that anyone with coronavirus symptoms should stay at home for at least 7 days. In this model, the correspondent Clinical stage lasts for 3.5 days. After this stage, they would be neither infectious nor possible to be infected. So, there is no difference between the UK policy and the action taken in the simulation. In the simulation with Policy 1-1, any individual that enters Clinical stage does not engage in the movement and contacts during the day, but is still possible to infect the family members.

The trend of newly infected cases and clinical cases for both study areas are shown in Graph 7. The curves of infection were flattened with lower peak and wider span. On average, this policy significantly decreases the peak number of new infection and clinical cases by 27% and 29%. However, the proportion of people having been infected does not decrease significantly. Most of the people got infected at the end of the simulation, although the process took longer time than that without any intervention. Therefore, the application of self-isolation does not stop the pandemic from spreading but only slowed down the transmission.

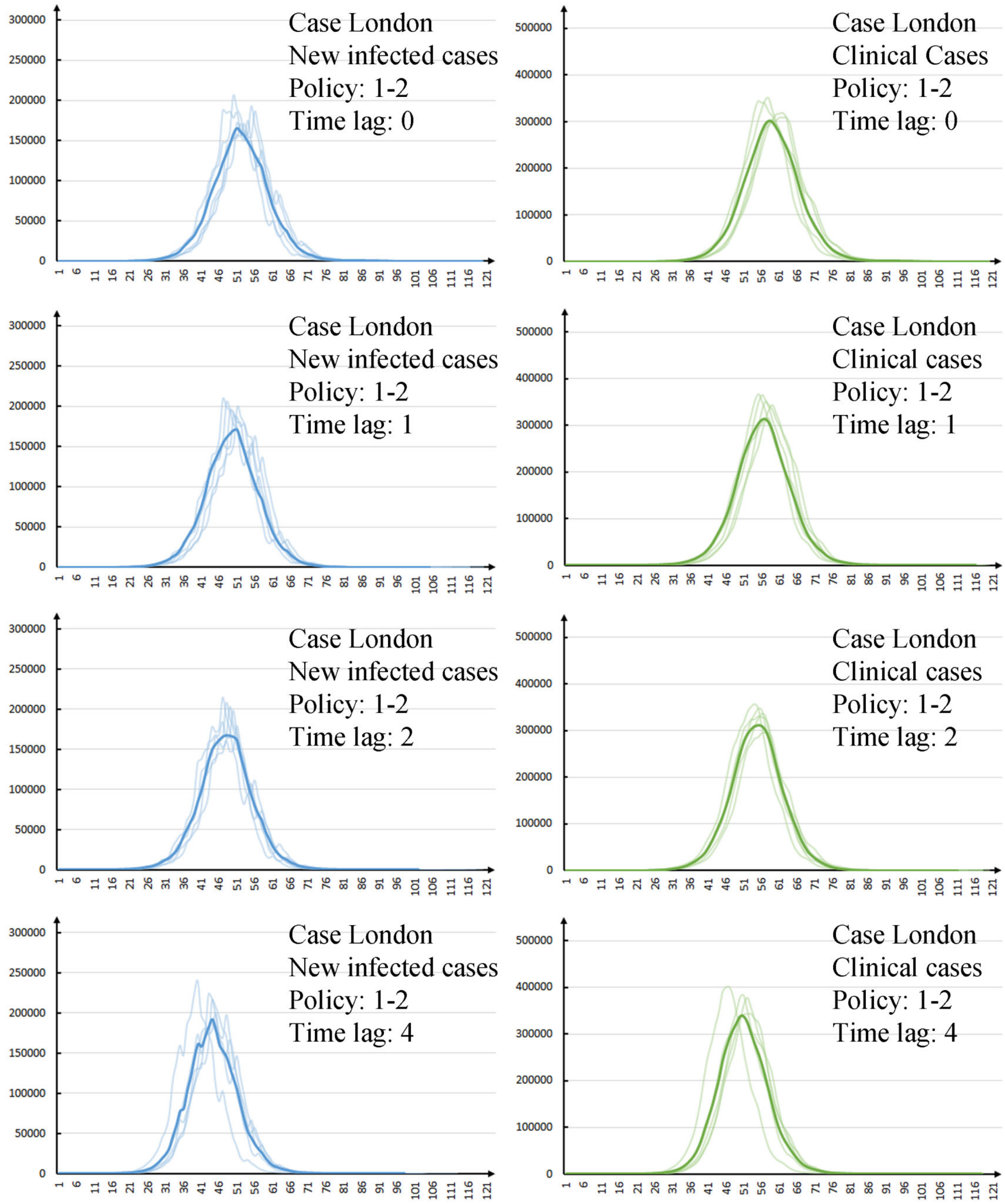
Afterwards, an enhanced version of the policy above was tested, where all clinical cases are immediately sent to a specific segregation area and receive medical care and treatment. This policy is based on Chinese policies and is denoted as Policy 1-2. The difference between this policy and Policy 1-1 is that unified isolation cuts down any way of transmission between the clinical cases and others, including potential transmission at home. In this way, this policy is expected to have a better control over the pandemic. However, as are shown in Graph 8, the average results are nearly the same as those of Policy 1-1. This is because the impact of obedience is not considered in the simulation. It is always assumed that any policy is completely obeyed in the simulation, but in reality, violation is still possible under the requirement of self-isolation, so the effect of self-isolation policy is slightly overestimated. On the other hand, since all clinical cases are moved to the hospital in the case of unified isolation, unexpected transmission is less likely to happen.



Graph 7 Simulation Results of Policy 1-1, Case London

(Source: the author)

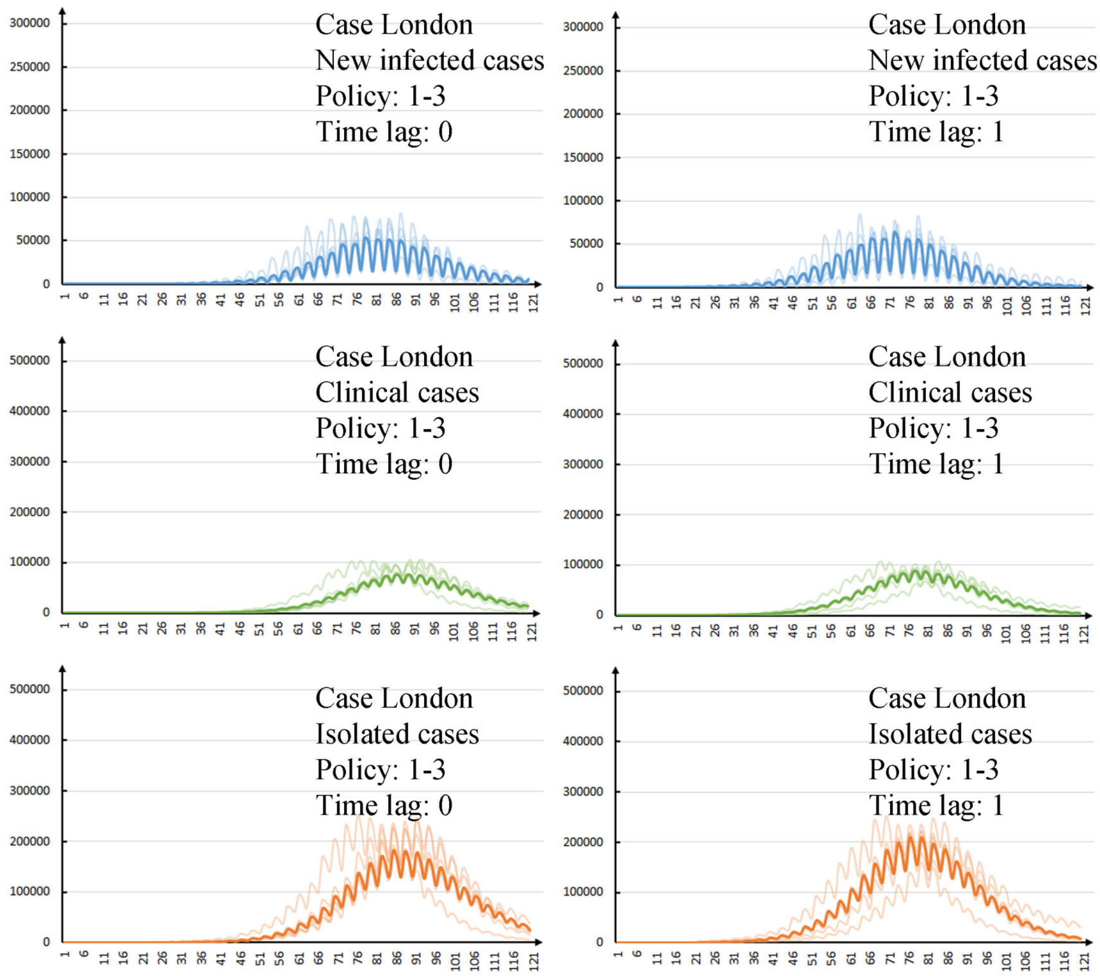
Besides, the expense of carrying out unified isolation should also be considered. Since each clinical case are isolated, the number of people under unified isolation is the same as the number of clinical cases shown in Graph 8. It can be expected that treating every new clinical case applies more pressure to the medical system and public budget. If not accompanied by other policies, the medical system will quickly be overwhelmed by rapidly increasing patients.



Graph 8 Simulation Results of Policy 1-2, Case London

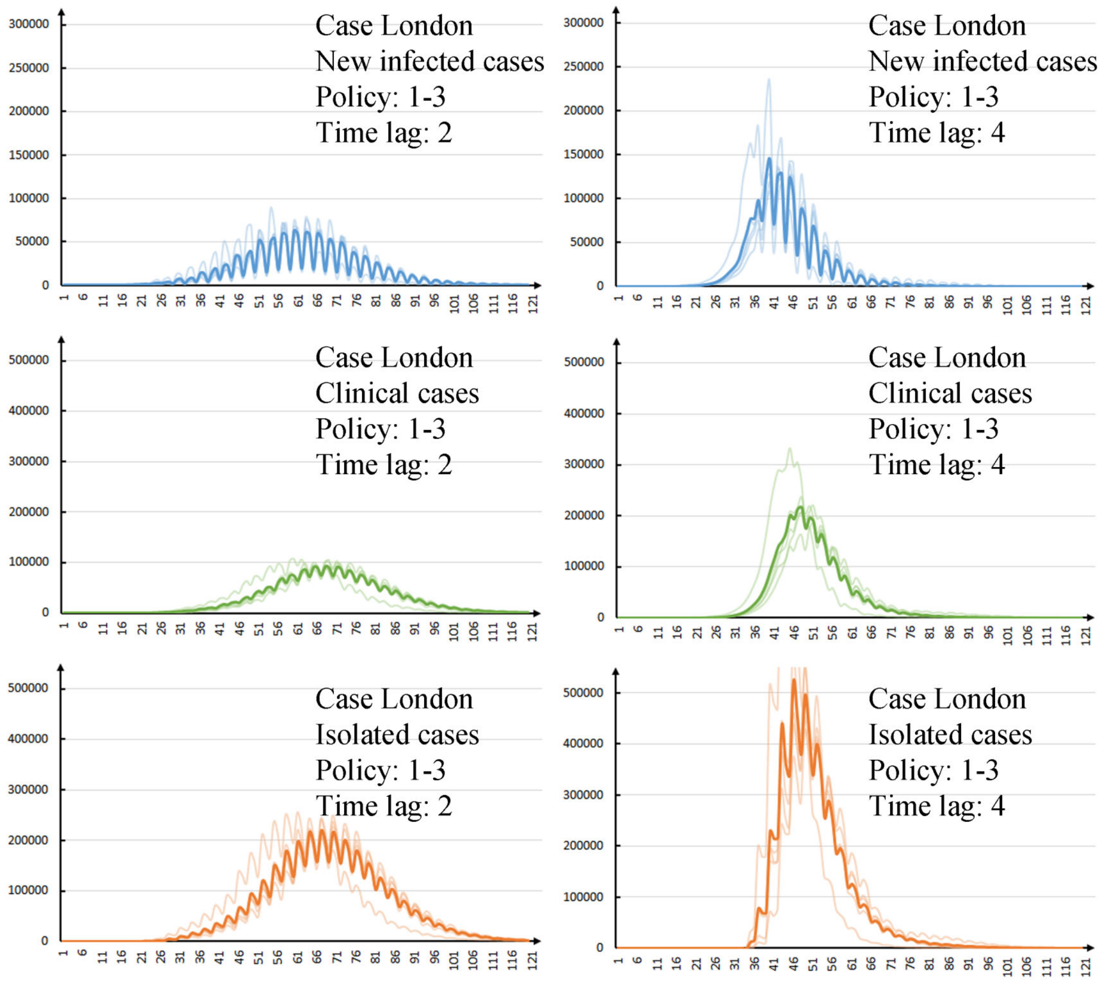
(Source: the author)

In the third case of Policy 1-3, another Chinese-based policy is applied besides the rule of unified isolation. Every three days, everyone is required to take a nuclear acid test and all infectious individuals, including ones in Preclinical, Clinical and Subclinical stages, are detected, and moved to specific isolation areas. As is shown in Graphs 9 and 10, the spread of the pandemic is significantly slowed down compared to the results of Policies 1-1 and 1-2. By broadening the range of detection and isolation, the sources of infection are blocked more efficiently, resulting in the significant decrease of infection. Therefore, it is surprising that although the range of isolation broadens from Clinical cases to all infectious individuals, the number of people under isolation has in turn decreased.



Graph 9 Simulation Results of Policy 1-3 (a), Case London

(Source: the author)



Graph 10 Simulation Results of Policy 1-3 (b), Case London

(Source: the author)

5.2.2 Wearing face masks

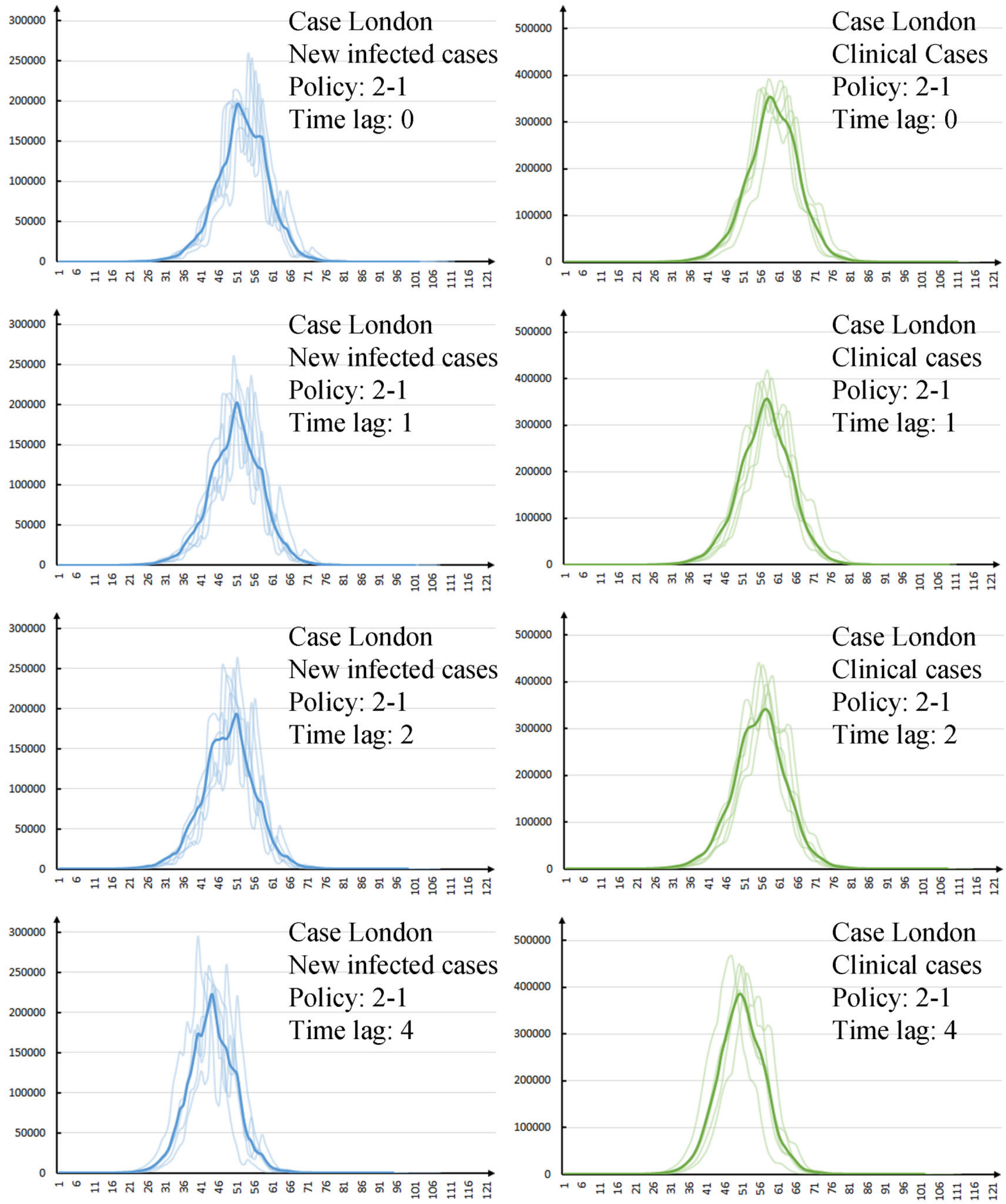
Secondly, the effect of wearing face masks was tested. This is realised by adjusting the infection rate upon contact, so it can also represent any other intervention which does not affect movements and gatherings, for example social distancing or disinfection. Here two scenarios were considered. In the first scenario denoted as Policy 2-1, which is common worldwide, everyone is required to wear face masks during outdoor activities. However, they do not wear face masks when they work or study indoors, or stay at home. In the second scenario denoted as Policy 2-2, everyone also wears face masks during their stay at workspace or schools. In both scenarios, the average infection rate per contact was assumed to be cut down from 2% to 1%, namely a half.

Graph 11 shows the trend of the newly infected and clinical cases in the first scenario. On average, the peak of the curve has been significantly decreased, showing that the speed of the transmission has been greatly slowed down. However, people are not protected during their indoor contact in workspace or schools in weekdays. So, the performance of Policy 2-1 is similar to that of Policy 1-1 and 1-2, where only part of potential virus transmission is blocked. Besides, lack of intervention with indoor contacts also brings much uncertainty to the effect of Policy 1-1. It is shown in the graph that there exist some cases where the peak of infection is almost the same as the basic model without any intervention.

Graph 12 shows the result of the second scenario, where people are required to wear face masks both indoors and outdoors. It is clear that the pandemic was supressed significantly. Both the average effect and stability are better than the case where only outdoor activities are protected. However, it is not often the case that this policy is executed with complete obedience. Although the improvement may not be as much as Policy 1-3, wearing face masks has its own advantages. Compared to other policies which restricts movements or contacts, wearing face masks provides significant protection while leaves the least effect on daily lives. In this way, wearing face masks and other similar policies are the most economic when dealing with the pandemic and should be primarily considered. The benefits and costs of other policies and especially certain combinations of policies will be further discussed below.

