

# Configuration Manual

MSc Research Project Cloud Computing

Pratyush Nigel Baxla Student ID: 21179158

School of Computing National College of Ireland

Supervisor: F

Punit Gupta

#### National College of Ireland Project Submission Sheet School of Computing



Student Name:	Pratyush Nigel Baxla
Student ID:	21179158
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Punit Gupta
Submission Due Date:	14/08/2023
Project Title:	Configuration Manual
Word Count:	564
Page Count:	5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Pratyush Nigel Baxla
Date:	9th August 2023

#### PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).Attach a Moodle submission receipt of the online project submission, to<br/>each project (including multiple copies).You must ensure that you retain a HARD COPY of the project, both for

your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

## Configuration Manual

#### Pratyush Nigel Baxla 21179158

This configuration manual provides instructions for setting up and configuring the simulation environment for the research project on task scheduling in Fog and Edge computing environments. The simulation environment is based on the PureEdgeSim simulator Mechalikh et al. (2019) and incorporates custom implementations for algorithm and data analysis.

## 1 Introduction

**Purpose of the Configuration Manual** The aim of this configuration manual is to guide the users through the process of setting up the simulation environment and custom implementations used in the research project. It provides detailed instructions to ensure the successful execution of simulations, custom algorithm implementations, and data analysis.

## 2 System Requirements

Before proceeding with the installation, ensure that your system meets the following requirements:

- Operating System: Windows, MacOS or Linux
- Java Runtime Environment (JRE) version 8 or higher
- Python version 3.7.x or higher
- Python package manager pip
- Visual Studio Code, Jupyter Notebook, or a compatible environment for running Python scripts

## 3 Installation and Setup

#### 3.1 PureEdgeSim Simulator

Since we have done our implementation on PureEdgeSim Simulator, we have provided a copy of the modified PureEdgeSim called "myPureEdgeSim".

eclipse-workspace - myPureEdgeSim/PureEdgeS	m/examplex/Example8.jsva - Eclipse IDE	– o ×
Eile Edit Source Refactor Navigate Se		
🖬 = 📾 🗞 😥 💷 😥 🗞 = 😥 = 🗞 = 🏪 =	# C +   C M   1   1   1   1 +   1 +   + + + + +   1   1	् । 🖬 🕴
😫 Package Explorer × 👘 🖻 🖻 🗖	D Example8.iava X 🗈 simulation parameters.properties D Example8FuzzyLogicOrchestratoriava 💀 Problems 🗢 Javadoc 🖻 Declaration 🖉 Terminal 💻 Console	
w 💥 myPureEdgeSim	20 * DuraEdgaSim: A Simulation Framework for Derformance Evaluation of Cloud Edge and Migt (	omputing
PureEdgeSim	2) performed a similar of the second se	l
> III com.mechalikh.pureedgesim	21 package examples,	
>  com.mechalikh.pureedgesim.datacent		
> II com.mechalikh.pureedgesim.energy	23 import com.mechalikh.pureedgesim.simulationmanager.Simulation;	
> = com.mechalikh.pureedgesim.locationr		
> E commechalikh.pureedgesim.network	26• * In this example we show how to implement a Fuzzy Logic based orchestration	
> II commechalikh.pureedgesim.scenarior	43 public class Example8 (	
> II com.mechalikh.pureedgesim.simulatic	44	
> 🐉 com.mechalikh.pureedgesim.simulatic	45 // Pelow is the neth for the settings folder of this example	
> # com.mechalikh.pureedgesim.simulatic	40 77 below is the path for the setting force of chirs chample	
> II commechalikh.pureedgesim.taskgene	<pre>private static string settingspath = "PureEdgesim/examples/Examples_settingspath";</pre>	
> 38 com.mechalikh.pureedgesim.taskorche	47	
<ul> <li>examples</li> </ul>	48 // The custom output folder is	
> 🗾 Example1 java	49 private static String outputPath = "PureEdgeSim/examples/Example8_output/";	
> Example2.java	50	
> D Example2CustomMobilityModel.jan	51• public static void main(String[] args) {	
> 🖬 Example3.java	52	
> 🗾 Example4java	52 // Operate a DupeRder simulation	
<ul> <li>Example4CustomComputingNode.j</li> </ul>	55 // Cleate a Purendgesim Simulation	
> D ExampleSjava	54 Simulation sim = new Simulation();	
> 2 Example6 Java	55	
> Di Example/ Java	56 // changing the default output folder	
<ul> <li>Example/CachingDevicesava</li> <li>Example/ClusteringDevicesava</li> </ul>	57 sim.setCustomOutputFolder(outputPath);	
<ul> <li>Example/ClusteringDevice.java</li> <li>Example/ClusteringDevice.java</li> </ul>	58	
<ul> <li>Example0 custominetworkinouelge</li> <li>Example0 isua</li> </ul>	59 /** if we want to change the path of all configuration files at once • */	
<ul> <li>Example 8Fuzzul onic Orchestratoria</li> </ul>	so	
A examples Example3 output	() () hereing the simpletic estring follow	
examples Example3 settings	oi // changing the simulation settings folder	
> examples.Example6 settings	62 sim.setCustomSettingsFolder(settingsPath);	
A examples.Example7 settings	63	
> Recomples.Example8_output	64 // telling PureEdgeSim to use the custom <u>orchestrator</u> class	
> sexamples.Example8_settings	65 sim.setCustomEdgeOrchestrator(Example8FuzzyLogicOrchestrator.class);	
	66	
	67 // launching the simulation	
	68 gim launchGimulation().	
	Juin Tuthentinutation () /	
	Writable Smart Insert 43:1:1753 :	

