Cairo University

Faculty of Economics and Political Science

English Section

Socio-Computing Department

COMPUTATIONAL MODELING FOR THE SOCIAL SCIENCES PROJECT

<u>Project Name:</u> "Modeling the Impact of Economic Resources on Immigrants' Behavior Patterns"

<u>Project Number:</u> 5

Student Name	Email	ID
1. Saif Essam Abdelmaboud	Saif.Essam2020@feps.edu.eg	5200356
2. Yahia Mohamed Quolquela	Yahia.quolqeula2020@feps.edu.eg	5200901
3. Youssef Hatem Mohamed	Youssef.Hatem2020@feps.edu.eg	5200836
4. Shaymaa Sherif Fouad	Shaima.sherif2020@feps.edu.eg	5200389

<u>Under The Supervision of:</u>

Dr. Rasha Hassan

TABLE OF RESPONSIBILITY:

	Task	Saif Essam	Yahia Mohamed	Youssef Hatem	Shaymaa Sherif
1	Report	✓	✓	✓	✓
2	Flowchart and Pseudocode	✓	✓	✓	✓
3	NetLogo Coding	✓	✓	✓	✓
4					

TABLE OF CONTENTS

LIST OF FIGURES

I. Introduction:

Migration, the process of individuals relocating from one place to another, is a multifaceted and widespread phenomenon that has shaped human societies throughout history. It is a dynamic process influenced by a various factor, including economic opportunities, environmental conditions, and infrastructural development. The complex interactions of these factors and their impact on the decision-making process of migrants has long captured the attention of scholars and policymakers alike.

In the modern world, migration has become a defining characteristic of globalization, presenting both challenges and opportunities for societies across the globe. As individuals seek better livelihoods, improved living conditions, and new opportunities, the patterns and dynamics of migration continue to evolve. Understanding the complexities of migration is crucial not only for academic inquiry but also for the formulation of effective policies that address the needs of both sending and receiving communities.

This paper goes deep into the heart of the migration phenomenon, employing the lens of computational modeling and agent-based modeling (ABM) to reveal the complex relationships between economic resources and immigration behavior. We aim to simulate and analyze the accuracy in degrees of difference decision-making processes of immigrants in response to environmental, economic, and infrastructural factors. The use of ABM allows us to capture the individual agency of immigrants and the emergent patterns that arise from their interactions within a dynamic environment.

As we getting started on this exploration, it is essential to recognize that migration is not a homogenous concept but a diverse and context-dependent phenomenon. From rural-to-urban migration driven by economic opportunities to international migration shaped by geopolitical factors, the motivations and consequences of migration are multifaceted. By delving into the complications of migration through computational modeling, we aspire to contribute to the growing body of knowledge that informs our understanding of this complex and ever-evolving facet of human behavior.

II. Defining the Problem:

The problem at the core of this research revolves around understanding the impact of economic resources on immigrants' behavior. Migration, a complex phenomenon influenced by a myriad of factors, is particularly sensitive to economic conditions. To address this complexity, the research seeks to model and simulate the decision-making processes of individual immigrants, considering their preferences, environmental factors, and the economic resources available in their surroundings.

The primary question to be answered is: "*How do environmental, economic, and infrastructure factors influence the migration behavior of agents (immigrants), and how does this impact the overall pattern of migration?*"

To answer this question, the model examines the interactions between agents (immigrants) and their environment, exploring how specific economic indices impact the decision to migrate. Additionally, the research seeks to identify thresholds and critical livability values that trigger migration, providing insights into the conditions under which individuals are compelled to seek new environments.

III. Why Agent-Based Modeling (ABM)?

ABM is a model that allows for the modeling of individual agents, each with unique characteristics, preferences, and behaviors. It enables a granular analysis of individual-level dynamics in migration, which often involves personal factors like economic conditions, environmental preferences, and infrastructure. ABM captures emergent behavior from interactions within agents, allowing for a better understanding of macro-level patterns. It also allows for the representation of dynamic environments, where migration occurs in response to changing conditions.