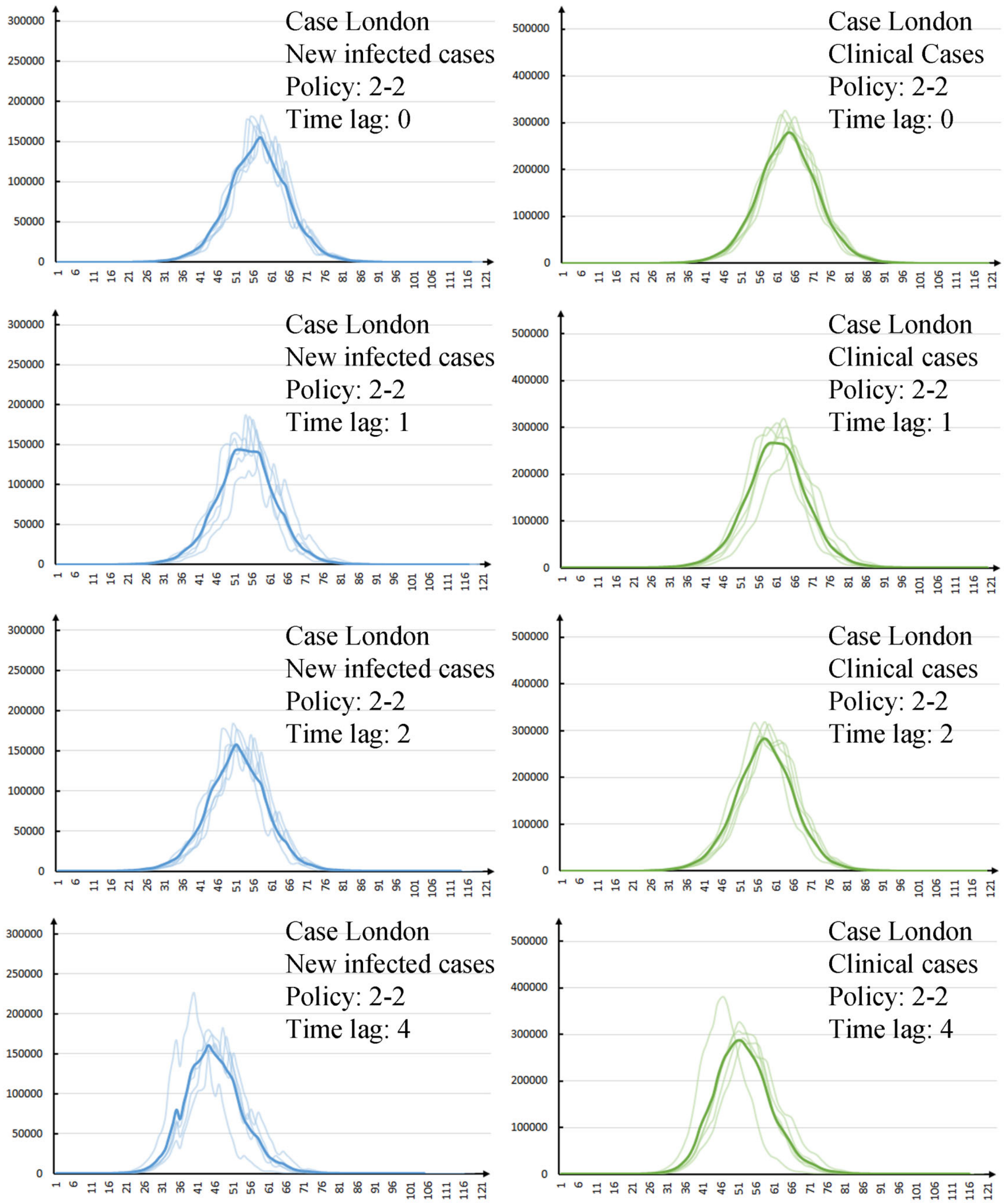
5.2.3 Rule of six and overall lockdown

Finally, two policies based on the UK policies were tested. The Rule of six, denoted as Policy 3, is that any gathering of more than six people is prohibited. In this model, this policy was tested by applying the rule that anyone can only meet up to 5 other people in every Δt. In this case, the effect of the policy might be overestimated because the original rule does not limit the number of gatherings for an individual. It is possible that one person meets 5 people for a short period of time and then goes immediately for another gathering of 5 other people, resulting in 10 effective contacts in a certain time period, but this will not happen in the simulation. Graph 13 shows the trend of the newly infected and clinical cases. It is shown that the pandemic has been significantly supressed as previous cases.



Graph 11 Simulation Results of Policy 2-1, Case London

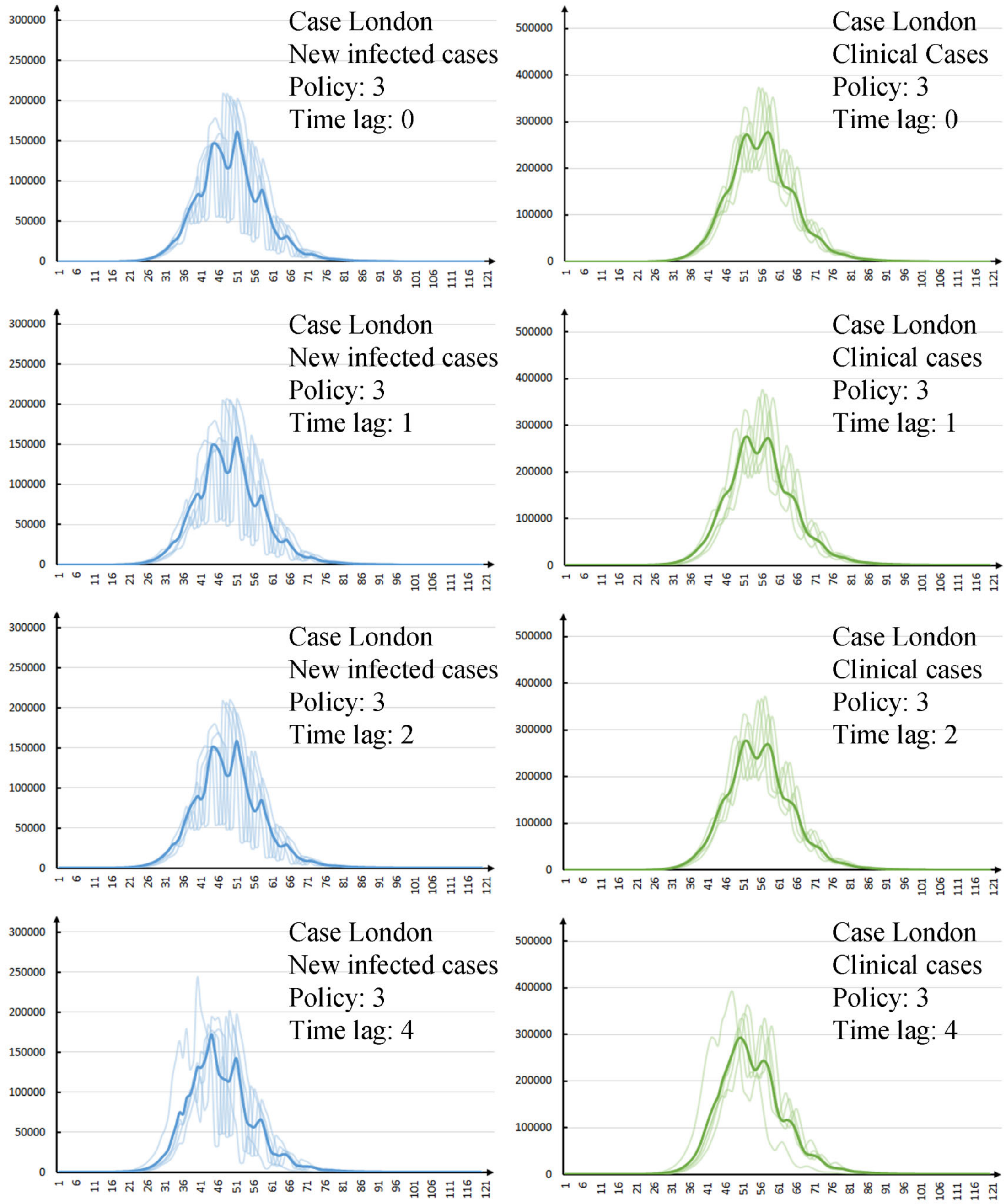
(Source: the author)



Graph 12 Simulation Results of Policy 2-2, Case London

(Source: the author)

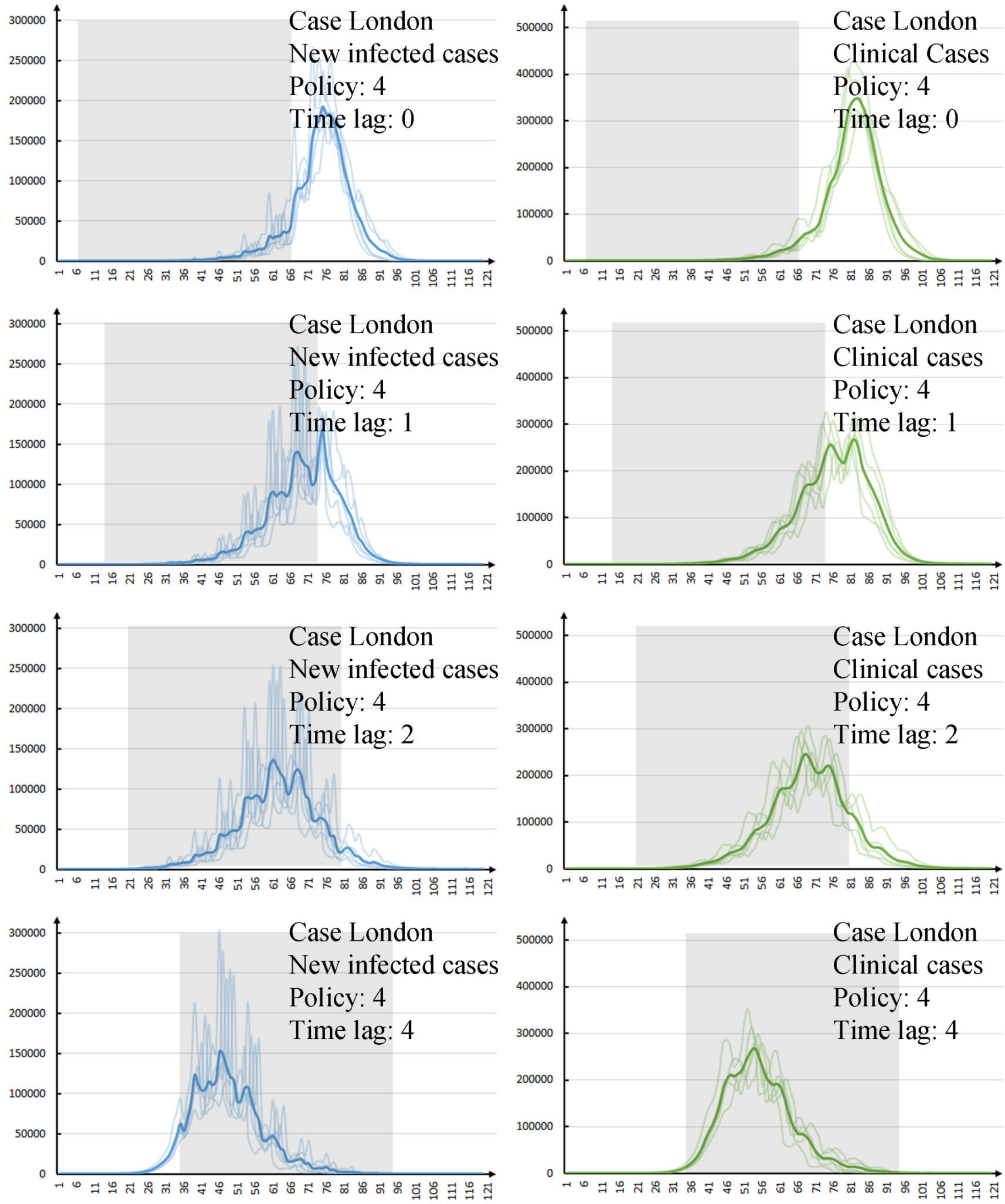
Another policy denoted as Policy 4, lockdown, is self-explained. During the lockdown, people are not allowed to leave home without a reasonable excuse. Although this policy seems more effective than others, it would leave a great negative impact on the economy and daily lives. Therefore, it cannot last for a long time. The total duration of national lockdown in the UK was 27 weeks out of one year, which means that the



Graph 13 Simulation Results of Policy 3, Case London

(Source: the author)

duration of lockdown takes up nearly half of the time. So, in this model, a simplified rule of regular lockdown is applied. From the beginning of Policy4, an overall lockdown is executed every other two months, and each lockdown lasts for two months. During the lockdown, all movements with long distances are cancelled. Since it is not possible to stop all movements for months, necessary movements are still allowed in practice, for example movements for daily necessities. Therefore, in this model, one movement within the segment per day is allowed for each individual, and people are allowed to move around neighbouring segments on Sunday.



* Lockdown periods are marked in dark backgrounds.

Graph 14 Simulation Results of Policy 4, Case London

(Source: the author)

The results of simulation with Policy 4 are shown in Graph 14, where the periods of lockdown are marked in dark backgrounds. It is shown that the peak of the infection curve is significantly postponed, and the peak number of the infection slightly decreases under the overall lockdown. The policy does not completely stop the transmission because a few movements are still allowed. Although the transmission is controlled during the weekdays, the virus keeps spreading on weekends. So, there is a large fluctuation in the curve of newly infection of each sample, which is shown in light blue curve in the graph. The average infection also shows a stepwise upward trend, indicating a phased increase of infection. This simulation underestimates the performance of the overall lockdown in reality, since the time when people go out is focused on Sunday rather than scattered into all seven days of a week, leading to more potential contacts.

5.3 Application of multiple policies

In this section, two sets of combined policies were tested on both Case London and Case Beijing. These two policy sets are originated from the policies applied in the UK and China. However, considering the fact that only part of policies was discussed in this paper and lots of details were simplified in the simulation, the results cannot be used to represent, evaluate, or judge the performance of the governments' response to the pandemic. In the real world, the policies depend on various circumstances such as medical progress, public budget, or national economy. This paper only considers the effects of policies on blocking the transmission.

As are shown in Table 12, various policies discussed in Section 5.2 are included in these two sets. Since wearing face masks is always recommended during the pandemic, it is included in both policy sets. Besides, Policy Set A requires self-isolation, rule of six and movement restriction, aiming at reducing movements and contacts generally while leaving the least impact on normal daily lives. Policy Set B tends to be stricter, requiring regular nuclear acid test for everyone. Once an infectious case is found, an additional nuclear acid test will be applied to people in the segment of the infection, and all infectious individuals will be sent to a unified isolation area until recovery. Moreover, all segments neighbouring to the segments where a case is found are locked down until no new case is found for 14 days. Everyone in the lockdown areas cannot move out of home, which equals to self-isolation. This set of policies aims at stopping the transmission to the maximum degree and end the pandemic as soon as possible, so that the economy and daily lives can go back on track. However, the process of coping with the pandemic might have a great negative impact on normal daily lives, especially for those who live in the segments locked down for a long time. It can be expected that the duration of the pandemic may be longer than the scenarios discussed in Section 5.2 where only one policy is applied at a time, since the spread of the pandemic can be slower under the restriction from multiple policies. So, the data of one year were tracked in the simulation of this section. Since policies against the pandemic cannot be long-term policies, they may be changed frequently according to the circumstances. Therefore, it is not valuable to track the effects longer than one year. If the pandemic ends within one year in any simulation, the model would stop along with the policy sets. Correspondingly, the time lags tested are also adjusted to 0, 2, 4 and 6 weeks.

Table 12 Policy Sets Tested in the Model

(Source: the author)

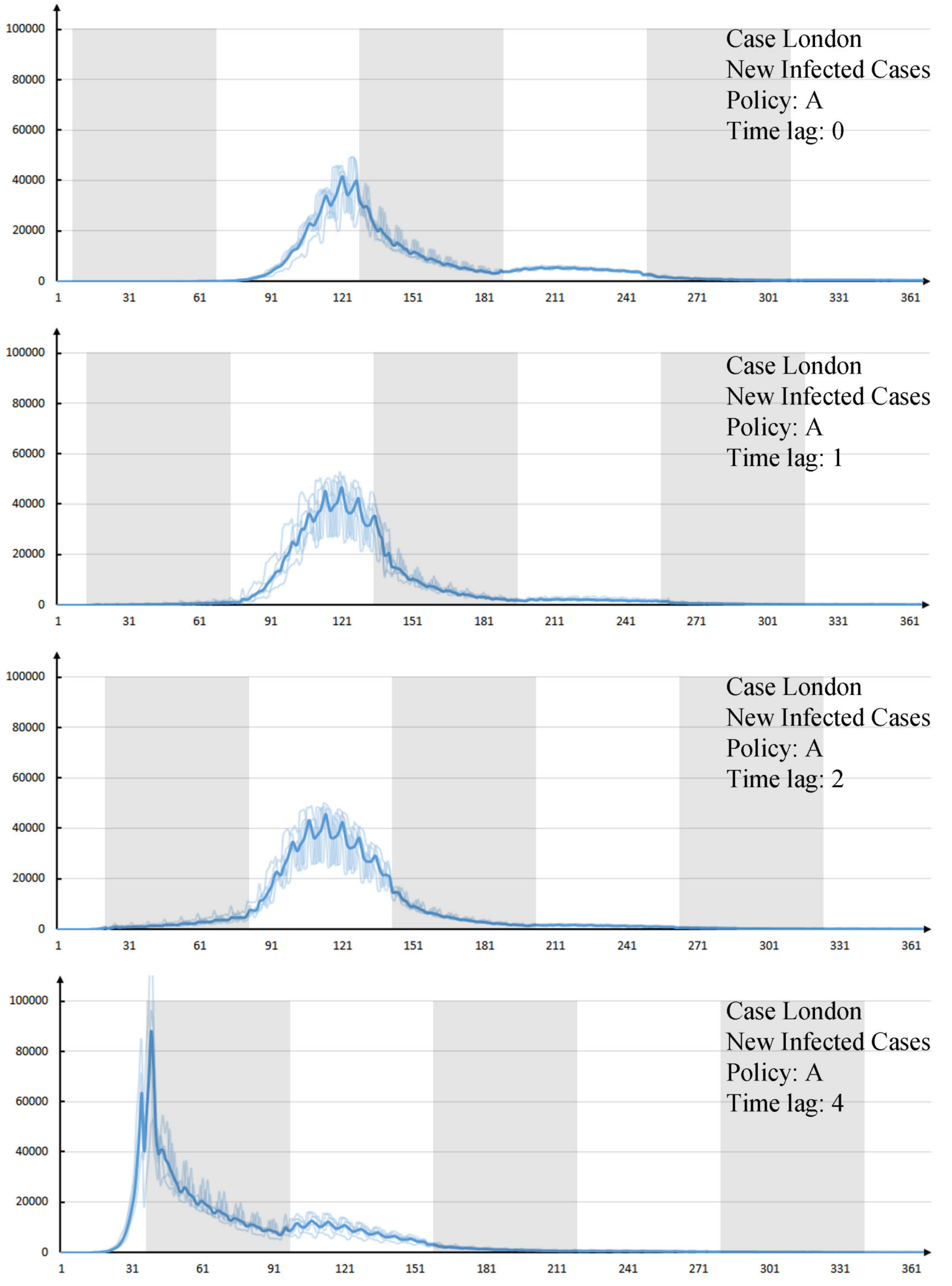
Policy Set A	**Policy Set B**
• Self-Isolation for clinical cases. • Face masks required in all contacts. • Maximum gathering of 6 people. • Overall lockdown for 2 months with an interval of 2 months.	• Unified isolation for all infectious cases plus regular nuclear acid test. • Lockdown and additional nuclear acid test for neighbouring segments where an infection is detected. • Face masks required in all contacts.

The results of simulation are shown in Graphs 15 to 26. To have them understood better, the contents of graphs are shown in Table 13 ahead of all graphs. The horizontal axes are stretched into one year and the vertical axes are also stretched, so the number of the infection are actually better than the scenarios where only single policy is applied although the curves in the graphs may seem high.