Figure 1: PureEdgeSim Simulator

### 3.2 Installing Python and pip

- 1. Download and install python from the official website  $^{1}$ .
- 2. Verify the installation by opening a terminal or command prompt and entering 'python -version'.
- 3. Generally, pip is automatically installed if you are using Python downloaded from the official website.

## 4 Running Simulations and Analysis

#### 4.1 Running the Customized Example 8

- 1. Open Eclipse IDE.
- 2. Go to 'File' >'Open Projects from File System'.
- 3. Navigate to 'PureEdgeSim' >'examples' >'Example8.java'.
- 4. Right-click on the project and select 'Run As' >'Java Application'.
- 5. After completion, the output should display "Data file written".

<sup>1</sup>https://www.python.org/downloads/



Figure 2: Highlighting the Example8 file location

Simulation - Loading simulation files
EdgeDevicesParser - Checking file: PureEdgeSim/examples/Example8_settings/edge_devices.xml
EdgeDevicesParser - PureEdgeSim/examples/Example8_settings/edge_devices.xml file successfully Loaded!
DatacentersParser - Checking file: PureEdgeSim/examples/Example8_settings/edge_datacenters.xml
DatacentersParser - PureEdgeSim/examples/Example8_settings/edge_datacenters.xml file successfully Loaded!
DatacentersParser - Checking file: PureEdgeSim/examples/Example8_settings/cloud.xml
DatacentersParser - PureEdgeSim/examples/Example8_settings/cloud.xml file successfully Loaded!
ParametersParser - Checking simulation properties file
ParametersParser - Properties file successfully Loaded propoerties file!
ApplicationFileParser - Checking applications file.
ApplicationFileParser - Applications XML file successfully loaded!
SimulationThread - Initializing the Network Module
SimulationThread - Initializing the Datacenters Manager Module
DataCentersManager - Generating computing nodes
DataCentersManager - Creating the network topology
SimulationThread - Initializing the Task Generator
SimulationThread - Initializing the Task Orchestrator
SimulationThread - All modules were successfully launched
task list15090nodes2
Data file writen

Figure 3: Console showing "Data file written"

## 4.2 Running Algorithm Implementations with mealpy

**MEALPY** is the largest Python library for a wide range of advanced population-based meta-heuristic algorithms. In the field of approximate optimization, population meta-heuristic algorithms (PMA) are the most widely used algorithms Van Thieu and Mirjalili (2023).

- 1. Open the Python implementation of the selected algorithm (ex: 'copy of mealpy\_<algorithm >.ipynb')
- 2. Run the application using Visual Studio Code, Jupyter Notebook, or an alternative solution.

3. Press Enter in the console window of Eclipse to resume execution after the ipynb program finishes running.



Figure 4: Algorithm Implementation

#### 4.3 Collecting and Analyzing Results

- 1. Results will be saved in the Example 8 outputs folder ('examples.Example8\_output') for the modified Example 8.
- 2. Each output is named with the 'current date' inside of which the results are dumped in an excel file 'Sequential\_simulation'.
- 3. Compile all algorithm results into a single CSV file.