ABM is well-suited for modeling complex adaptive systems, where agents adapt their behavior in response to internal and external stimuli. It captures heterogeneity in individuals' preferences, motivations, and decision-making processes, allowing for a better understanding of overall migration patterns. ABM also facilitates scenario exploration, allowing researchers to manipulate parameters and observe changes in migration patterns. It offers insights into individual decision-making processes, providing insights into micro-level motivations driving migration.

IV. Identifying the Agents and Interactions:

The individual entities in the model are immigrants. Each immigrant is conceptualized as a turtle possessing distinct attributes and characteristics that influence their behavior. These attributes include environmental, economic, and infrastructure preferences, as well as a satisfaction score.

- **Agent's Behavior:**

The behavior of the agents (immigrants) is characterized by their decision-making process regarding migration. The key elements of their behavior include:

a. **Movement Towards Higher Livability:** Agents assess the livability of their current patch based on environmental, economic, and infrastructure scores. If their satisfaction with the current patch falls below a predefined threshold, they explore nearby patches to find locations with higher livability scores.
b. **Calculation of Satisfaction Score:** Agents calculate their satisfaction scores by taking the dot product of their preference vector and the livability vector of their current patch. This reflects how well the current patch aligns with the agent's preferences in terms of environmental, economic, and infrastructure conditions.
c. **Consideration of Threshold:** Agents only initiate migration if their satisfaction with the current patch is below a defined threshold. This threshold represents the minimum livability value required for an agent to decide to explore nearby patches in search of a more suitable environment.

- **Agent-Environment Interaction:**

Agents interact with their environment (patches) through the following mechanisms:

a. **Assessment of Livability:** Agents evaluate the livability of their current patch based on the environmental, economic, and infrastructure scores assigned to that patch. This evaluation influences their satisfaction and migration decisions.
b. **Movement Towards Better Conditions:** Agents interact with the environment by moving towards patches with higher livability scores. The decision to explore nearby patches is driven by the desire to find a location that better matches their preferences.

c. **Dynamic Environment:** The environment is dynamic, with randomly assigned environmental, economic, and infrastructure values to patches. This dynamic nature simulates the changing conditions that may influence agents' decisions over time.

V. Designing the Environment:

The environment in the model is a two-dimensional space represented by patches in NetLogo. This spatial context is divided into two halves, each simulating a different set of conditions to capture the heterogeneity of economic resources and livability scores. The design of the environment involves the following key elements:

a. **Spatial Division:** The world is divided along the horizontal axis. The lower half represents a region with fewer resources and lower livability scores, while the upper half symbolizes a region with more abundant resources and higher livability scores. This spatial division introduces a gradient of economic conditions and livability across the environment.

b. **Random Assignment of Economic Resources:** Economic resources are represented by the variables "environment," "economic," and "infrastructure" assigned to each patch. In the lower half, these variables are randomly assigned values between 1 and 5, simulating a region with limited resources. In the upper half, values between 6 and 10 are randomly assigned, representing a region with more abundant resources.

c. **Calculation of Livability Score:** The livability score for each patch is determined by calculating the Euclidean L2 norm of the vector composed of environmental, economic, and infrastructure variables. This score reflects the overall desirability of a patch, considering the combination of economic resources.

d. **Color Representation:** The color of each patch is assigned based on its livability score, using a gradient from green to brown. Patches with higher livability scores are assigned greener colors, while patches with lower livability scores are assigned browner colors. This visual representation allows for an intuitive understanding of livability across the environment.

e. **Dynamic Nature:** The environment is dynamic, as economic, environmental, and infrastructure conditions are randomly assigned to patches. This dynamic nature simulates the real-world variability and unpredictability of resource distribution and livability.