Table 13 Contents of Graphs Below

(Source: the author)

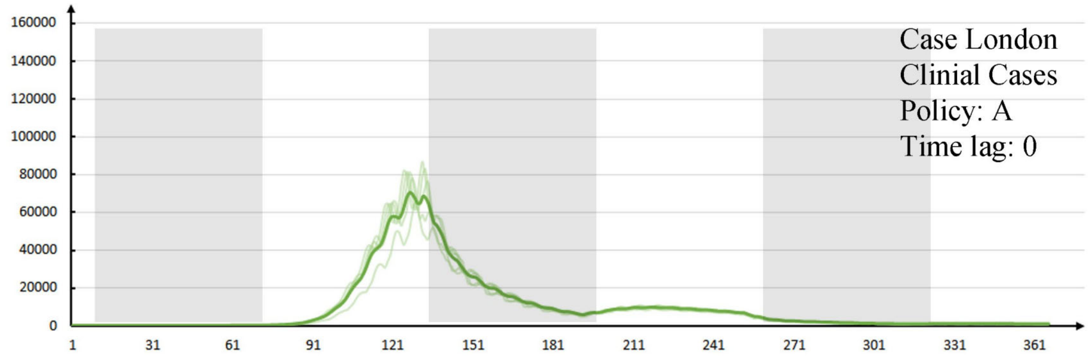
Content	**Case London**	**Case Beijing**
Policy A, New Infected Cases	Graph 15	Graph 21
Policy A, Clinical Cases	Graph 16	Graph 22
Policy B, New Infected Cases	Graph 17	Graph 23
Policy B, Clinical Cases	Graph 18	Graph 24
Policy B, Isolated Cases	Graph 19	Graph 25
Policy B, Lockdown Coverage	Graph 20	Graph 26

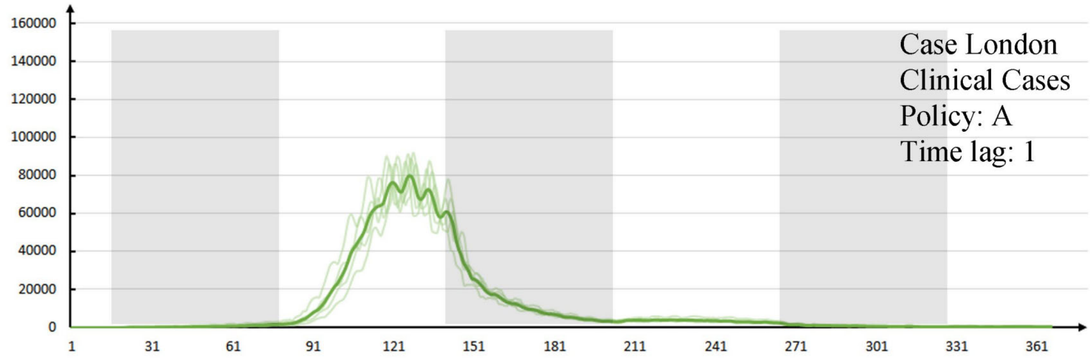


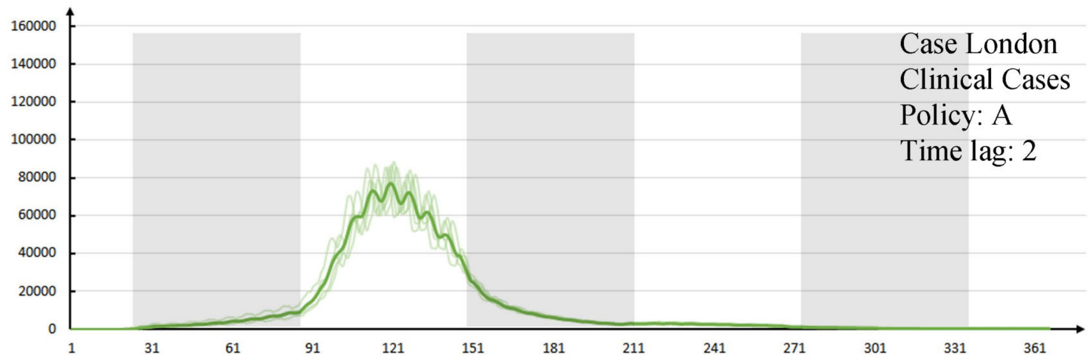
* Lockdown Periods in Policy 4 are marked in dark backgrounds.

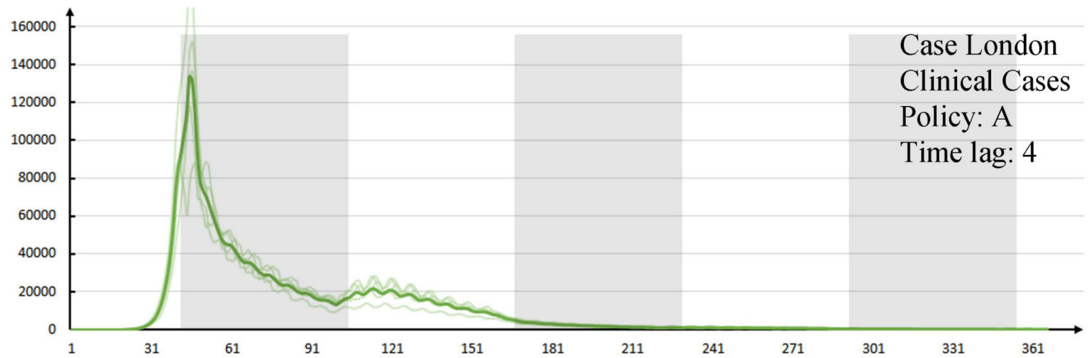
Graph 15 Simulation Results of Policy Set A, Case London (a)

(Source: the author)





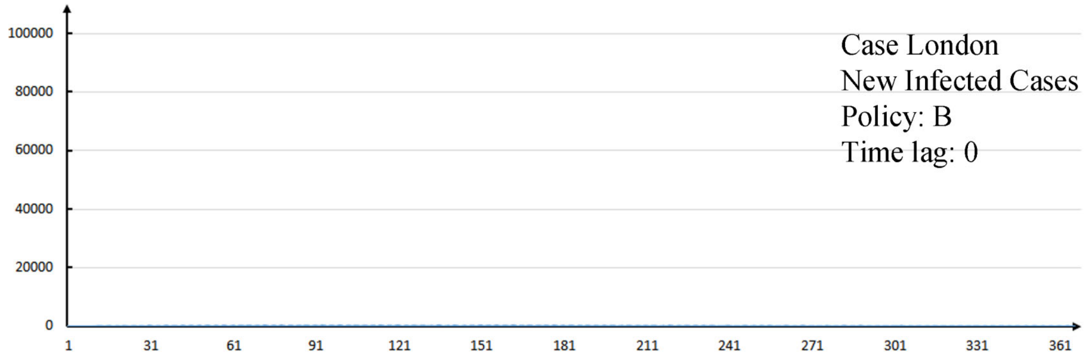


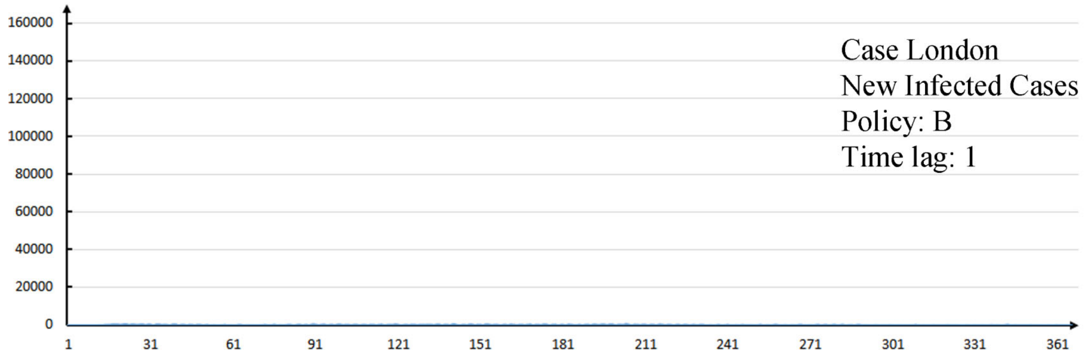


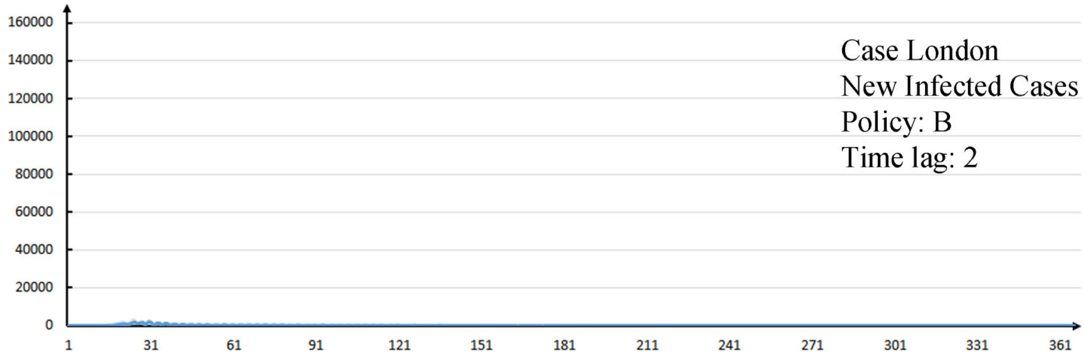
* Lockdown Periods in Policy 4 are marked in dark backgrounds.

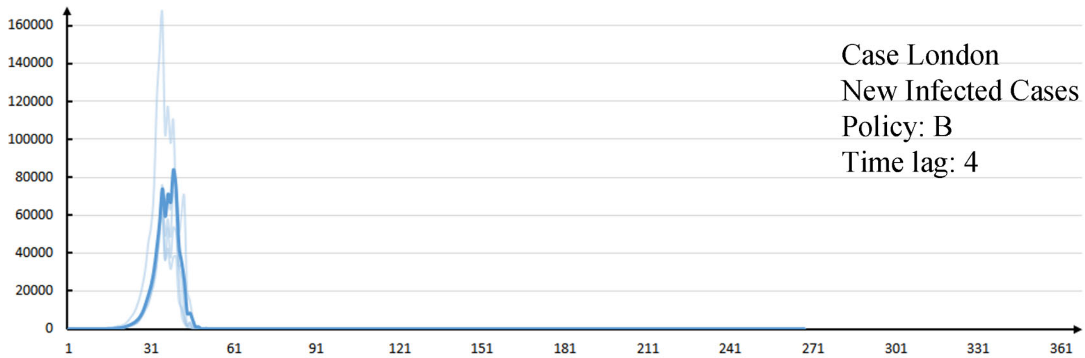
Graph 16 Simulation Results of Policy Set A, Case London (b)

(Source: the author)



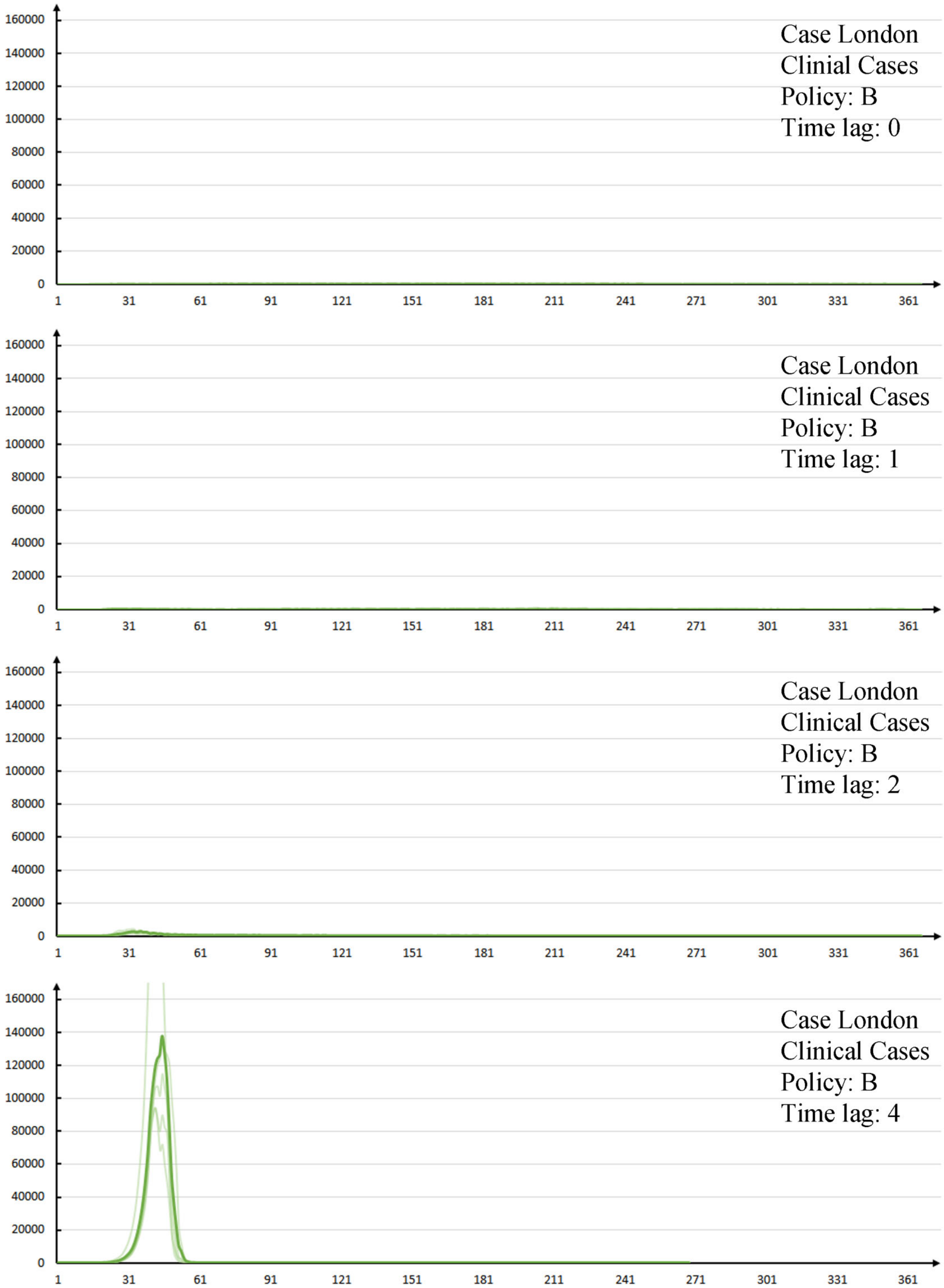






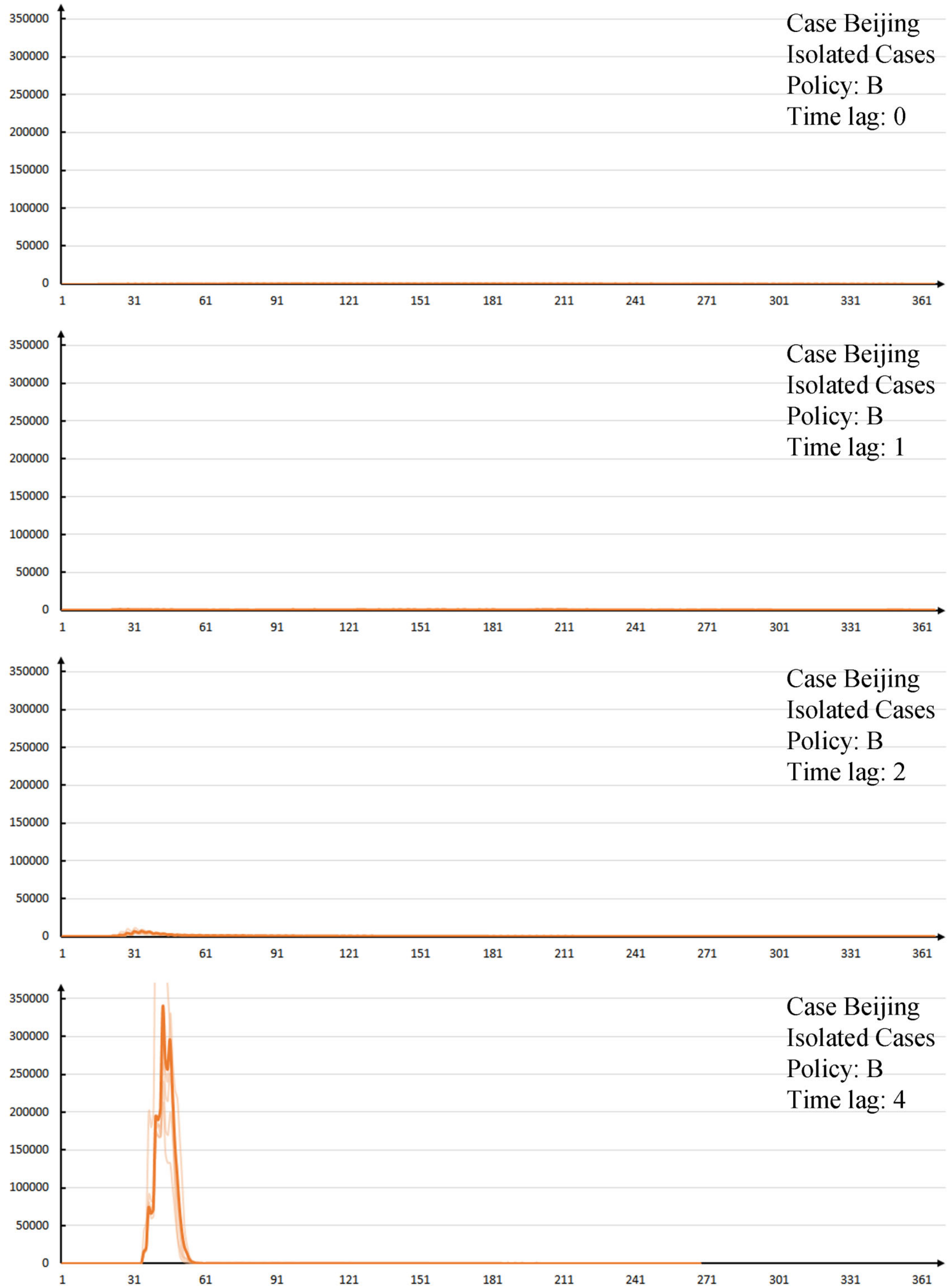
Graph 17 Simulation Results of Policy Set B, Case London (a)

(Source: the author)



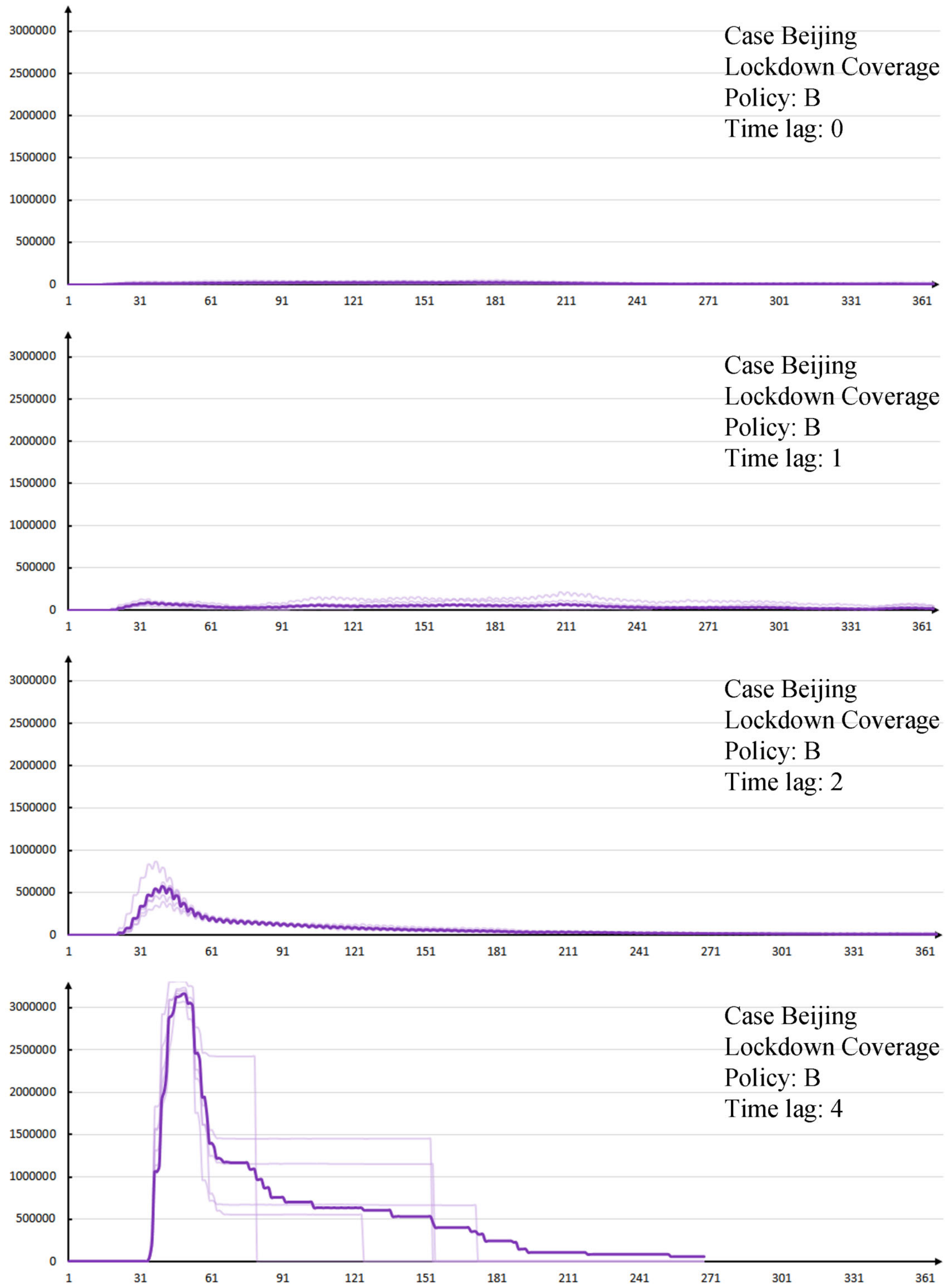
Graph 18 Simulation Results of Policy Set B, Case London (b)