																									_	_	
× 🔊	utoSave (	• o#	回じ、		Sequenti	al_simulatio	n - Exce															Pratyush N	ligel Baxla	(🕘 🛛 🖓			
File	Home	Insert	Page L	ayout Fi	ormulas	Data R	eview	View	Autor	ate Hel	5													무여	omments	🖻 Sha	re v
(2)	X																	FTTP.			Σ Aut	oSum ~			a   r	18	
	0	Calibri			9 A A		97		εş wra	o lext	6	eneral			Condi	Honal	Eormat ar			Eormat	E Fill					©] ≓tiuitu	
Paste		BI		≝ •   ∞	× A ×	= =			🖭 Mer	ge & Center	~   E	° ~ %	5 9	.00 -00	Format	ting *	Table ~	Styles ~	insert De		🞸 Clea		ilter * Sele	ct~ Da	ta sens		
Cipb	oard 🔤			ont	6			Alignm	ient			N	mber	5			Styles		ه ا	ells		Editin	9	Anal	ysis Sent	ativity	
			∕ fx][	Orchestrat	tion archite																						
	A	в	с	D	E	F	G		н	I.	J		<	L	M		N	0	Р	Q	R	S	т	U	v	w	
1 Orc	nestrat Or	chestrat I	dge devi	Total task	: Average e	o Total tas	k: Averag	pe w Ni	umber of	Tasks succ	Task not	e Task	s faile	Tasks faile	e Tasks fa	ile Ta	sks not ( T	otal task:	Tasks succ 1	otal task: T	asks succ 1	otal tasks	Tasks succ	Network u	Wan usage	Lan usage	e To
2 ALL 3	OP	т	100	5340.375	0.5261	1	0	0	10150	9499		0	651	0		0	0	798	798	1985	1985	7367	6716	77.28603	0.57456	77.2860	3
4																											
5																											- 1
7																											-
8																											
9																											- 1
11																											
12																											
13																											_
14																											
16																											
17																											
18																											
20																											
21																											
22																											
24																											
25																											
26																											
20																											-
		Seque	ntial_sim	ulation																							
Ready	Th Arranii	hiller Linner	dable																					- m			10030

Figure 5: Results of a single algorithm (all will be named 'OPT'

Orchestrat Orchestrat E	dge devic T	otal tasks	Average ex	Total tasks	Average w I	Number of 1	asks succ	Fask not e	Tasks faile	Tasks faile	Tasks faile	Tasks not	Total tasks	Tasks succ	Total tasks	Tasks succ 1	Fotal task: Ta	sks succ	Network u	Wan usage	Lan usage
EDGE_ANI Hybrid Gre	50	797.5	0.1614	0.0375	0	4940	4940	0	0	0	0	0	1740	1740	3200	3200	0	0	25.21231	16.536	25.21231
EDGE_ANE Honey Bac	50	797.5	0.1614	0.025	0	4940	4940	0	0	0	0	0	1870	1870	3070	3070	0	0	21.90154	20.84	21.90154
EDGE_ANI Sand Cat S	50	797.5	0.1614	0	0	4940	4940	0	0	0	0	0	2500	2500	2440	2440	0	0	15.68	28.928	15.68
EDGE_ANI Artificial R	50	797.5	0.1614	0	0	4940	4940	0	0	0	0	0	2540	2540	2400	2400	0	0	19.44	24.04	19.44
EDGE_ANI Dwarf Mor	50	797.5	0.1614	0.0125	0	4940	4940	0	0	0	0	0	1810	1810	3130	3130	0	0	22.89846	19.544	22.89846
EDGE_ANI Genetic Al	50	797.5	0.1614	0.3875	0.0001	4940	4940	0	0	0	0	0	2580	2580	2360	2360	0	0	23.2	19.152	23.2
EDGE_ANI Optimized	50	797.5	0.1614	0.0625	0	4940	4940	0	0	0	0	0	1600	1600	3340	3340	0	0	24.26462	17.768	24.26462
EDGE_ANI Parallel Pa	50	797.5	0.1614	0	0	4940	4940	0	0	0	0	0	2600	2600	2340	2340	0	0	18.44308	25.336	18.44308
EDGE ANI Hybrid Gre	100	1610	0.1586	0.85	0.0001	10150	10150	0	0	0	0	0	5980	5980	4170	4170	0	0	32.65846	59.24	32.65846
EDGE_ANE Honey Bac	100	1610	0.1586	51.625	0.0051	10150	10150	0	0	0	0	0	3500	3500	6650	6650	0	0	53.60615	32.008	53.60615
EDGE ANI Sand Cat S	100	1610	0.1586	5.85	0.0006	10150	10150	0	0	0	0	0	4520	4520	5630	5630	0	0	39.57538	50.248	39.57538
EDGE_ANE Artificial R	100	1610	0.1586	16.775	0.0017	10150	10150	0	0	0	0	0	4490	4490	5660	5660	0	0	45.64923	42.352	45.64923
EDGE_ANI Dwarf Mor	100	1610	0.1586	7.4625	0.0007	10150	10150	0	0	0	0	0	5050	5050	5100	5100	0	0	41.74154	47.432	41.74154
EDGE_ANI Genetic Al	100	1610	0.1586	7.325	0.0007	10150	10150	0	0	0	0	0	4900	4900	5250	5250	0	0	42.11077	46.952	42.11077
EDGE_ANI Optimized	100	1610	0.1586	0.85	0.0001	10150	10150	0	0	0	0	0	5950	5950	4200	4200	0	0	33.15692	58.592	33.15692
EDGE_ANI Parallel Pa	100	1610	0.1586	36.775	0.0036	10150	10150	0	0	0	0	0	3740	3740	6410	6410	0	0	51.74154	34.432	51.74154
EDGE_ANE Hybrid Gre	200	3220	0.1586	186.525	0.0092	20300	20300	0	0	0	0	0	10560	10560	9740	9740	0	0	76.81231	103.536	76.81231
EDGE ANE Honey Bac	200	3220	0.1586	188.975	0.0093	20300	20300	0	0	0	0	0	10450	10450	9850	9850	0	0	75.36615	105.416	75.36615
EDGE_ANI Sand Cat S	200	3220	0.1586	347.5875	0.0171	20300	20300	0	0	0	0	0	9370	9370	10930	10930	0	0	84.81846	93.128	84.81846
EDGE_ANE Artificial R	200	3220	0.1586	528.55	0.026	20300	20300	0	0	0	0	0	7580	7580	12720	12720	0	0	100.6708	72.52	100.6708
EDGE_ANI Dwarf Mor	200	3220	0.1586	206.75	0.0102	20300	20300	0	0	0	0	0	10510	10510	9790	9790	0	0	74.36923	106.712	74.36923
EDGE_ANI Genetic Al	200	3220	0.1586	269.5875	0.0133	20300	20300	0	0	0	0	0	9930	9930	10370	10370	0	0	83.03385	95.448	83.03385
EDGE_ANI Optimized	200	3220	0.1586	157.9625	0.0078	20300	20300	0	0	0	0	0	10810	10810	9490	9490	0	0	73.63077	107.672	73.63077
EDGE_ANI Parallel Pa	200	3220	0.1586	321.6125	0.0158	20300	20298	0	2	0	0	C	8777	8777	11523	11521	0	0	86.59643	90.81664	86.59643