VI. Obtaining Data: Rules and Behaviors:

The data in the model is obtained through a combination of real and artificial mechanisms. The rules governing the behavior of agents and the dynamics of the environment are designed to emulate real-world influences on migration. Here's how data is obtained within the model:

1. Artificial Data Generation:

a. **Environment Variables:** The model generates artificial economic, environmental, and infrastructure variables for each patch. These variables are randomly assigned within specified ranges, representing the artificial creation of a diverse and dynamic environment.

b. **Agent Preferences:** Individual preferences of agents (env, econ, infra) are also artificially generated. The preferences are assigned as random values between 0 and 1, reflecting the artificial diversity of agent inclinations towards environmental, economic, and infrastructure factors.

2. Calculation of Livability Score:

- **L2 Norm Calculation:** The livability score for each patch is calculated using the L2 norm (Euclidean norm) of the vector composed of environmental, economic, and infrastructure variables. This mathematical calculation synthesizes the artificial data to represent an overall livability score for each patch.

3. Agent Satisfaction Calculation:

- **Dot Product Calculation:** The satisfaction score of each agent is calculated through the dot product of the agent's preference vector and the livability vector of its current patch. This computation, while based on artificial preferences and livability scores, mirrors the real-world process of individuals assessing their satisfaction with their environment.

4. Random Movement and Exploration:

a. **Random Movement:** Agents move randomly within their environment, simulating the unpredictability of human movement. This randomness is reflected in both the initial placement of agents and their subsequent exploration of nearby patches.

b. **Exploration Based on Satisfaction:** Agents explore nearby patches when their satisfaction falls below a predefined threshold. This action emulates the real-world behavior of individuals seeking new environments when their current conditions become unsatisfactory.

VII. Define the Time Step:

The time step in the model is represented by the tick command as (weeks), suggesting a discrete time progression. The movement of agents and the evaluation of their satisfaction scores occur within each time step.