(Source: the author)



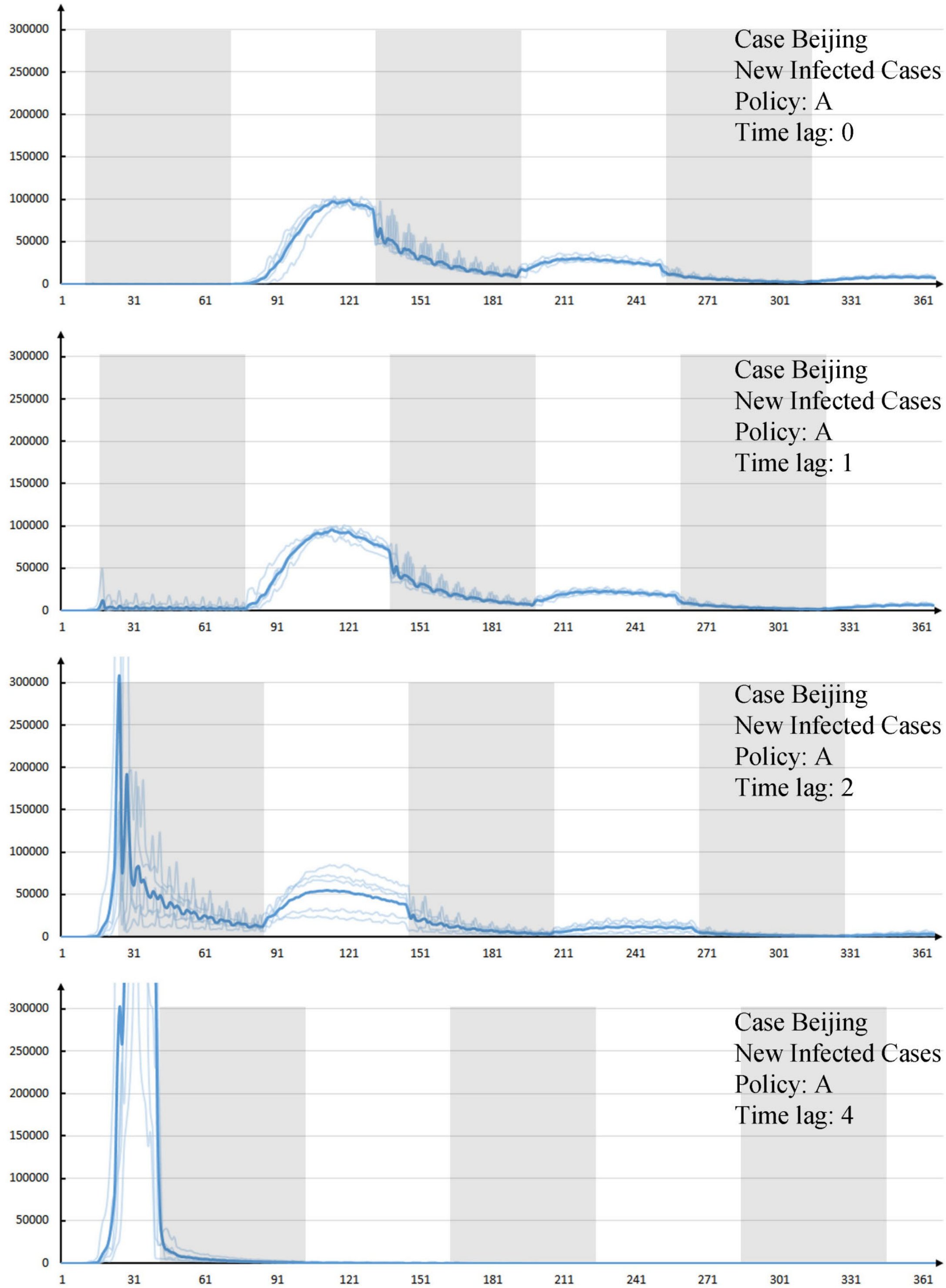
Graph 19 Simulation Results of Policy Set B, Case London (c)

(Source: the author)



Graph 20 Simulation Results of Policy Set B, Case London (d)

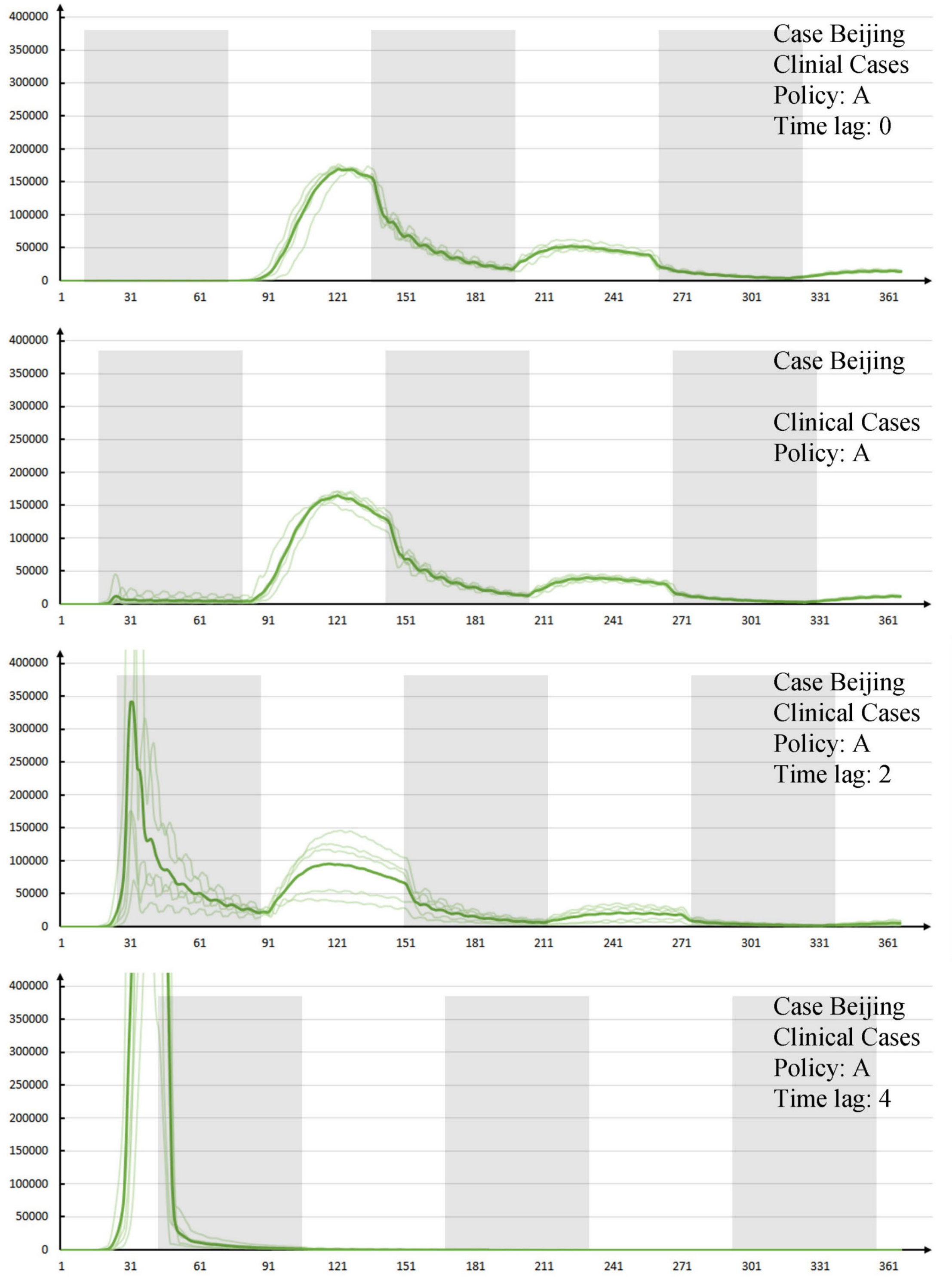
(Source: the author)



* Lockdown Periods in Policy 4 are marked in dark backgrounds.

Graph 21 Simulation Results of Policy Set A, Case Beijing (a)

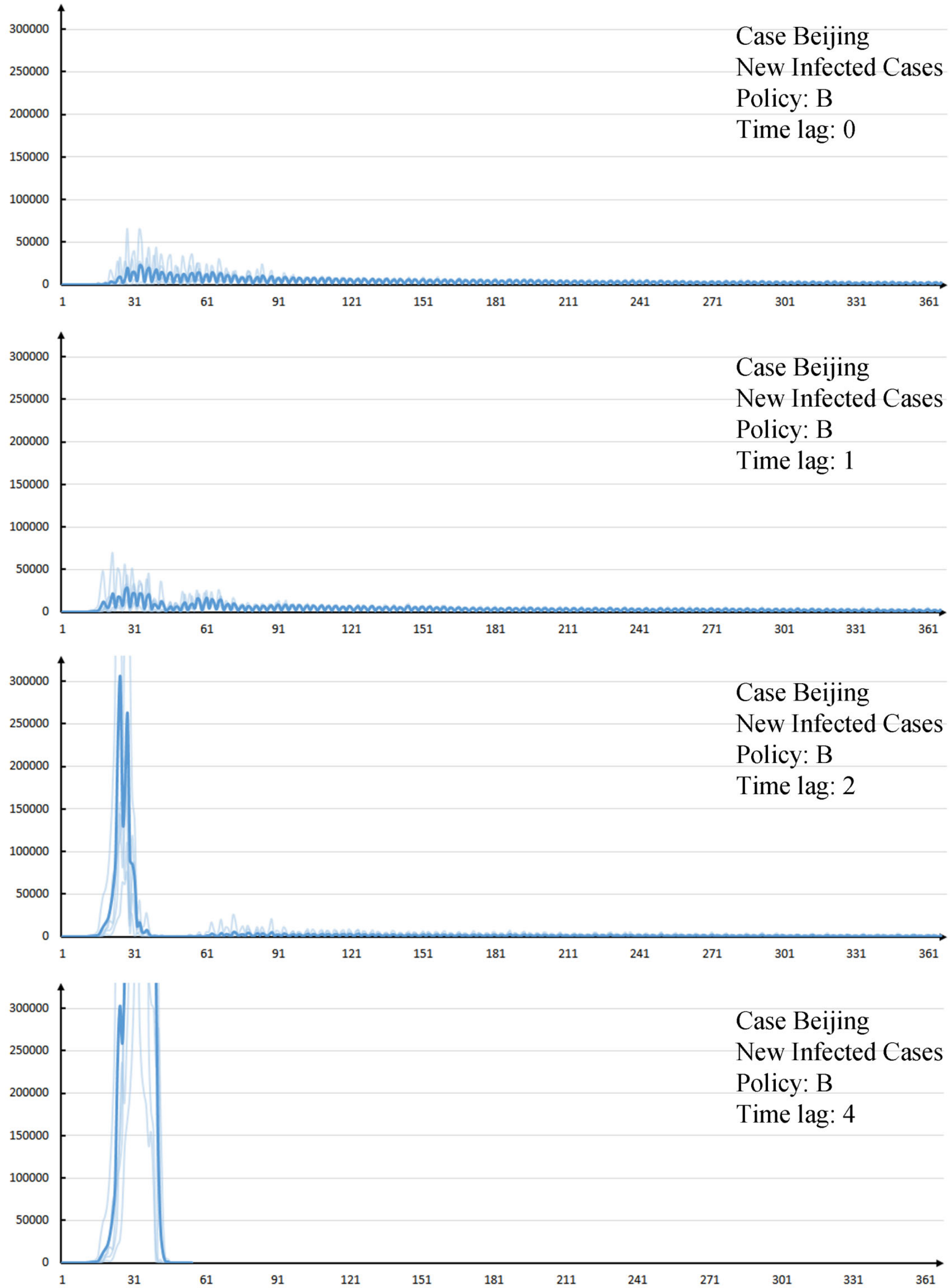
(Source: the author)



* Lockdown Periods in Policy 4 are marked in dark backgrounds.

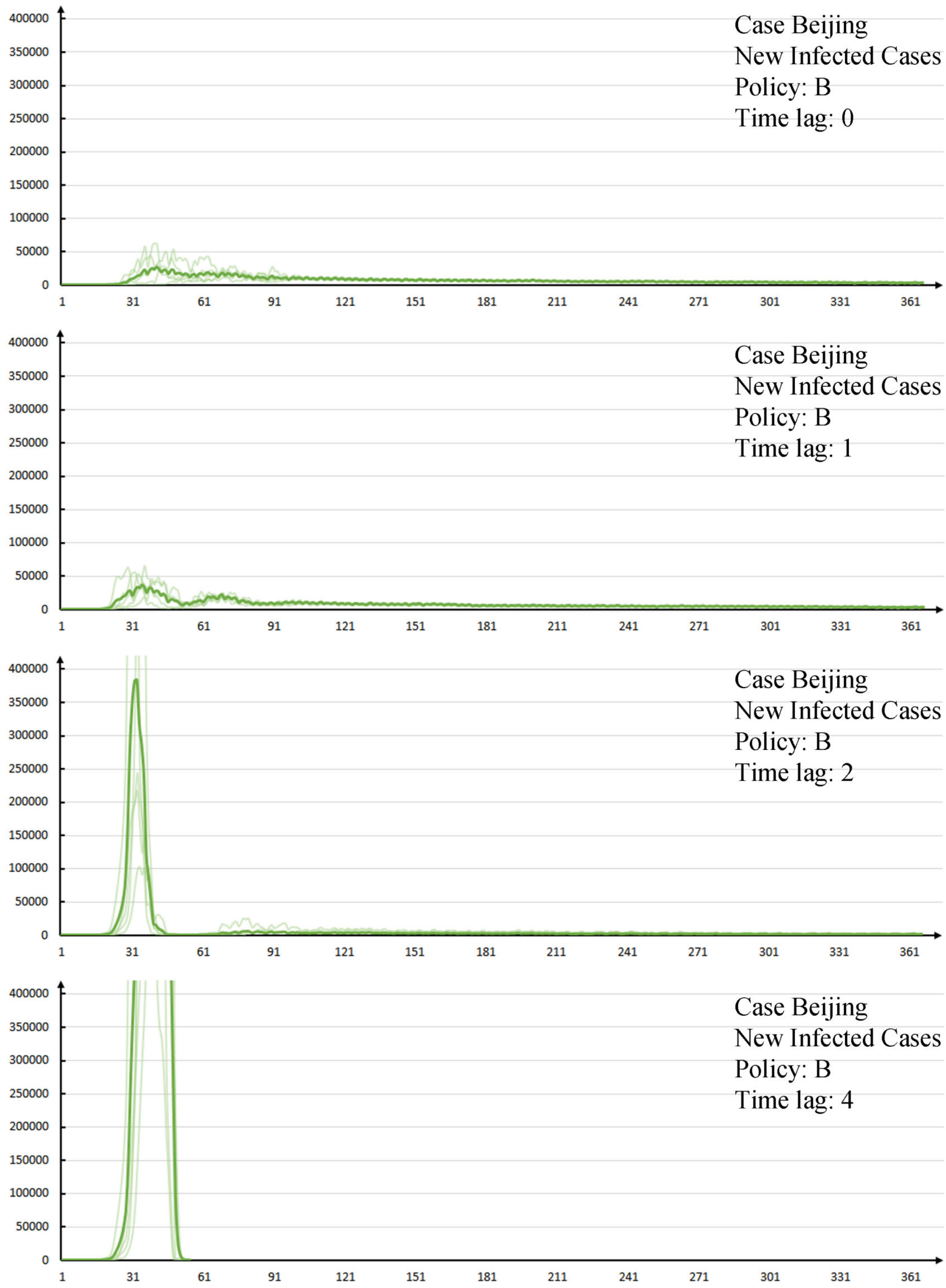
Graph 22 Simulation Results of Policy Set A, Case Beijing (b)

(Source: the author)



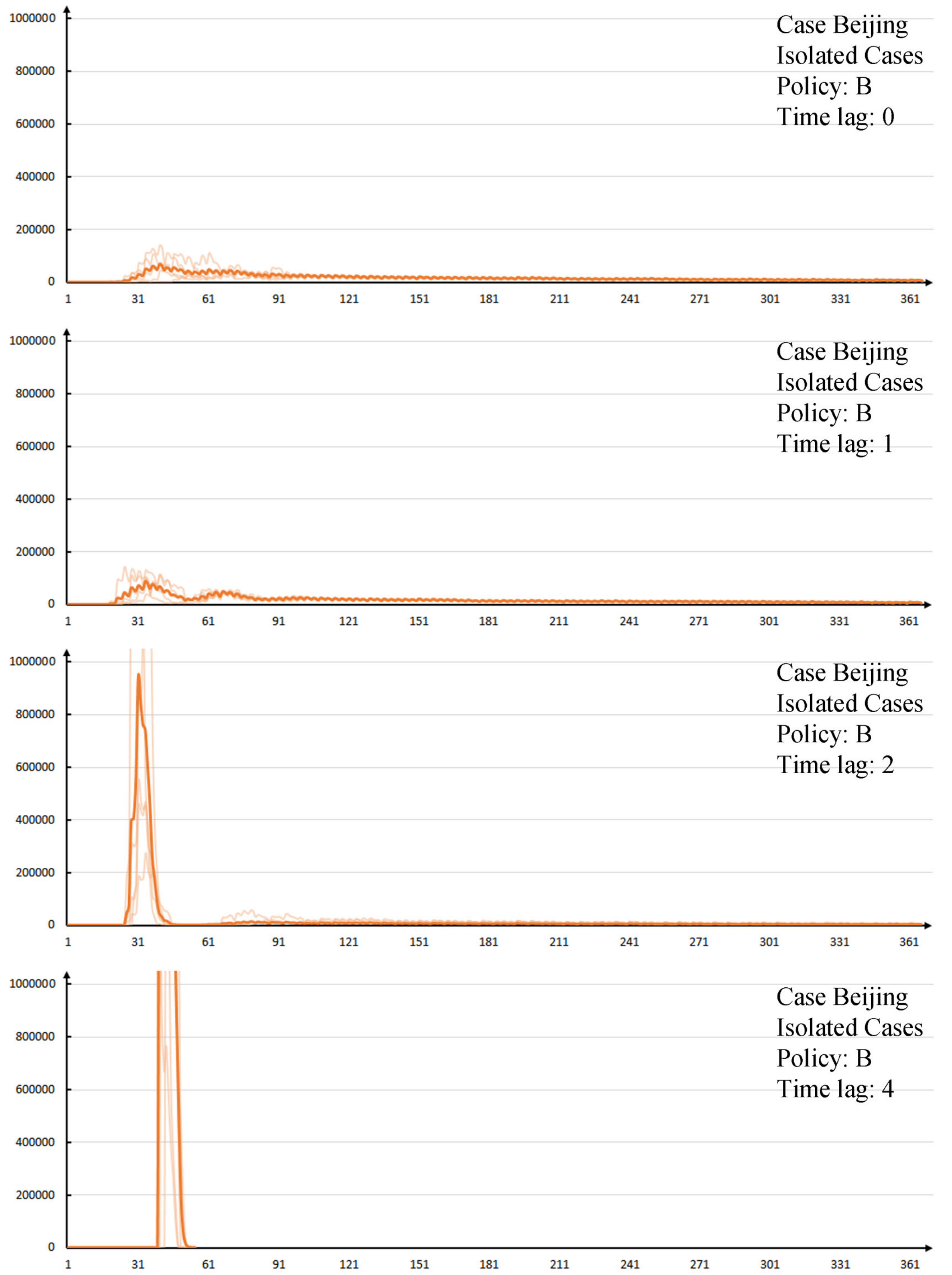
Graph 23 Simulation Results of Policy Set B, Case Beijing (a)