Figure 6: Combining all the results

#### 4.3.1 Analyzing the Results

For analyzing the results we have used a Python script.

- 1. Open the Python program 'Data Analysis' located in the root folder 'myPureEdgeSim'.
- 2. Change 'file\_url to 'file\_path' and assign it with the path/of/the/combined results.csv (Note: one can skip this step if they are interested in seeing the results of my data)
- 3. Run the program. It will analyse and print the results.

😂 Dat	a Analysis.ipyn	ь 🗙 😂 сору_					nb 🗧 copy_of 🕲 🖽 …
C: > N	CI > Research	in Computing > S					
+ Cod	le 🕂 Markd	own   Þ Run A	II 'O Restart 🗮 Clea	All Outputs   🔟 Variables 🛛 🗮 Ou	tline ····		🚊 Python 3.10.1
[6]	import import import ✓ 0.0s	pandas <i>as</i> p matplotlib. seaborn <i>as</i>	od pyplot <i>as</i> plt sns				Python
							떠 <u> </u>
▷ ∽	file_u # Read df = p print(	rl = " <u>https:</u> the data in d.read_csv(f df.head())	//drive.google.c ato a pandas Data file_url)	om/uc?id=15V5a2jEV1-lnAni Frame	tcUrJ4rf0dOdu8Hi	<u>יוו</u> "	
[7]							Python
	Orchest 0 1 2 3 4 Edge d	ration archi EDGE_AN EDGE_AN EDGE_AN EDGE_AN EDGE_AN EDGE_AN	tecture D_CLOUD Hybrid ( D_CLOUD D_CLOUD D_CLOUD D_CLOUD Total tasks exc	Orchestra irey Wolf - Whale Optimiz Honey Bi Sand Cat Swan Artificial Rabbin Dwarf Mongoose Optimiz ecution delay (s)	ation algorithm ation Algorithm adger Algorithm rm Optimization cs Optimization ation Algorithm		
	0	50.0		797.5 \			
	1	50.0		797.5			
	2	50.0		797.5			
) 42 🛆 14	43 🗸 AWS: A	WS Builder ID		707 5			

Figure 7: Analysis of all the results

## References

- Mechalikh, C., Taktak, H. and Moussa, F. (2019). Pureedgesim: A simulation toolkit for performance evaluation of cloud, fog, and pure edge computing environments, 2019 International Conference on High Performance Computing Simulation (HPCS), pp. 700– 707.
- Van Thieu, N. and Mirjalili, S. (2023). Mealpy: An open-source library for latest metaheuristic algorithms in python, *Journal of Systems Architecture*.