VIII. Flowchart:

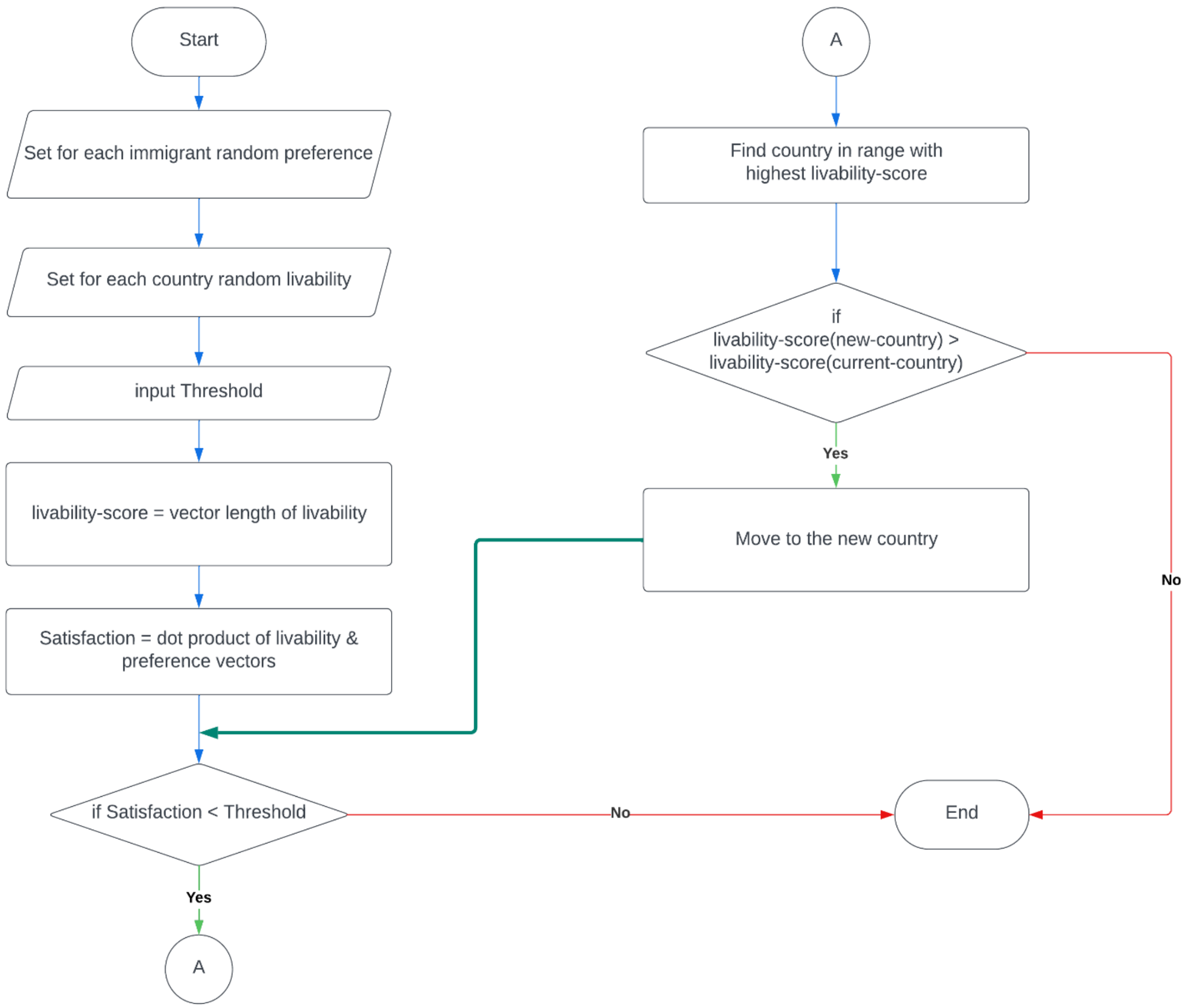

Figure (1): Flow Chart for the Immigrants Model

IX. Pseudocode:

Input for countries: “livability” as a vector of (environment, economic, infrastructure)

Input for immigrants: “preference” as a vector of weights for (environment, economic, infrastructure)

livability-score = $\sqrt{\text{environment}^2 + economic^2 + infrastructure^2}$

Satisfaction = (environment * environment weight) + (economic * economic weight) + (infrastructure * infrastructure weight)