(Source: the author)



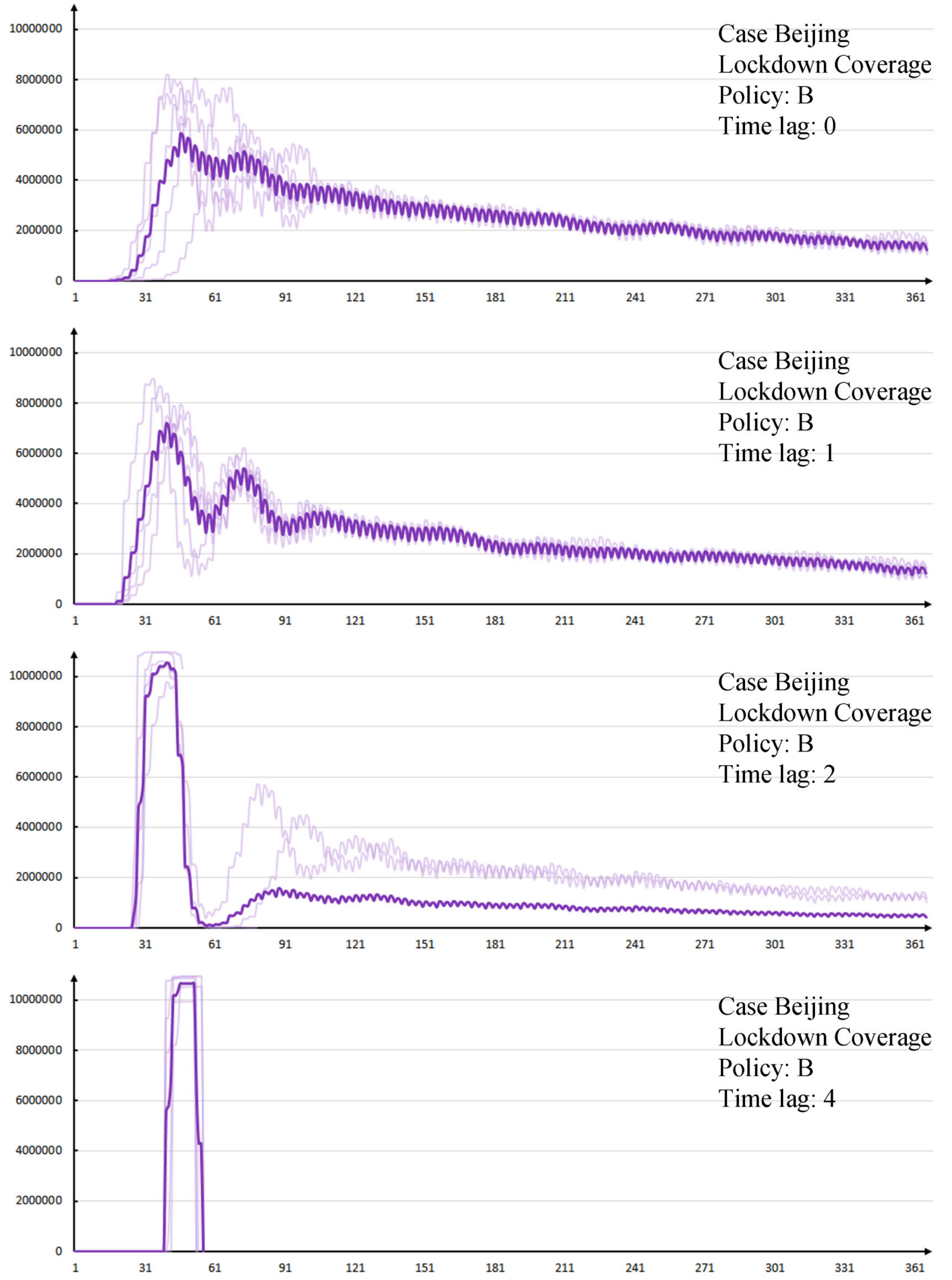
Graph 24 Simulation Results of Policy Set B, Case Beijing (b)

(Source: the author)



Graph 25 Simulation Results of Policy Set B, Case Beijing (c)

(Source: the author)



Graph 26 Simulation Results of Policy Set B, Case Beijing (d)

(Source: the author)

6 Discussion

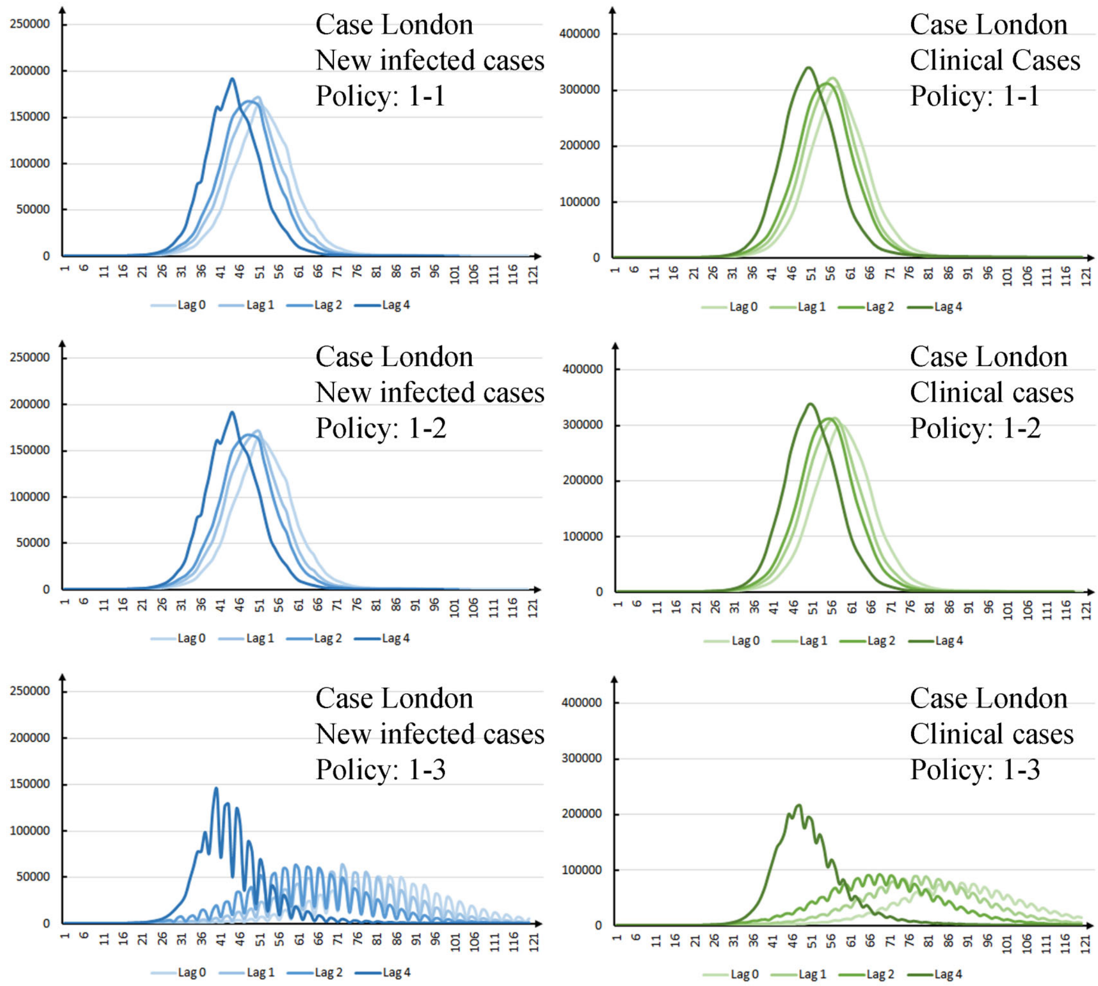
6.1 The effects of time lag to the performance of policies

Time lag plays an important role in controlling the spread of pandemic. Graphs 27 and 28 briefly show the average simulation results in the scenarios with single policy applied in different time lags. With the increase of time when the policy starts to be executed, the change of average infection curves is shown in gradually darker colours.

The effect of time lag can be discussed in several aspects. Firstly, the degrees of time lag effects are different among policies. For most policies, the average infection curves gradually move leftward when the lag of policy execution increases. Here, some special cases should be noticed. The performance of Policy 1-3, which is regular citywide nuclear acid test plus unified isolation, is very sensitive to the time lag. This also applies to Policy 4, or the regular overall lockdown. On the other hand, Policy 3, namely the rule of six, has little change among different time lags.

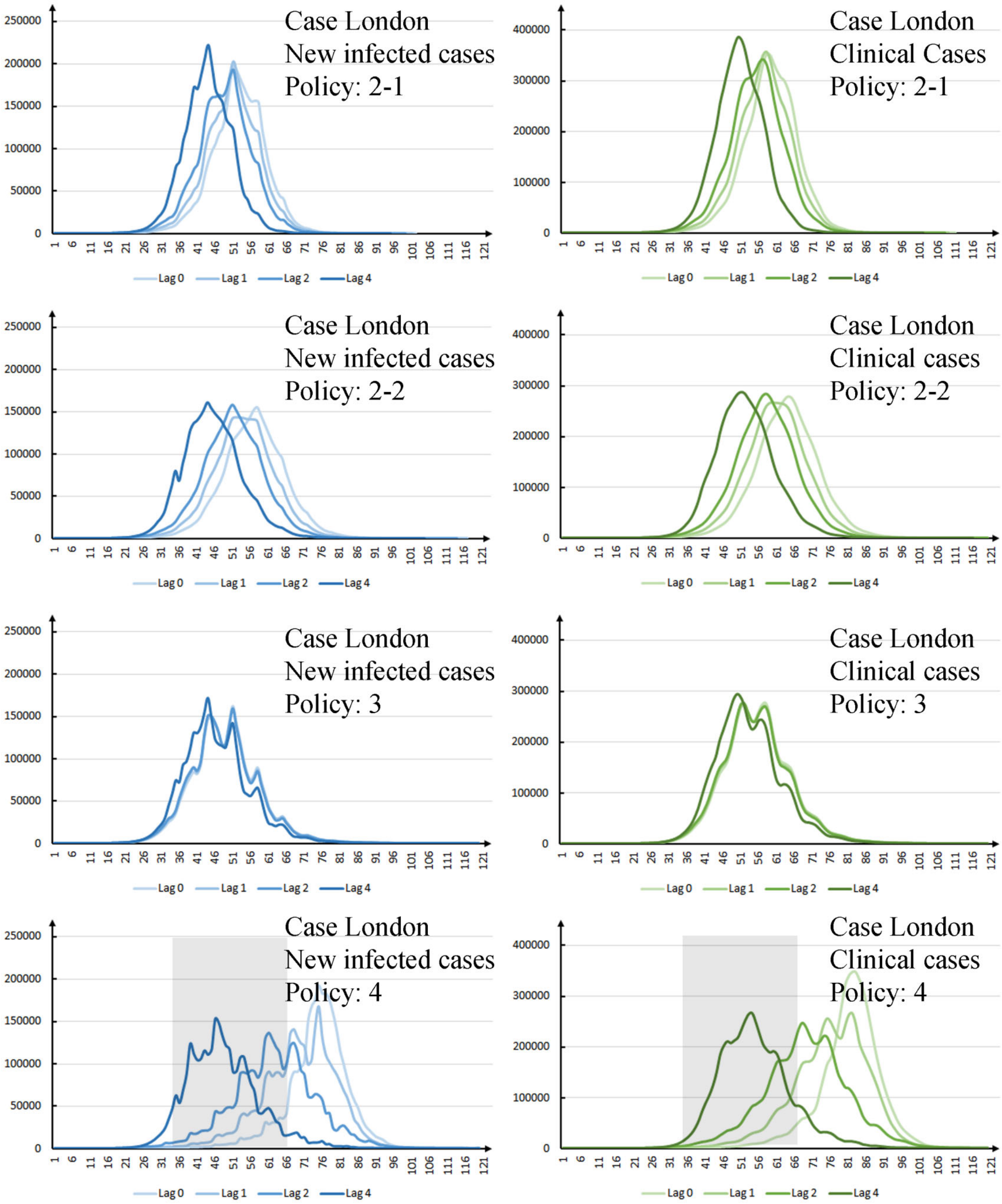
Then, the effect of time lag can be interpreted along two axes in the graphs. Comparing the average infection curves of the basic model and the model with policies, it can be concluded that the policies affect the infection curve in two aspects during the process of slowing down the transmission. On the one hand, the peak number of infections decreases with the execution of effective policies, resulting in the downward movement of the infection curve. On the other hand, the time of the infection peak is postponed by the policies, resulting in the rightward movement of the infection curve. Specifically, this also applies to Policy 4. Considering that Policy 4 is not executed throughout the simulation, the part of curves in the overlapping period of lockdown (shown with dark background) among various time lags shows the same trend as the scenarios of other policies.

In this way, the effect of policies against the pandemic on the infection curve can be divided into two dimensions, namely the number and time of the infection peak. In turn, the time lag in execution weakens the effect of policies and makes the infection curve gradually moves back to the shape of the basic model. With more lag in execution, the peak of the infection curve comes earlier, and the peak number becomes higher. It can be expected that if the time lag is infinite, the infection curve will be the same as that in the basic model, because no policy is actually executed.



Graph 27 Time Lag Effects of Single Policies (a), Case London

(Source: the author)



* Overlapping periods of lockdown in Policy 4 are marked in dark backgrounds.

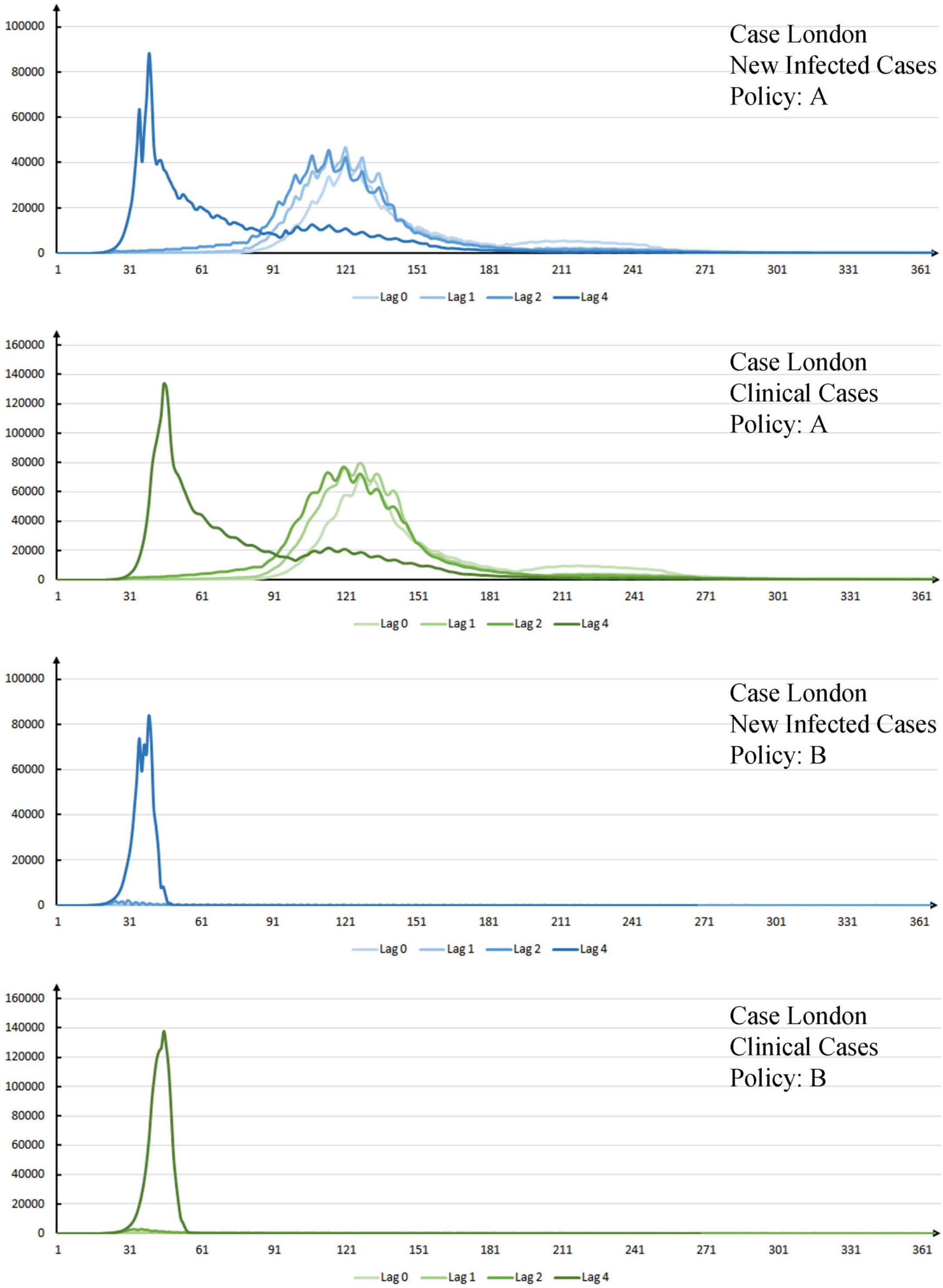
Graph 28 Time Lag Effects of Single Policies (b), Case London

(Source: the author)

6.2 Differences in policies

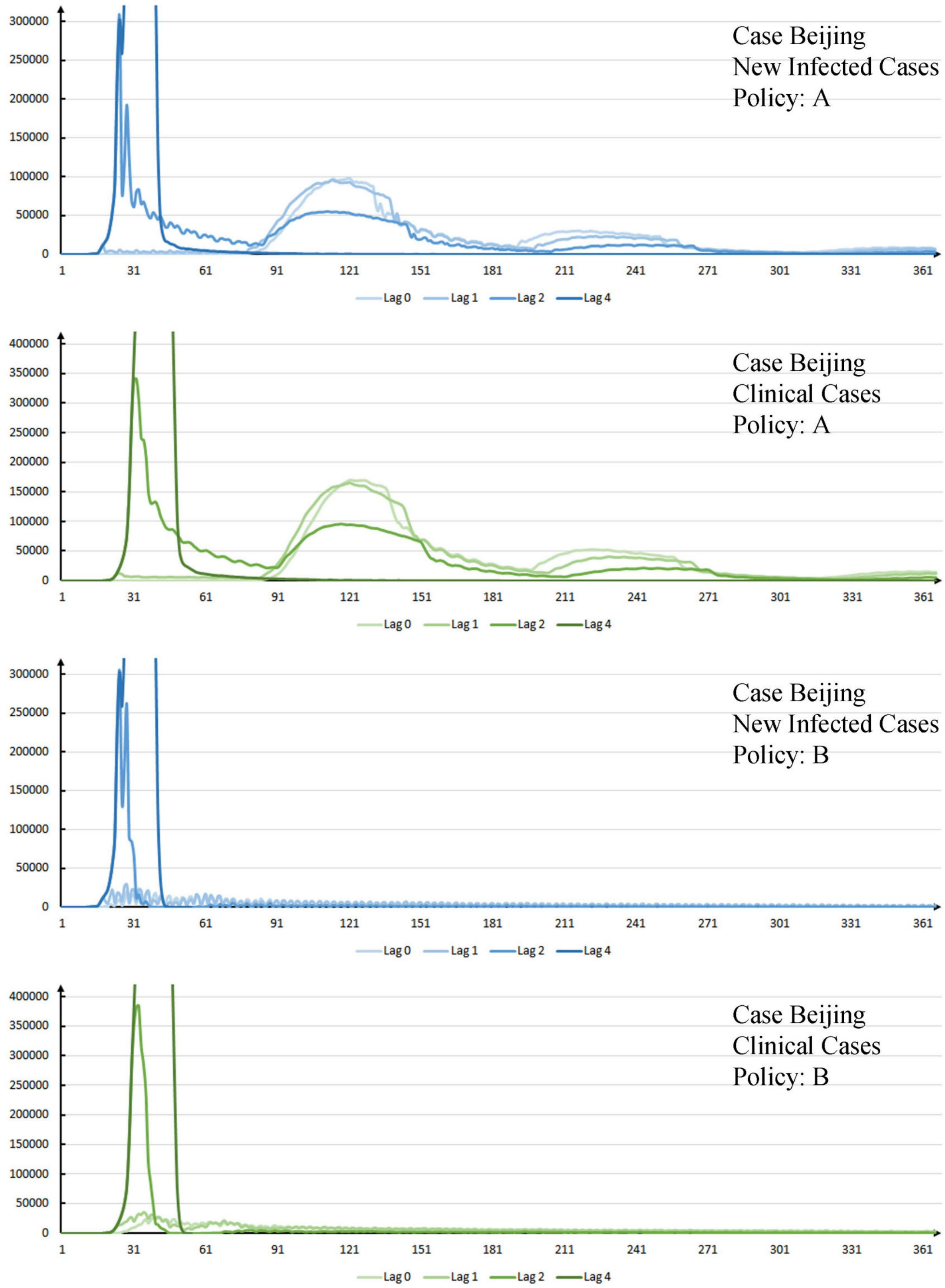
Graphs 29 and 30 briefly show the average results of Policy Set A and B with different time lags in Case London and Beijing. Overall, both policy sets are efficient in controlling the spread of pandemic within an *acceptable* time lag of execution, compared to the scenarios where only single policy is applied or the basic model without any public intervention. However, Policy Set B, characterised by stricter, spatial specific lockdown, tends to perform better than Policy Set A, characterised by more gentle, global control over the pandemic. In the results of Policy Set A in both cases, there is a significant rebound in the trend of infection between two periods of overall lockdowns. However, Policy Set B controls the speed of transmission to a stabler, lower level.

Detailed statistical indices of the simulation results are shown in Tables 14 and 15. Three indices of the controlling performance of policies, including the proportion of people free from infection after one year, the peak proportion of clinical cases in total population, and R value, prove that Policy Set B has a better controlling effect than Policy Set A. In the aspect of costs, the coverage indices of Policy Sets A and B cannot be directly compared. Although the coverage of Policy Set B is less, meaning that people are less affected by the policies, the restrictions themselves in Policy Set B are stricter than those in Policy Set A.



Graph 29 Brief Results of Multiple Policies, Case London

(Source: the author)



Graph 30 Brief Results of Multiple Policies, Case Beijing

(Source: the author)

Table 14 Statistical Indices of Policy Set A

(Source: the author)

Case	Time Lag (weeks)	Population Free from Infection	Peak of Clinical Cases	R Value	Self-Isolation Coverage	Overall Lockdown Coverage
London	0	41.06%	2.36%	1.466	0.28%	50%
London	1	34.19%	2.72%	1.56	0.32%	50%
London	2	30.33%	2.59%	1.622	0.33%	50%
London	4	32.05%	4.14%	1.594	0.33%	50%
Beijing	0	33.58%	1.59%	1.57	0.32%	50%
Beijing	1	33.32%	1.53%	1.572	0.32%	50%
Beijing	2	32.81%	4.61%	1.58	0.32%	50%
Beijing	4	2.44%	20.56%	2.478	0.47%	50%

Table 15 Statistical Indices of Policy Set B

(Source: the author)

Case	Time Lag (weeks)	Population Free from Infection	Peak of Clinical Cases	R Value	Unified Isolation Coverage	Regional Lockdown Coverage
London	0	99.73%	0.004%	1.056	0.004%	0.47%
London	1	99.17%	0.02%	1.148	0.01%	1.42%
London	2	98.30%	0.09%	1.09	0.02%	2.32%
London	4	77.15%	4.15%	1.53	0.66%	32.04%
Beijing	0	86.06%	0.48%	1.16	0.15%	22.82%
Beijing	1	85.88%	0.49%	1.16	0.15%	22.96%
Beijing	2	81.42%	4.76%	2.34	1.05%	28.32%
Beijing	4	5.64%	20.56%	3.564	2.63%	23.08%

6.3 Differences in cases

Besides, differences can also be found between Case London and Case Beijing. Given the same policy set and time lag, it is more difficult to control the pandemic in Case Beijing. On the one hand, both policy sets have poorer performance but higher side effects in Case Beijing compared to Case London. On the other hand, the performance of policies in Case Beijing is more sensitive to the change in time lag. In Case London, when the time lag increased to six weeks, the transmission gets completely out of control, and most people have been infected before the policies are taken into action. For Case Beijing, the time lag where most people get infected before policies is as short as four weeks.

Here two assumptions are raised to explain the difference. Firstly, the population density is higher in Case Beijing than Case London. In the simulation model, the average population per residential segment is 6.253 in Case London and 10.679 in Case Beijing. Secondly, in the simulation model, people living in Case Beijing need to walk longer to reach the nearest station for public transport, leading to more chances of contact and infection during daily commute. Figures 12 to 15 show the topological step depth and metric distance from each segment to the nearest segment with public transport stations. The average step depth in Case London is 2.50, and the average metric distance is 210m. The average step depth in Case London is 4.89, and the average metric distance is 616m.

Figure 12 Topological Step Depth from Nearest Public Transport, Case London

(Source: the author)

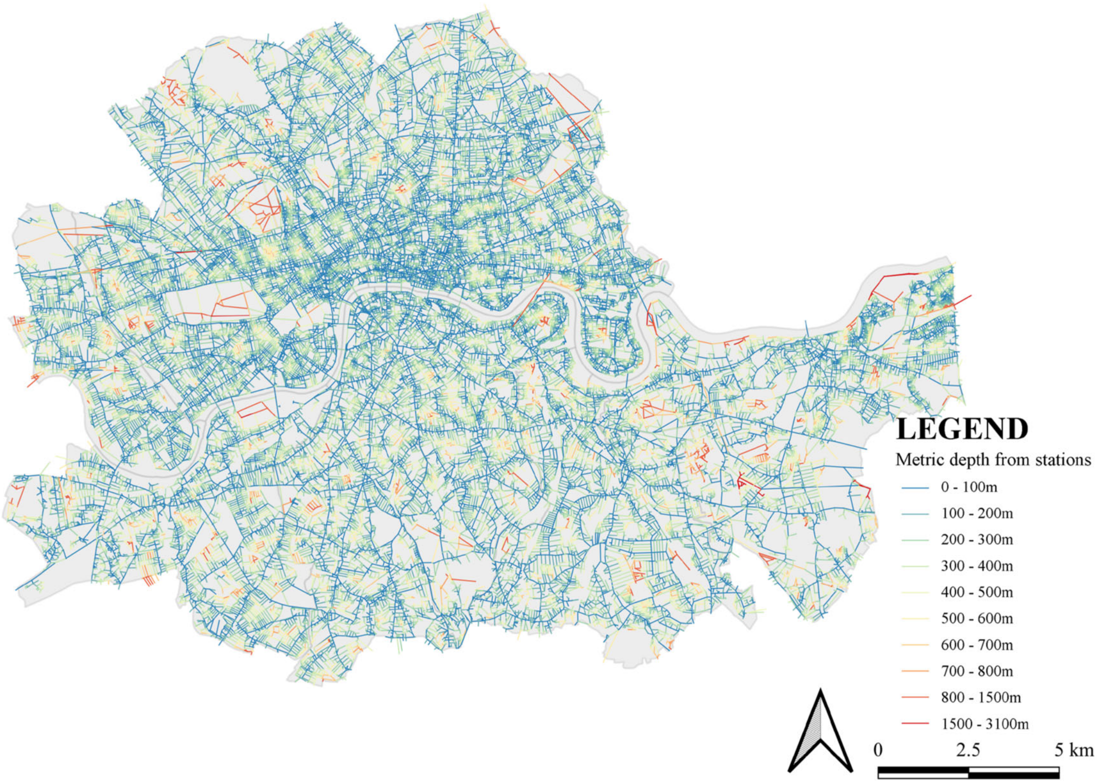


Figure 13 Metric Distance from Nearest Public Transport, Case London

(Source: the author)

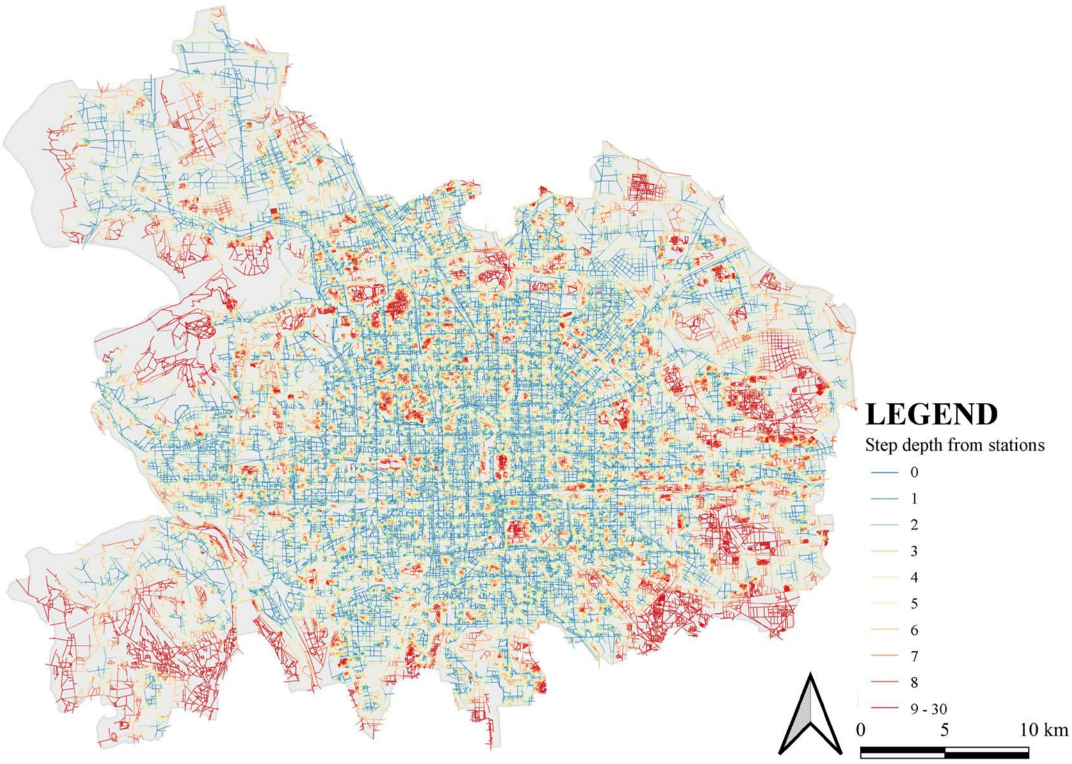


Figure 14 Topological Step Depth from Nearest Public Transport, Case London

(Source: the author)

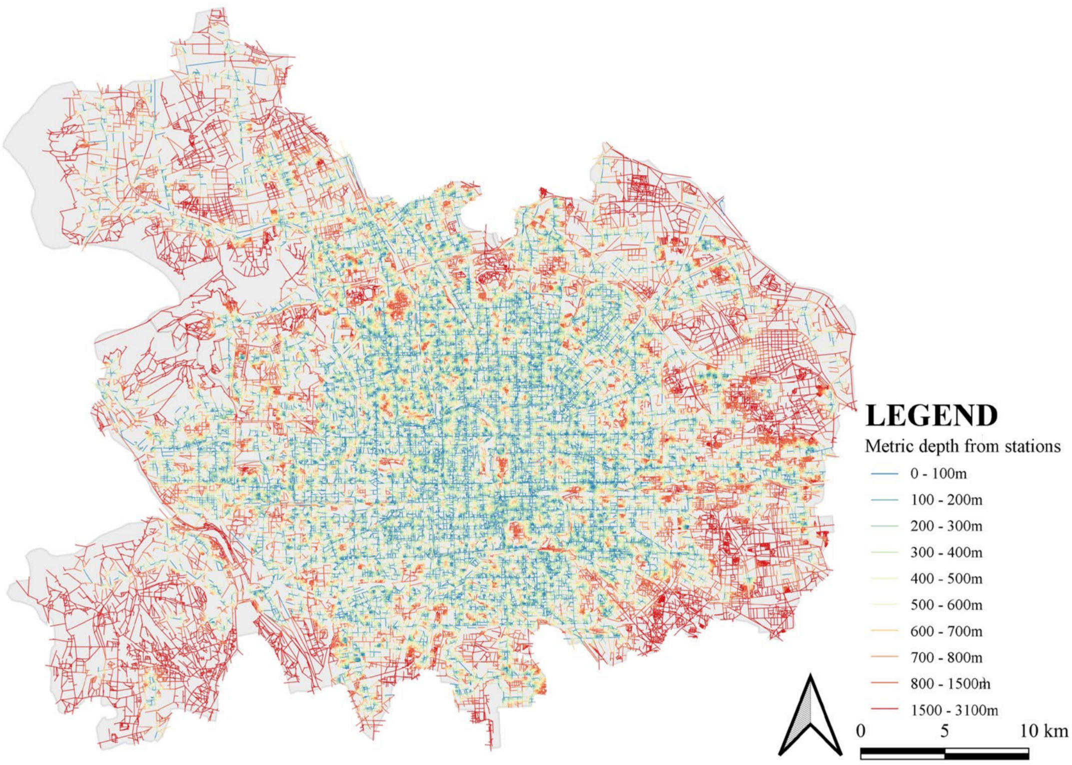


Figure 15 Metric Distance from Nearest Public Transport, Case London

(Source: the author)

7 Conclusion

7.1 Key findings

In this paper, the transmission process of pandemic was simulated on the segment maps of two case areas. Based on this, the effects of several policies against the pandemic were tested. Simulation results prove that the performance of policies on controlling the pandemic are affected by multiple factors including case areas, the combination of policies and time lag of executing policies. The main findings are listed below.

Time lag effects of policy execution

(1) The effect of time lag in execution varies among different policies.

(2) On the dimension of time, the peak of the infection curve comes earlier as the lag increases.

(3) On the dimension of number, the peak number of the infection becomes higher as the lag increases.

Performance of two policy sets

(1) Policy Set B, characterised by stricter, spatial specific lockdown, performs better in controlling the pandemic with acceptable side effects than Policy Set A, characterised by more gentle, global control over the pandemic.

(2) The performance of policies is relatively stable with different time lags of execution before the threshold where the transmission gets out of control before application of policies.

Difference in case areas

(1) The transmission of pandemic is faster in Case Beijing than Case London given the same infection rate in simulation, leading to worse performance of policies and earlier threshold of the time lag.

(2) It is assumed that the difference between Case London and Case Beijing originates from the differences in population density and average distance to nearest public transport stations.

7.2 Implication

The main significance of this paper is that a spatial-based method of simulating the pandemic is introduced. Here, space is no longer a statistical concept containing contacts among people but returns to its nature of supporting movements and usage. Contact also returns to its nature, namely the product of movements and encounters. Based on this model, a series of indices for quantifying the performance of policies are also introduced, making it possible to test the effect of spatial-related policies.

The comparison between two sets of policies shows that in ideal scenarios, it is feasible to apply strict intervention within a small range where infections happen. If the pandemic could be stopped within a short period of time, the side effects caused by stricter policies may even be less than the case where gentler but less effective policies keep being applied for a long time. This finding actually constitutes theoretical support for the Chinese policy against the pandemic in the early period, which is often criticised by its strictness. However, considering other social and economic factors, it is not feasible for every country to act the same. It should also be noticed that the city is not an enclosed system in the real world. Faced with global pandemic, the duration of fight against the pandemic is often longer than expected, and the side effect of stricter policies may rapidly increase over time. Therefore, non-pharmaceutical interventions should be treated as temporary measures to win more time for medical progress, rather than the ultimate solution to be relied on in the long run. Besides, the results also emphasise the importance of time. The quicker the government takes action, the less cost is paid for the pandemic.

7.3 Research deficiencies and prospects

Technically, several factors are simplified due to lack of data or the restriction of hardware. For example, a newly published study pointed out that the risk of infection is related to the duration when one is exposed to the virus (Oza, 2023). Therefore, more factors can be included in the system of determining effective contacts and calculate probabilities of infection. Moreover, the movements of individuals are also simplified into fixed random patterns, but there are more varieties in movement patterns and spatial usage. It is also possible that behaviours may be affected by the pandemic.

As for the framework, although this study proposed a spatial-based simulation model of the pandemic, the link between spatial usage and the transmission is yet to be implemented. Whether or not the space still matters on the background of pandemic is worthy of further exploration. Besides, this study focuses on the transmission of pandemic, but public policies are determined by multiple factors, for example political issues, public budget, or other socio-economic pressures. With more cities included and more factors taken into consideration in the future, this model will be more helpful for providing reference for public medical emergencies.

References

ABDUL NASIR, N. A. B., HASSAN, A. S., KHOZAEI, F. & ABDUL NASIR, M. H. B. 2021. Investigation of spatial configuration management on social distancing of recreational clubhouse for COVID-19 in Penang, Malaysia. *International Journal of Building Pathology and Adaptation,* 39**,** 782-810.

ASKARIZAD, R., MEHRINEJAD, M. & SOMASUNDARASWARAN, K. 2022. The impact of COVID-19 on visitors' wayfinding within healthcare centers. *Ain Shams Engineering Journal*.

BARBER, S., BROWN, J. & FERGUSON, D. 2022. Coronavirus: lockdown laws. House of Commons.

BROWN, J. & KIRK-WADE, E. 2-21. Coronavirus: A history of 'Lockdown laws' in England. House of Commons.

BüYüKŞAHIN, S. 2023. Effects of COVID-19 pandemic on spatial preferences and usage habits of users in shopping malls and its relation with circulation layout. *Ain Shams Engineering Journal,* 14**,** 101838.

CHEN, P., ZHANG, D., LIU, J. & JIAN, I. Y. 2022. Assessing personal exposure to COVID-19 transmission in public indoor spaces based on fine-grained trajectory data: A simulation study. *Building and Environment,* 218**,** 109153.

CHINA, T. S. C. I. O. O. T. P. S. R. O. 2020. Fighting Covid-19 China in Action.

COLBOURN, T. 2020. Unlocking UK COVID-19 policy. *The Lancet Public Health,* 5**,** e362-e363.

DAVIES, N. G., KUCHARSKI, A. J., EGGO, R. M., GIMMA, A., EDMUNDS, W. J., JOMBART, T., O'REILLY, K., ENDO, A., HELLEWELL, J., NIGHTINGALE, E. S., QUILTY, B. J., JARVIS, C. I., RUSSELL, T. W., KLEPAC, P., BOSSE, N. I., FUNK, S., ABBOTT, S., MEDLEY, G. F., GIBBS, H., PEARSON, C. A. B., FLASCHE, S., JIT, M., CLIFFORD, S., PREM, K., DIAMOND, C., EMERY, J., DEOL, A. K., PROCTER, S. R., VAN ZANDVOORT, K., SUN, Y. F., MUNDAY, J. D., ROSELLO, A., AUZENBERGS, M., KNIGHT, G., HOUBEN, R. M. G. J. & LIU, Y. 2020. Effects of non-pharmaceutical interventions on COVID-19 cases, deaths, and demand for hospital services in the UK: a modelling study. *The Lancet Public Health,* 5**,** e375-e385.