While Satisfaction < Threshold then

  Find country in range with highest livability-score

  If livability-score(new-country) > livability-score(current-country) then

    Move to the new country

  Else End

End While

X. NetLogo Interface and Code:

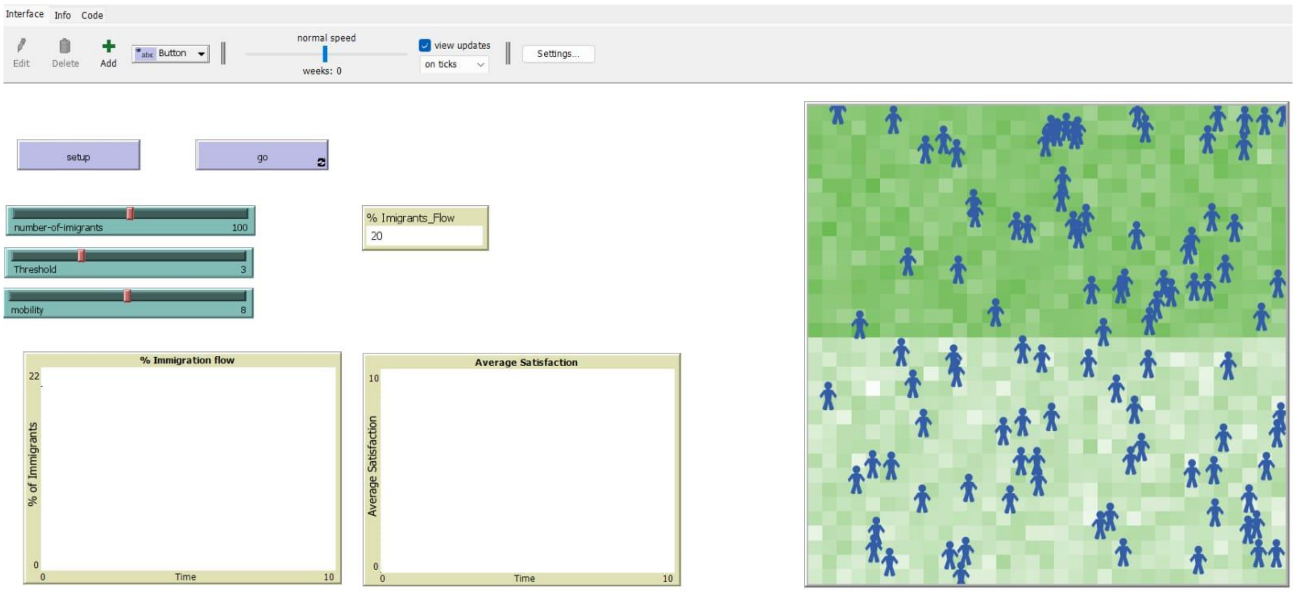

Figure (2): NetLogo Model Interface

Figure (2): Shows the Interface of the Immigrants Model.

- The patches and turtles together constitute the whole world.
- The interface contains the "Setup" button which initialize the model and the "Go" button that advances the model by one time step and it does not stop until the stop condition achieved.
- There are three sliders that changes the inputs for the model which are:
    1. **Number of Immigrants:** Initial number of immigrants
    2. **Mobility:** Mobility radius for exploring nearby patches
    3. **Threshold:** Threshold for satisfaction triggering migration
- Moreover, it contains the output when running the model provided with some graphs.
- One of them represents the percentage of immigration flow each week and the other represents the average satisfaction level for each week.

```
globals [
  environment
  economic
  infrastructure
  env
  econ
  infra
  average
]

patches-own[ livability
  livability-score]
;determining which variables that the breed of patches will take


turtles-own[preference
  satisfaction]
  ;determining which variables that the breed of turtles will take
```

Figure (3): Code for Global Variables, Turtles and Patches Properties.

Figure (3): Shows The global variables, turtles and patches properties. (i.e.: What properties the agents own).

- The global block is used to define global variables that can be accessed and modified by all agents in the model. And the global variables we have environment and economic and infrastructure and env and econ and infra and the average satisfaction.
- The patches-own and turtles-own declarations are used to define variables that are specific to patches and turtles, respectively. These variables are local to each patch or turtle and the following patch-specific variables are defined:
  1. **Livability:** This variable represents some aspect of livability associated with each patch. It could be a measure of how suitable or desirable a patch is for agents to occupy or interact with.
  2. **Livability-score:** This variable represents a numeric score or value associated with each patch's livability. It could be a calculated value based on various factors or criteria relevant to the model.
- While the following turtle-specific variables are defined:
  1. **Preference:** This variable represents a preference or choice associated with each turtle. It could be a factor that influences the turtle's decision-making or behavior in the model.

2. **Satisfaction:** This variable represents the satisfaction level of each turtle. It could be a measure of how content or fulfilled a turtle is based on its current state or conditions in the model.

```
;starting to setup defreent setting that we will need in the model
to setup
  clear-all
  setup-environmnet
  create-immigrants
  reset-ticks
end
```

Figure (4): Code for Setup Procedure.

Figure (4): Shows the Setup Procedure for the Immigrants Model.

- The code you provided appears to be a setup procedure in Net Logo. The setup procedure is typically used to initialize the model and set up the initial state of patches, turtles, and other variables. Let's break down the code:
    1. **Clear-all:** This command clears all patches and turtles from the model, resetting the simulation to its initial state.
    2. **Setup-environment:** This is likely a user-defined procedure that sets up the environmental conditions or properties of the model. The specific details of this procedure would depend on how you've defined it elsewhere in your code.
    3. **Create immigrants:** Creates and initializes turtles representing refugees in the model. It could involve specifying their initial positions, attributes, or behaviors.
    4. **Reset-ticks:** This command resets the internal simulation clock, known as ticks, back to zero. It is typically used in our models where time or iteration is a relevant factor.

```
to setup-environmnet
  ask patches with [pycor <= 0 ][
    set livability (list environment economic infrastructure)
    set environment random 5
    set economic random 5
    set infrastructure random 5
    let variables-square item 0 livability ^ 2 +  item 1 livability ^ 2 +  item 2 livability ^ 2
    set livability-score sqrt (variables-square)
    set pcolor scale-color green livability-score 40 0
  ]
  ask patches with [pycor > 0 ][
    set livability (list environment economic infrastructure)
    set environment  5 + random 6
    set economic 5 + random 6
    set infrastructure 5  + random 6
    let variables-square item 0 livability ^ 2 +  item 1 livability ^ 2 +  item 2 livability ^ 2
    set livability-score sqrt (variables-square)
    set pcolor scale-color green livability-score 40 0
  ]

end
```

Figure (5): Code to Setup the Environment.

Figure (5): Shows how the environment has been set up.

- Starting to setup the environment which the turtles (Immigrants) lives by splitting our environment into two halves.
- In the first half that contains line of center (0,0) and all points that bellow center to be the part of the word that does not contain much resource and less livability scores.
- To do this we have made vector livability that contain variables environment, economic, and infrastructure and assign different numbers from 1 to 5 to each variable to choose from it randomly to each patch.
- Then calculating L2 norm of vector livability to be able to calculate livability-score assign colors to patches in range from 40 to 0 according to livability-score.
- We have made the same progress in the second half that contains points above center to be the part of the word that contain much resource and higher livability scores but the only change is that we have made variables environment, economic, and infrastructure assign different numbers from 6 to 10.

```
to-report dot [vector1 vector2]
  let dot-product item 0 vector1 * item 0 vector2 + item 1 vector1 * item 1 vector2 + item 2 vector1 * item 2 vector2
report dot-product
end
```

Figure (6): Code for Creating Dot-Product Function

Figure (6): Shows the creation for “dot-product function”.

- We have found that there is so built in function in the NetLogo that makes a dot product for two vectors thus, we have decided to build a function called “dot” that enables us to get the inner product between two vectors by making summation of multiplication of each element in vector to the corresponding element in other vector.

```
to create-imigrants
  create-turtles number-of-imigrants [
   set color red
   set shape "person"
   set size 2
   setxy random-xcor random-ycor
   set preference (list env econ infra)
   let env1  random-float 1.1
   let econ1  random-float 1.1
   let infra1   random-float 1.1
   let x (env1 + econ1 + infra1)
   set env env1 / x
   set econ econ1 / x
   set infra infra1 / x
   set satisfaction dot livability preference
  ]
end
```

Figure (7): Code for Creating Agents (Immigrants)

Figure (7): Shows how the immigrants, by all their properties, have been created in the model.

- Creating turtles that have “person” shape, “red” color, and its size is 2, making them randomly distributed all over the world.
- Furthermore, for each turtle we have made a “preference” vector that composed of the three variables (env, econ, infra), making those variables take a decimal point from 0 to 1 to choose from it randomly and make variable “x” that is a summation of another 3 variables (env1, econ1, infra1) while assigning weights to each turtle for each of the three variables (env, econ, infra) to make different preference for each turtle.
- We have also assigned “satisfaction score” to each turtle by making inner product between the two vectors (“livability” and “preference”).

```
to-report neigh-mobility
  let nearby-patches patches in-radius mobility
  report nearby-patches
end
```

Figure (8): Code for Creating Function to Determine Neighbors.