FERGUSON, N., LAYDON, D., NEDJATI GILANI, G., IMAI, N., AINSLIE, K., BAGUELIN, M., BHATIA, S., BOONYASIRI, A., CUCUNUBA PEREZ, Z. & CUOMO-DANNENBURG, G. 2020. Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand.

HILLIER, B. & HANSON, J. 1984. *The social logic of space*, Cambridge university press.

HUNT, J., BRISTOW, P., COOPER, R., DAVIES, J., EVANS, L., KEELEY, B. O., TAIWO OWEN, SARAHANUM, QAISAR-JAVED, A., RUSSELL, D. & TROTT, L. 2021. Coronavirus: lessons learned to date. House of Commons.

ISTIANI, N. F. F., ALKADRI, M. F., VAN NES, A. & SUSANTO, D. 2023. Investigating the spatial network of playgrounds during covid-19 based on a space syntax analysis case study: 10 playgrounds in Delft, the Netherlands. *Cogent Social Sciences,* 9**,** 2163754.

LEGEBY, A., KOCH, D., DUARTE, F., HEINE, C., BENSON, T., FUGIGLANDO, U. & RATTI, C. 2022. New urban habits in Stockholm following COVID-19. *Urban Studies,* 0**,** 00420980211070677.

LI, J., SHU, Y., CHEN, N., WANG, F. & LI, H. 2022. 'Re-socialisation'in isolated spaces: A case study on the social organisation of Fangcang shelter hospital patients under extreme spatial conditions. *Indoor and Built Environment,* 31**,** 1210-1223.

LIMA, F. T., BROWN, N. C. & DUARTE, J. P. 2021. Understanding the Impact of Walkability, Population Density, and Population Size on COVID-19 Spread: A Pilot Study of the Early Contagion in the United States. *Entropy,* 23**,** 1512.

MUSTAFA, F. A. & AHMED, S. S. 2022. The role of waiting area typology in limiting the spread of COVID-19: Outpatient clinics of Erbil hospitals as a case study. *Indoor and Built Environment***,** 1420326X221079616.

OZA, A. 2023. COVID infection risk rises the longer you are exposed-even for vaccinated people. *Nature*.

PARK, J. Y., MISTUR, E., KIM, D., MO, Y. & HOEFER, R. 2022. Toward human-centric urban infrastructure: Text mining for social media data to identify the public perception of COVID-19 policy in transportation hubs. *Sustainable Cities and Society,* 76**,** 103524.

SCHNEIDER, K. A., NGWA, G. A., SCHWEHM, M., EICHNER, L. & EICHNER, M. 2020. The COVID-19 pandemic preparedness simulation tool: CovidSIM. *BMC infectious diseases,* 20**,** 1-11.

TFL 2020. Travel in London Report 13.

VAN BASSHUYSEN, P., WHITE, L., KHOSROWI, D. & FRISCH, M. 2021. Three ways in which pandemic models may perform a pandemic. *Erasmus Journal for Philosophy and Economics,* 14**,** 110–127-110–127.

WALKER, P., WHITTAKER, C., WATSON, O., BAGUELIN, M., AINSLIE, K., BHATIA, S., BHATT, S., BOONYASIRI, A., BOYD, O. & CATTARINO, L. 2020. Report 12: The global impact of COVID-19 and strategies for mitigation and suppression.

YUAN, K., ABE, H., OTSUKA, N., YASUFUKU, K. & TAKAHASHI, A. 2022. Impact of the COVID-19 Pandemic on Walkability in the Main Urban Area of Xi’an. *Urban Science,* 6**,** 44.

Appendix I Chinese Terminology

The terms marked with * are not the official English names of corresponding Chinese terms. They are translated by the author and are only used in this paper.

Term	Chinese	Definition or Description
Close contact*	密切接触者	People who have had direct contact or copresence with infected cases. They are facing the risk of being infected with COVID-19.
*Covid-19 Prevention and Control Guideline**	《新型冠状病毒肺炎防控方案》	An official document published by National Health Commission of the People's Republic of China, acting as the main guideline of policies against COVID-19 in China.
District	区，县	A level of administrative division below the city.
(High/Medium/Low) Risky Areas*	（高/中/低）风险区	Official updated data indicating the areas facing the threat of COVID-19. They are usually divided into 3 levels.
National Health Commission of the People's Republic of China	中华人民共和国国家卫生健康委员会	The national department responsible for medical issues in China.
National Metadata Pass*	通信大数据行程卡	An official metadata platform for analysing the risk of being infected of an individual through movement traces. This is used for entry permission to public indoor spaces.
Official Press Conferences on COVID-19 in Beijing	北京市新冠肺炎疫情防控工作新闻发布会	A set of press conferences held by the government of Beijing for updating the dynamics of infected cases and relevant policies.
*Risk Grading Standards for Covid-19 in Beijing**	《北京市新冠肺炎疫情风险分级标准》	An official document published by the government of Beijing, including the detailed rules of division of Risky Areas in Beijing.
Risky Status*	风险状态	A term by the author for the colour of National Metadata Pass of an individual. Different colours indicate different levels of risk of being infected.
Sub-District	街道，乡，镇，地区	A level of administrative division below the district.
WeChat	微信	A social application popular in China.

Appendix II Code Realisation of Simulation

The simulation model is realised with Python 3.11.

Simulation

Main9.py

```
"""
Version Change:
1. Basic model for small area.
2. Numpy and iGraph introduced.
3. Trial on pre-computation.
4. Means of transport introduced.
5. Change from class to function and vectorisation applied.
6. Change on basic logic for full speed.
7. Multiprocessing method introduced.
8. Policies introduced.
9. Code refactoring.
"""

import time
import numpy as np
import json
from multiprocessing import Pool
from multiprocessing.shared_memory import SharedMemory
from multiprocessing.managers import SharedMemoryManager
from Const import *
from Simulation import simulation


def base(_ss=0, _city=0, _p=None, loc=''):    # fixed seed

    # speed test
    _start = time.time()

    # columns: 0.origin, 1.geoid, 2.population per segment (divided by 10)
    segment = np.loadtxt(PRE_FILE[_city]['segment'], delimiter=',',
                         skiprows=1, dtype='int32', encoding='utf-8', quotechar='"')
    _segment_len = segment.shape[0]
    _population = int(segment[:, 2].sum())

    # list of segments as origins
    origins = segment[segment[:, 0] == 0, 1]

    # Shared memory 1 of 2
    with SharedMemoryManager() as smm:

        # input parameters including seed, city and policies
        _sm0 = SharedMemory('parameters', True, 28)
        _seed = np.ndarray((7,), 'int32', _sm0.buf)
        _seed[:] = [_ss + 1000, _city] + list(_p)

        """
        Each parameter in policy0:
        0: policy trigger.
        1: isolation policy.
        2: face masks.
        3: gathering policy.
```

```
        4: lockdown policy.
        """
        # _sm8 = SharedMemory('policy0', True, 20)    # policy and trigger
        # _policy0 = np.ndarray((5,), 'int32', _sm8.buf)
        # _policy0[:] = _p

        _temp = int(np.sum(segment[:, 0] == 1))
        _sm1 = SharedMemory('destinations', True, _temp * 4)
        _destinations = np.ndarray((_temp,), 'int32', _sm1.buf)
        _destinations[:] = segment[segment[:, 0] == 1, 1]

        # population of each segment
        _sm2 = SharedMemory('region', True, _segment_len * 4)
        _region = np.ndarray((_segment_len,), 'int32', _sm2.buf)
        _region[:] = segment[:, 2]

        _sm3_1 = SharedMemory('station path index', True, (_segment_len + 1) * 4)
        _s_index = np.ndarray((_segment_len + 1,), 'int32', _sm3_1.buf)

        _sm4_1 = SharedMemory('neighbours index', True, (_segment_len + 1) * 4)
        _n_index = np.ndarray((_segment_len + 1,), 'int32', _sm4_1.buf)

        _sm5 = SharedMemory('sample', True, _population * 20)
        # attributes fixed for each individual in sample
        # columns: 0.Age, 1.Home, 2.Work/School, 3.Car, 4.Household
        _sample = np.ndarray((_population, 5), 'int32', _sm5.buf)

        _sm6 = SharedMemory('status', True, _population * 12)
        # changing status of individuals in sample
        # 0.Status, 1.Count, 2.Location
        _status = np.ndarray((_population, 3), 'int32', _sm6.buf)

    # read all paths from segment to the nearest station
    origin_s_path = []
    _count = 0
    with open(PRE_FILE[_city]['station path'], 'r') as f:
        for a, line in enumerate(f):
            _s_index[a] = _count
            _temp = json.loads(line)
            origin_s_path += _temp
            _count += len(_temp)
    _s_index[-1] = len(origin_s_path) + 1

    # read list of neighbours for all segments
    origin_n_path = []
    _count = 0
    with open(PRE_FILE[_city]['neighbours'], 'r') as f:
        for a, line in enumerate(f):
            _n_index[a] = _count
            _temp = json.loads(line)
            origin_n_path += _temp
            _count += len(_temp)
    _n_index[-1] = len(origin_n_path) + 1

    # Shared memory 2 of 2
    with SharedMemoryManager() as smm:

        _sm3 = SharedMemory('station path', True, len(origin_s_path) * 4)
        _station_path = np.ndarray((len(origin_s_path),), 'int32', _sm3.buf)
        _station_path[:] = origin_s_path

        _sm4 = SharedMemory('neighbours', True, len(origin_n_path) * 4)
        _neighbours = np.ndarray((len(origin_n_path),), 'int32', _sm4.buf)
        _neighbours[:] = origin_n_path

        _sm7 = SharedMemory('household', True, 4)    # number of household
        _household = np.ndarray((1,), 'int32', _sm7.buf)

    # sample initialisation
    """
    Here we assume that there are several stages for each individual.
```

```
    0.Susceptible, where one is healthy and moves daily.
    1.Exposed, where one is effectively infected while is not infectious. An exposed person is 50% likely to enter
    Preclinical and then clinical stage, while 50% to Subclinical stage with 50% ability of infection.
    2.Preclinical, where one is infected and infectious.
    3.Clinical
    4.Subclinical
    5.Removed, where one is recovered or dead and is removed from the model.
    """
    # random number generator (seed fixed for basic sample)
    _rng = np.random.default_rng(seed=_ss)

    _status[:, 1] = -1
    _sample[:, 0] = _rng.choice(3, _population, p=AGE[_city])    # age assigned
    _count = 0    # used in people creation
    for a in range(len(origins)):
        _temp = 0    # record the number of people in the current household
        for j in range(segment[origins[a], 2]):
            _sample[_count, 1] = origins[a]
            # household assigned
            if _rng.random() > HOUSEHOLD_SIZE[_city][_temp]:
                _temp += 1
            else:
                _household[0] += 1
                _temp = 1
            _sample[_count, 4] = _household[0]
            _count += 1
        _household[0] += 1
    _status[:, 2] = _sample[:, 1]
    _sample[:, 2] = _sample[:, 1]

    # car assigned
    _sample[_rng.random(_population) < 0.5, 3] = 1
    # work place or school assigned
    _temp = _sample[:, 0] < 2
    _sample[_temp, 2] = _rng.choice(_destinations, np.argwhere(_temp).shape[0])

    # simulation(0)

    core = 10
    with Pool(10) as pool:
        pool.map(simulation, range(core))

    print('Simulation model ended in {} seconds. seed = {}'.format(time.time() - _start, _ss))

    # summarising
    _summary = np.loadtxt(r'Log\temp\summary0.txt', delimiter=',', skiprows=1, dtype='int32', encoding='utf-8')
    for a in range(1, core):
        _temp = np.loadtxt(r'Log\temp\summary{}.txt'.format(a), delimiter=',', skiprows=1, dtype='int32',
                           encoding='utf-8')
        if _temp.shape[0] > _summary.shape[0]:
            _summary = np.append(_summary, np.tile(_summary[-1], (_temp.shape[0] - _summary.shape[0], 1)),
0)
        else:
            _temp = np.append(_temp, np.tile(_temp[-1], (_summary.shape[0] - _temp.shape[0], 1)), 0)
        _summary += _temp

    # distribution
    _distribution = np.loadtxt(r'Log\temp\distribution0.txt', delimiter=',', skiprows=1, dtype='float64',
                               encoding='utf-8')
    if len(_distribution.shape) == 1:
        _distribution = _distribution.reshape((_distribution.shape[0], 1))
    for a in range(1, core):
        _temp = np.loadtxt(r'Log\temp\distribution{}.txt'.format(a), delimiter=',', skiprows=1, dtype='int32',
                           encoding='utf-8')
        if len(_temp.shape) == 1:
            _temp = _temp.reshape((_temp.shape[0], 1))
        if _temp.shape[1] > _distribution.shape[1]:
            _distribution = np.append(_distribution,
                                      np.tile(_distribution[:, -1], (_temp.shape[1] - _distribution.shape[1],
1)).T, 1)
        else:
```

```
            _temp = np.append(_temp, np.tile(_temp[:, -1], (_distribution.shape[1] - _temp.shape[1], 1)).T, 1)
        _distribution += _temp
    for a in range(_segment_len):
        if segment[a, 2] != 0:
            _distribution[a] /= 10 * segment[a, 2]

    # abstract daily summary by rows (t = 0, 4, 8, ...)
    _temp = np.empty((_summary.shape[0] // 4 + 1, 4), 'int32')
    for a in range(_temp.shape[0] - 1):
        _temp[a] = _summary[a * 4, [0, 3, 6, 7]]
    _temp[-1] = _summary[-1, [0, 3, 6, 7]]
    # abstract daily summary by columns (newly infected cases, current clinical cases)
    _brief = _temp.copy()
    _brief[0] = [0, 0, 0, 0]
    for a in range(1, _brief.shape[0]):
        _brief[a, 0] = _temp[a - 1, 0] - _temp[a, 0]

    _temp = time.strftime('%Y%m%d-%H%M%S')
    with open(r'Log\log.txt', 'a+') as f:
        f.write('Write log of seed {} at {}.\n'.format(_ss, _temp))
    np.savetxt(r'Log\{}summary\summary-seed{}.txt'.format(loc, _ss), _summary, fmt='%d', delimiter=',',
               header='Susceptible, Exposed, Preclinical, Clinical, Subclinical, Removed')
    np.savetxt(r'Log\{}distribution\distribution-seed{}.txt'.format(loc, _ss), _distribution, fmt='%.3f',
               delimiter=',', header='Week')
    np.savetxt(r'Log\{}brief\brief-seed{}.txt'.format(loc, _ss), _brief, fmt='%d', delimiter=',',
               header='New cases, Current clinical cases')


if __name__ == '__main__':
    base(0, City.BEIJING, (0, 0, 0, 0, 0), r"Backup2\BR-lag4\\")
```