<u>Figure (8):</u> Shows creating another function for determining neighbors.

- We have built another function called “neigh-mobility” which determines the nearby patches that are all the patches that in radius of size mobility.

```
to avg-satisfaction
   let immigrants turtles
 set average mean [satisfaction] of immigrants
end
;; to calculate mean of satisfaction of immigrants
```

Figure (9): Code for Calculating Average Satisfaction.

<u>Figure (9):</u> Shows how to calculate the mean of satisfaction of immigrants that moved to new patches (countries).

```
to go
  if all? turtles [ satisfaction > Threshold] [
   stop]

 ask turtles [
    if satisfaction < Threshold [
    let next-patch max-one-of neigh-mobility [
      livability-score
      ]
      face next-patch
      fd 0.1
  set satisfaction dot livability preference
  avg-satisfaction]]
  tick

end
```

Figure (10): Code for "Go" Procedure.

Figure (10): Shows the “GO” Procedure. (i.e.: how the immigrants behave):

- In this code, it asks all turtles if there any turtle that has a satisfaction higher than the threshold then it will not move.
- The patches will move only when its satisfaction become higher than its threshold and then stop.
- It also asks all turtles that if satisfaction is less than the threshold to see if the next that in the range of “neigh-mobility” that has maximum livability-score then the turtle will go forward by a distance of 0.1 units in the direction it is facing.
- Lastly, it updates its satisfaction and then repeat the steps again.

XI. Explaining Outputs:

- Figure (11): Shows the behavior space by varying the variable “Threshold” in 3 different time (3, 7, 10), holding the other variables constant at specific level, where the “mobility” is set to be 8, and the “number of immigrants” is set to be 100.
- By repeating it 10 times, we have got a data containing 10 values for each threshold value.
- To analyze these data, we took an average of the 10 repeated values at each threshold level and we configure the following table.

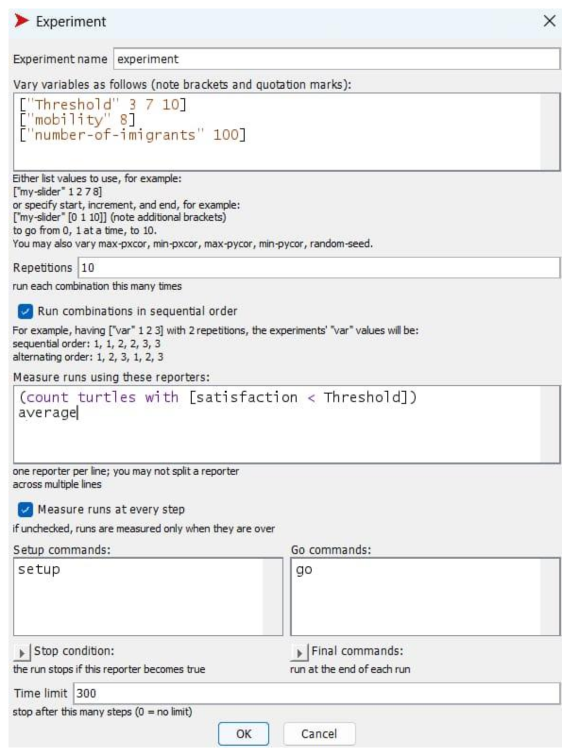

Figure (11): Behavior Space Varying "Threshold" Parameter

Table 1. Immigrants Satisfaction Data			
Threshold	3	7	10
Average Satisfaction	5.849585409	6.958120987	7.711999855
Average Number of dissatisfied Immigrants	18	60	100

Table 1. Shows how the "threshold" parameter affects both the "average satisfaction level" and the "number of dissatisfied immigrants" in the model.

- Holding for the number of immigrants (100) and mobility (8), we can see that as the level of threshold increases, the average number of dissatisfied immigrants increase. Therefore, they tend to move to other countries with better livability, hence their average satisfaction tend to increase.