Simulation.py

```
import time
import numpy as np
from multiprocessing.shared_memory import SharedMemory
from Const import *


def simulation(number=0):    # sub-model id

    start = time.time()
    print('Sub-Model #{} started.'.format(number))

    # pass data
    sm0 = SharedMemory('parameters', False)
    ss = np.ndarray((7,), 'int32', sm0.buf)
    city = ss[1]
    policy0 = ss[2:]

    sm1 = SharedMemory('destinations', False)
    destinations = np.ndarray(MEMORY_SIZE[city]['destinations'], 'int32', sm1.buf)

    sm2 = SharedMemory('region', False)
    region = np.ndarray(MEMORY_SIZE[city]['region'], 'int32', sm2.buf)

    sm3 = SharedMemory('station path', False)
    station_path = np.ndarray(MEMORY_SIZE[city]['station_path'], 'int32', sm3.buf)

    sm3_1 = SharedMemory('station path index', False)
    s_index = np.ndarray(MEMORY_SIZE[city]['station_path_index'], 'int32', sm3_1.buf)

    sm4 = SharedMemory('neighbours', False)
    neighbours = np.ndarray(MEMORY_SIZE[city]['neighbours'], 'int32', sm4.buf)

    sm4_1 = SharedMemory('neighbours index', False)
    n_index = np.ndarray(MEMORY_SIZE[city]['neighbours_index'], 'int32', sm4_1.buf)

    sm5 = SharedMemory('sample', False)
```

```python
base_sample = np.ndarray(MEMORY_SIZE[city]['sample'], 'int32', sm5.buf)

sm6 = SharedMemory('status', False)
base_status = np.ndarray(MEMORY_SIZE[city]['status'], 'int32', sm6.buf)

sm7 = SharedMemory('household', False)
household = np.ndarray((1,), 'int32', sm7.buf)[0]

# random number generator (seed fixed for infected cases)
rng1 = np.random.default_rng(seed=ss[0])
# internal random number generator (seed not fixed for randomness)
rng = np.random.default_rng(seed=0)

policy = np.array([0, 0, 0, 0], 'int32')
# trigger for announce policies (-2: not triggered yet; -1: having triggered)
if policy0[0] == -1:
    trigger = 0   # trigger policy immediately
else:
    trigger = -2
# infection rate of wearing masks
mask_out, mask_in = 0.98, 0.98
sample = base_sample.view()
status = base_status.copy()
population = sample.shape[0]

# fixed route in weekdays
trace = np.empty(population, dtype=object)
for i in range(population):
    if sample[i, 0] < 2:
        if sample[i, 3]:
            trace[i] = sample[i, [1, 2]]
        else:
            temp1 = sample[i, 1]
            temp2 = sample[i, 2]
            trace[i] = np.append(station_path[s_index[temp1]:s_index[temp1 + 1]],
                                 station_path[s_index[temp2]:s_index[temp2 + 1]])
    else:
        temp1 = sample[i, 1]
        trace[i] = neighbours[n_index[temp1]:n_index[temp1 + 1]]

# set the initial case as exposed
temp = rng1.choice(population)
infective = np.ones(population, dtype='int32')   # distinguish how infectious one is
status[temp, [0, 1]] = [1, 16]
# parameters in order: Susceptible, Exposed, Preclinical, Clinical, Subclinical, Removed
summary = np.array([[population - 1, 1, 0, 0, 0, 0, 0, 0]], dtype='int32')
# record spatial distribution of infected in wd_spread0
distribution = np.empty((SEGMENT_LEN[city], 0), 'int32')

# weekday history
wd_spread0 = np.zeros((SEGMENT_LEN[city],), 'int32')   # weekday newly infected person (for record)
wd_spread1 = np.zeros((SEGMENT_LEN[city],), 'int32')   # weekday spread in time 1 and 3
wd_spread2 = np.zeros((SEGMENT_LEN[city],), 'int32')   # weekday spread in time 2
h_infection = np.zeros((household,), 'int32')   # infection in each household
# pre-computation
wd_spread0[sample[temp, 1]] += infective[temp]
wd_spread2[sample[temp, 2]] += infective[temp]
wd_spread1[trace[temp]] += infective[temp]
h_infection[sample[temp, 4]] += infective[temp]

# isolation region with columns: 0.status, 1.time, 2.home
isolation = np.zeros((0, 3), 'int32')
# local lockdown (time limited; -1 if not locked)
lockdown = np.zeros((SEGMENT_LEN[city],), 'int32')
lockdown -= 1
# national lockdown
all_lock = -1

t = 0
total_pop = population   # total population
w_deviation = rng1.choice(7)   # create randomness for the day when the infected individual was introduced
```

```
    # main loop
    while summary[t, 0] != population and t < 1460:    # one year

        t += 1
        period = t % 4
        weekday = ((t // 4 + w_deviation) % 7) // 5

        # print(summary[-1])
        # print('Sub-Model #{} Time:{} Period:{} Weekday:{}'.format(number, t, period, weekday))

        # create and record movement
        # shortcut: if there is no infectious or infected person, the process of creating movement is skipped
        if summary[t - 1, 0] > 0 and summary[t - 1, 2:5].sum() > 0:
            # list of numbers of infected visitors in each segment
            spread = np.zeros(SEGMENT_LEN[city], dtype='int32')
            # sort individuals by the infectious and susceptible
            if policy[0] > 3 and period != 0:    # remove those who are in the lockdown area
                temp = np.isin(sample[:, 1], np.argwhere(lockdown < 0).flatten())
                _filter = [np.argwhere(np.logical_and(temp, status[:, 0] > 1)).flatten(),
                            np.argwhere(np.logical_and(temp, status[:, 0] == 0)).flatten()]
            else:
                _filter = [np.argwhere(status[:, 0] > 1).flatten(), np.argwhere(status[:, 0] == 0).flatten()]

            # national lockdown
            if all_lock == 0:
                all_lock = 480    # return normal for 2 months
                temp = sample[:, 0] < 2
                temp2 = np.logical_and(temp, status[:, 0] > 1)
                wd_spread2[sample[temp2, 2]] += infective[temp2]
                wd_spread2[sample[temp2, 1]] -= infective[temp2]
                for i in np.argwhere(temp).flatten():
                    if status[i, 0] > 1:
                        wd_spread1[trace[i]] -= infective[i]
                    if sample[i, 3]:
                        trace[i] = sample[i, [1, 2]]
                    else:
                        temp1 = sample[i, 1]
                        temp2 = sample[i, 2]
                        trace[i] = np.append(station_path[s_index[temp1]:s_index[temp1 + 1]],
                                             station_path[s_index[temp2]:s_index[temp2 + 1]])
                    if status[i, 0] > 1:
                        wd_spread1[trace[i]] += infective[i]
                all_lock -= 1
            elif 0 < all_lock < 241:    # national lockdown
                if all_lock == 240:
                    temp = sample[:, 0] < 2
                    temp2 = np.logical_and(temp, status[:, 0] > 1)
                    wd_spread2[sample[temp2, 1]] += infective[temp2]
                    wd_spread2[sample[temp2, 2]] -= infective[temp2]
                    for i in np.argwhere(temp).flatten():
                        if status[i, 0] > 1:
                            wd_spread1[trace[i]] -= infective[i]
                        temp1 = sample[i, 1]
                        trace[i] = neighbours[n_index[temp1]:n_index[temp1 + 1]]
                        if status[i, 0] > 1:
                            wd_spread1[trace[i]] += infective[i]
                weekday = 0
                all_lock -= 1
            elif all_lock > 240:
                all_lock -= 1

            if weekday == 0:    # weekdays
                if period == 0:    # 0-6am stay at home
                    status[_filter[1][
                        rng.random(_filter[1].shape[0]) > BASIC_RATE ** (h_infection[sample[_filter[1],
4]])], 1] = 1
                elif period == 2:    # 12am-6pm
                    temp = sample[_filter[1], 2]
                    temp = wd_spread2[temp] * (lockdown[temp] < 0)
                    if policy[2] > 0:
```

```
                        temp[temp > 10] = 10
                    status[_filter[1][rng.random(_filter[1].shape[0]) > mask_in ** temp], 1] = 1
                elif all_lock < 1 or all_lock > 240 or ((t // 4 + w_deviation) % 7) // 6 == 1:    # 6-12am or 6pm-0am
                    temp = np.empty(_filter[1].shape[0], dtype='float64')
                    for i in range(_filter[1].shape[0]):
                        temp1 = trace[_filter[1][i]]
                        temp[i] = (wd_spread1[temp1] * (lockdown[temp1] < 0)).sum()
                    if policy[2] > 0:
                        temp[temp > 10] = 10
                    status[_filter[1][rng.random(_filter[1].shape[0]) > mask_out ** (temp / BETA)], 1] = 1

            else:    # weekend
                if period == 0:    # 0-6am stay at home
                    status[_filter[1][
                        rng.random(_filter[1].shape[0]) > BASIC_RATE ** (h_infection[sample[_filter[1], 4]])], 1] = 1
                elif period == 3:    # 6pm-0am go back home
                    # the infectious
                    for i in _filter[0]:
                        # function: move_home
                        target = sample[i, 1]
                        if target == status[i, 2]:    # already home
                            spread[target] += infective[i]
                        elif sample[i, 3]:
                            spread[[status[i, 2], target]] += infective[i]
                        else:
                            temp = status[i, 2]
                            spread[np.append(station_path[s_index[temp]:s_index[temp + 1]],
                                             station_path[s_index[target]:s_index[target + 1]])] += infective[i]
                    status[_filter[0], 2] = sample[_filter[0], 1]
                    # the susceptible
                    temp = np.empty(_filter[1].shape[0], dtype='float64')
                    for i in range(_filter[1].shape[0]):
                        # function: move_home
                        target = sample[_filter[1][i], 1]
                        if target == status[_filter[1][i], 2]:    # already home
                            temp1 = target
                        elif sample[_filter[1][i], 3]:    # car
                            temp1 = [status[_filter[1][i], 2], target]
                        else:
                            temp1 = status[_filter[1][i], 2]
                            temp1 = np.append(station_path[s_index[temp1]:s_index[temp1 + 1]],
                                                   station_path[s_index[target]:s_index[target + 1]])
                        temp[i] = (spread[temp1] * (lockdown[temp1] < 0)).sum()
                    status[_filter[1], 2] = sample[_filter[1], 1]
                    if policy[2] > 0:
                        temp[temp > 10] = 10
                    status[_filter[1][rng.random(_filter[1].shape[0]) > mask_out ** (temp / BETA)], 1] = 1

                else:    # 6am-6pm
                    # the infectious
                    temp = rng.choice(2, _filter[0].shape[0])    # random move: 50% free and 50% targeted
                    for i in _filter[0][temp == 0]:
                        temp1 = status[i, 2]
                        spread[neighbours[n_index[temp1]:n_index[temp1 + 1]]] += infective[i]
                    for i in _filter[0][temp == 1]:
                        # function: targeted_move
                        target = rng.choice(destinations)
                        if sample[i, 3]:
                            spread[[status[i, 2], target]] += infective[i]
                        else:
                            temp1 = status[i, 2]
                            spread[np.append(station_path[s_index[temp1]:s_index[temp1 + 1]],
                                             station_path[s_index[target]:s_index[target + 1]])] += infective[i]
                        status[i, 2] = target
                    # the infected
                    temp = rng.choice(2, _filter[1].shape[0])
```

```
                tt = np.empty(_filter[1].shape[0], dtype='float64')
                for i in np.argwhere(temp == 0).flatten():
                    temp1 = status[_filter[1][i], 2]
                    temp1 = neighbours[n_index[temp1]:n_index[temp1 + 1]]
                    tt[i] = (spread[temp1] * (lockdown[temp1] < 0)).sum()
                for i in np.argwhere(temp == 1).flatten():
                    # function: targeted_move
                    target = rng.choice(destinations)
                    if sample[i, 3]:    # car
                        temp1 = [status[i, 2], target]
                    else:
                        temp1 = status[i, 2]
                        temp1 = np.append(station_path[s_index[temp1]:s_index[temp1 + 1]],
                                          station_path[s_index[target]:s_index[target +
1]])
                    tt[i] = (spread[temp1] * (lockdown[temp1] < 0)).sum()
                    status[i, 2] = target
                if policy[2] > 0:
                    tt[tt > 10] = 10
                status[_filter[1][rng.random(_filter[1].shape[0]) > mask_out ** (tt / BETA)], 1] = 1

        # update isolation area
        # print(isolation)
        isolation[:, 1] -= 1
        isolation[np.logical_and(isolation[:, 0] == 2, isolation[:, 1] == 0), :2] = [3, 14]
        if policy[0] == 1:
            for i in np.argwhere(isolation[:, 1] == 0).flatten():
                h_infection[isolation[i, 2]] -= 5 - isolation[i, 0]    # equals to 2 if clinical or 1 if subclinical
        isolation = isolation[isolation[:, 1] > 0]

        # update lockdown
        lockdown -= 1

        # update status
        status[:, 1] -= 1
        # enter next stage (for status != 1)
        temp1 = status[:, 1] == 0
        temp2 = np.logical_and(temp1, status[:, 0] == 0)
        status[temp2, 2] = sample[temp2, 1]    # send those who enters exposed stage home

        # change sample
        temp2 = status[:, 0] == 1
        status[np.ix_(temp1, [0, 1])] = np.array([[1, 16], [1, 0], [3, 14], [5, -1], [5, -1]])[status[temp1, 0]]

        # enter next stage (for status == 1)
        temp1 = np.logical_and(temp1, temp2)
        status[np.ix_(temp1, [0, 1])] = rng.choice([[2, 6], [4, 20]], temp1.sum())
        temp2 = np.logical_and(status[:, 0] == 2, status[:, 1] == 6)
        infective[temp2] = 2

        # pre-compute (add)
        wd_spread0 += np.bincount(sample[temp1, 1], None, SEGMENT_LEN[city]).astype('int32')
        wd_spread2 += np.bincount(sample[temp1, 2], infective[temp1], SEGMENT_LEN[city]).astype('int32')
        for i in np.argwhere(temp1).flatten():
            wd_spread1[trace[i]] += infective[i]
        h_infection += np.bincount(sample[temp1, 4], infective[temp1], household).astype('int32')
        # pre-compute (delete)
        if policy[0] > 0:    # delete those who enters clinical stage or are removed
            temp2 = status[:, 0] == 3    # records to be moved to isolation area
            temp3 = status[:, 0] == 5    # used in home isolation
            if policy[0] == 3 and (t + 1) % 12 == 0:
                temp2 = np.logical_and(status[:, 0] > 1, status[:, 0] < 5)
            elif policy[0] > 3:
                if (t + 1) % 12 == 0:
                    temp2 = np.logical_and(status[:, 0] > 1, status[:, 0] < 5)
                else:
                    temp2 = np.logical_and(np.logical_and(status[:, 0] > 1, status[:, 0] < 5),
                                           np.isin(sample[:, 1], sample[temp2, 1]))
                # lockdown
                temp = []
                for i in sample[temp2, 1]:
```

```
                    temp += neighbours[n_index[i]:n_index[i + 1]].tolist()
                # send those who are in lockdown area home
                temp1 = np.isin(sample[:, 1], temp)
                status[temp1, 2] = sample[temp1, 1]
                lockdown[temp] = 55

            # isolation
            isolation = np.append(isolation, np.append(status[np.ix_(temp2, [0, 1])].reshape((np.sum(temp2), 2)),
                                                                        sample[temp2, 1].reshape((np.sum(temp2), 1)),
1), 0)
            temp1 = np.logical_or(temp2, temp3)     # records to remove
            if policy[0] == 1:
                h_infection -= np.bincount(sample[temp3, 4], infective[temp3], household).astype('int32')
            else:
                h_infection -= np.bincount(sample[temp1, 4], infective[temp1], household).astype('int32')
        else:
            temp1 = status[:, 0] > 4
            h_infection -= np.bincount(sample[temp1, 4], infective[temp1], household).astype('int32')
        wd_spread2 -= np.bincount(sample[temp1, 2], infective[temp1], SEGMENT_LEN[city]).astype('int32')
        for i in np.argwhere(temp1).flatten():
            wd_spread1[trace[i]] -= infective[i]

        # remove item
        sample = np.delete(sample, temp1, 0)
        status = np.delete(status, temp1, 0)
        trace = np.delete(trace, temp1, 0)
        infective = np.delete(infective, temp1, 0)
        population = sample.shape[0]

        # summary of the day
        summary = np.append(summary, np.array([[
            len(status[status[:, 0] == 0]),
            len(status[status[:, 0] == 1]),
            len(status[status[:, 0] == 2]) + len(isolation[isolation[:, 0] == 2]),
            len(status[status[:, 0] == 3]) + len(isolation[isolation[:, 0] == 3]),
            len(status[status[:, 0] == 4]) + len(isolation[isolation[:, 0] == 4]),
            total_pop - population,
            isolation.shape[0],
            np.sum(region[lockdown > -1])]]), axis=0)
        if (t + 1) % 28 == 0:
            distribution = np.append(distribution, wd_spread0.reshape((SEGMENT_LEN[city], 1)), 1)

        # policy trigger
        if trigger != -1:
            if trigger == -2:
                if summary[-1, 3] > 0:
                    trigger = 28 * policy0[0]
            elif trigger == 0:    # trigger policies
                policy[:] = policy0[1:]
                if policy[0] > 0:    # remove current clinical cases
                    temp = status[:, 0] == 3
                    if policy[0] > 3:
                        temp = np.logical_and(np.logical_and(status[:, 0] > 1, status[:, 0] < 5),
                                              np.isin(sample[:, 1], sample[temp, 1]))
                    wd_spread2 -= np.bincount(sample[temp, 2], infective[temp],
SEGMENT_LEN[city]).astype('int32')
                    for i in np.argwhere(temp).flatten():
                        wd_spread1[trace[i]] -= infective[i]
                    if policy[0] > 1:
                        h_infection -= np.bincount(sample[temp, 4], infective[temp], household).astype('int32')
                    # isolation and remove
                    isolation = np.append(isolation,
                                          np.append(status[np.ix_(temp, [0, 1])].reshape((np.sum(temp),
2)),
                                                    sample[temp, 1].reshape((np.sum(temp), 1)), 1), 0)
                    sample = np.delete(sample, temp, 0)
                    status = np.delete(status, temp, 0)
                    trace = np.delete(trace, temp, 0)
                    infective = np.delete(infective, temp, 0)
                    population = sample.shape[0]
                mask_out, mask_in = [(0.98, 0.98), (0.99, 0.98), (0.99, 0.99)][policy[1]]    # masks
```

```
                if policy[3] > 0:
                    all_lock = 240
                    temp = sample[:, 0] < 2
                    temp2 = np.logical_and(temp, status[:, 0] > 1)
                    wd_spread2[sample[temp2, 1]] += infective[temp2]
                    wd_spread2[sample[temp2, 2]] -= infective[temp2]
                    for i in np.argwhere(temp).flatten():
                        if status[i, 0] > 1:
                            wd_spread1[trace[i]] -= infective[i]
                        temp1 = sample[i, 1]
                        trace[i] = neighbours[n_index[temp1]:n_index[temp1 + 1]]
                        if status[i, 0] > 1:
                            wd_spread1[trace[i]] += infective[i]
                trigger -= 1
            else:
                trigger -= 1

    # export summary
    np.savetxt(r'Log\temp\summary{}.txt'.format(number), summary, fmt='%d', delimiter=',',
               header='Susceptible, Exposed, Preclinical, Clinical, Subclinical, Removed')
    if distribution.shape[1] == 0:    # avoid the bug that distribution is empty when time is too short
        distribution = np.append(distribution, wd_spread0.reshape((SEGMENT_LEN[city], 1)), 1)
    np.savetxt(r'Log\temp\distribution{}.txt'.format(number), distribution, fmt='%d', delimiter=',',
               header='Week')

    end = time.time()
    print('Sub-Model {} ended in {} seconds. {} seconds per loop. t = {}.'.format(
        number, end - start, (end - start) / t, t))
```

Const.py

```
class City:
    LONDON = 0
    BEIJING = 1


PRE_FILE = {
    City.LONDON: {
        'segment': r'PRE\SegmentLondon.csv',
        'station path': r'PRE\StationLondon.txt',
        'neighbours': r'PRE\NeighbourIndexLondon.txt',
    },
    City.BEIJING: {
        'segment': r'PRE\SegmentBeijing.csv',
        'station path': r'PRE\StationBeijing.txt',
        'neighbours': r'PRE\NeighbourIndexBeijing.txt',
    }
}

MEMORY_SIZE = {
    City.LONDON: {
        'destinations': (12054,),
        'region': (65239,),
        'station_path': (228356,),
        'station_path_index': (65240,),
        'neighbours': (342343,),
        'neighbours_index': (65240,),
        'sample': (332570, 5),
        'status': (332570, 3),
    },
    City.BEIJING: {
        'destinations': (84676,),
        'region': (188660,),
        'station_path': (1110436,),
        'station_path_index': (188661,),
        'neighbours': (1040460,),
        'neighbours_index': (188661,),
        'sample': (997457, 5),
        'status': (997457, 3),
```

```
    },
}

AGE = {
    City.LONDON: [0.16, 0.75, 0.09],
    City.BEIJING: [0.08, 0.46, 0.46],
}

HOUSEHOLD_SIZE = {
    City.LONDON: [0, 0.26, 0.57, 0.74, 0.92, 0.98, 1],
    City.BEIJING: [0, 0.236, 0.552, 0.803, 0.908, 1],
}

SEGMENT_LEN = {City.LONDON: 65239, City.BEIJING: 188660}
BASIC_RATE = 0.98    # rate of infection
BETA = 22    # fix argument for acquaintance
```

Execution and Data Processing

Repeat.py

```
import main9

if __name__ == '__main__':
    for i in range(0, 100):
        print(i)
        main9.base(i, (0, 0, 0, 0, 0), r"Backup\basic model\\")
```

Data Process.py

```
import numpy as np

number = 100
loc = r'Backup\CN-lag0'
brief = np.loadtxt(r'Log\{}\brief\brief-seed0.csv'.format(loc), delimiter=',', skiprows=1, dtype='float', encoding='utf-8')
for i in range(1, number):
    temp = np.loadtxt(r'Log\{}\brief\brief-seed{}.csv'.format(loc, i), delimiter=',', skiprows=1, dtype='float',
encoding='utf-8')
    if temp.shape[0] > brief.shape[0]:
        brief = np.append(brief, np.tile(brief[-1], (temp.shape[0] - brief.shape[0], 1)), 0)
    else:
        temp = np.append(temp, np.tile(temp[-1], (brief.shape[0] - temp.shape[0], 1)), 0)
    brief += temp
brief /= number
np.savetxt(r'Log\{}\brief\brief.txt'.format(loc), brief, fmt='%d', delimiter=',', header='New cases, Current clinical
cases, Isolatied cases, Lockdown')
```

Pre-computation

Neighbours.py

```
import numpy as np
import igraph as ig

# Produce one-step-neighborhood list

# columns: 0.geoid
segment = np.loadtxt(r'PRE\simplified.csv', delimiter=',', skiprows=1,
                     dtype='int32', encoding='utf-8', quotechar='"', usecols=1)
# columns: 0.from, 1.to
```

```
link = np.loadtxt(r'PRE\Link.csv', delimiter=',', skiprows=1, dtype='int32',
                  encoding='utf-8', quotechar='"', usecols=(0, 1))
graph = ig.Graph(n=len(segment), edges=link[link[:, 0] < link[:, 1]])
neighbors = graph.neighborhood(order=1, mindist=0)

for i in range(len(segment)):
    # if i % 100 == 0:
    #     print(i)
    with open(r'PRE\NeighbourIndex.txt', 'a+') as f:
        f.write(str(neighbors[i]))
        f.write('\n')
```

Station.py

```
import numpy as np
import igraph as ig

# This is to save the shortest path from each segment to its nearest station.

# columns: 0.geoid, 1.whether this segment has any station
segment = np.loadtxt(r'PRE\simplified.csv', delimiter=',', skiprows=1,
                     dtype='int32', encoding='utf-8', quotechar='"', usecols=(1, 5))
# columns: 0.from, 1.to
link = np.loadtxt(r'PRE\Link.csv', delimiter=',', skiprows=1, dtype='int32',
                  encoding='utf-8', quotechar='"', usecols=(0, 1))
graph = ig.Graph(n=len(segment), edges=link[link[:, 0] < link[:, 1]])
stations = segment[segment[:, 1] == 1, 0]

# find path
for i in range(len(segment)):
    if i % 100 == 0:
        print(i)
    path = []
    dis = 9999
    temp = graph.get_shortest_paths(v=i, to=stations, output='vpath')
    for j in range(len(temp)):
        if len(temp[j]) < dis:
            dis = len(temp[j])
            path = temp[j]
    with open(r'PRE\Station.txt', 'a+') as f:
        f.write(str(path))
        f.write('\n')
```