XII. Suggested Scenarios (Extensions):

1) Disasters Extension:

The model developed in the context of this project addresses the livability of a given area. The greater the livability of a specific patch, the higher the overall quality of life in that area, making it more financially rewarding for the agents involved. Following a similar approach, we can replicate disasters by abruptly and significantly diminishing the livability of patches within a specific region. Such as:

- ✓ **Environmental Disasters:** Any natural calamity qualifies as an environmental disaster, encompassing events like floods, droughts, storms, forest fires, and more. When evaluating their impact on livability, such disasters would reduce the environmental component of the livability vector to zero and affects the other component too. This approach allows us to concentrate on understanding the responses of the agents within the system to environmental catastrophes.
- ✓ **Economic Disasters:** Essentially, an economic crisis or depression can be viewed as a disaster. Observations indicate that any type of economic downturn compels people to abandon regions that were previously thriving and densely populated.

2) Controlled Migration Influx Extension:

A pivotal objective was to investigate the feasibility of developing an Agent-Based Model (ABM) capable of assisting policymakers in testing the potential outcomes of their policy decisions within a simulated environment. This capability would empower policymakers to formulate more refined and carefully considered policies, minimizing the risk of unintended consequences. In line with this objective, the ABM's latest significant feature allows the restriction of immigrant flow into a specific region, specifically earmarked for this purpose in the top-right quadrant of the simulation. Users are provided with controls to set a limit on the number of agents permitted to enter this region.

3) Social Network Influence:

We can also introduce social network interactions among agents, to study how social connections affect migration decisions. Which shows how the social network for each immigrant affects their behavior in traveling to another country not only considering the economic resources of this country.

XIII. Conclusion:

In conclusion, this study has presented an agent-based simulation model that shed the lights on the complexities of migration dynamics, focusing on the economic factors influencing immigrants' behavior. Through the lens of the model and the NetLogo simulation platform, we have explored how environmental, economic, and infrastructure conditions impact the migration decisions of agents.

The model successfully captures the nuanced interplay between economic resources, environmental livability, and individual preferences. Findings from the simulation underscore the importance of these factors in shaping migration patterns, providing insights into the dynamics of human mobility.

Our exploration of the model aligns with the broader field of socio-computing and statistics, emphasizing the utility of agent-based modeling in understanding social phenomena. By extending simple models to incorporate real-world complexities, we contribute to the evolving discourse on computational modeling for social sciences.

As we reflect on the limitations and assumptions embedded in our model, we recognize the need for ongoing refinement and validation. Future research could further enrich the model by incorporating social network dynamics, exploring additional economic feedback loops, and considering some environmental disasters.

In the broader context, this paper contributes to the discourse on migration studies, leveraging computational tools to gain a deeper understanding of the intricate factors influencing migration decisions. As we move forward, embracing the interdisciplinary nature of socio-computing, we anticipate further advancements in modeling techniques and a more nuanced comprehension of the socio-economic dimensions of human migration.

In conclusion, this agent-based simulation model serves as a valuable tool for researchers and policymakers alike, offering insights into the economic empirical evidence and long-term effects of migration. The journey towards understanding the complexities of migration dynamics continues, and this model represents a meaningful step in that direction.

XIV. List of References:

1. Güngör, Ö., Günneç, D., Salman, F. S., & Yücel, E. (2022). **Prediction of Migration Paths Using Agent-Based Simulation Modeling: The Case of Syria**. Proceedings of the International Conference on Industrial Engineering and Operations Management, Istanbul, Turkey, March 7-10, 2022.
2. Deshmukh, V. (2018). **Modeling Human Migration Dynamics in NetLogo**. San Jose State University, Master’s Projects 627.
3. Gilbert, N., & Troitzsch, K. G. (2005). **Simulation for the Social Scientist**. Open University Press.
4. Rand, W. (2015). **An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo.** MIT Press.