Applied Mathematics and Computational Intelligence
Volume 14, No. 4, 2025 [92 – 108]

Optimizing the SEIRD Model for COVID-19 in Malaysia Using Pymoo Framework

Muhammad Salihi Abdul Hadi[1*], Muhammad Aiman Haziqh Amran[2], Norsyahidah Zulkarnain[3]

[1,2,3]Department of Computational and Theoretical Sciences, Kulliyyah of Science, International Islamic University Malaysia, Kuantan Campus, 25100 Kuantan, Pahang, Malaysia

[*] Corresponding author: salihi@iium,edu.my

Received: 14 March 2025
Revised: 6 April 2025
Accepted: 9 May 2025

ABSTRACT

Recently, two specialized compartmental epidemiological models; the modified SIRD and SEIRD were developed to study COVID-19 in Malaysia, with their validity tested by fitting them to actual data. In this case, the optimization problem has a single objective: minimizing the least square error between the numerical solution and real-world data, without any imposed constraints. However, the introduction of time-dependent coefficients in both models increases the number of optimization variables. To solve this, the Nelder-Mead and Pattern Search algorithms were recommended. While Nelder-Mead is widely available in Python optimization libraries, Pattern Search is less common, which motivated the choice of the Pymoo multi-objective optimization framework for this study. The fitting results show that the absolute error metrics improve highest by about 40% compared to the results obtained from other optimization packages in previous studies. Furthermore, as an extension of prior research, we incorporate the computation of the dynamic basic reproduction number and its sensitivity analysis, confirming the effectiveness of the movement control order in controlling the disease during the pre-vaccination phase.

Keywords: Covid-19, Model Fitting, Epidemiology, Optimization.

1 INTRODUCTION

Python has emerged as one of the most preferred programming languages for academic studies and commercial projects in the current trending fields such as data science, machine learning, and deep learning. One of the factors is it has numerous different libraries and frameworks that can be used in those fields including the ones related to optimization such as Lmfit, SciPy and Pymoo. Python multi-objective optimization (Pymoo) framework is relatively new compared to the others and interestingly, it offers more algorithms even for single-objective optimization problems [1].

In this study, the optimization capability of Pymoo is tested by fitting a published epidemiological model of the outbreak of COVID-19 cases in Malaysia to the time series data from 25th January 2020 to 23rd February 2021 based on a modified SEIRD model. The data are collected from the Minister of Health Malaysia (MOH)'s official GitHub repository [2]. The modifications incorporated several crucial control measures such as lockdown, social distancing, quarantine, lockdown lifting time and the percentage of individuals who adhere to the prescribed regulations in terms of some dynamical kinetic functions [3].

Recently, this model and a similar modified SIRD model has been tested to fit the same COVID-19 data [4]. The SIRD model was fitted to COVID-19 data using a combination of Nelder-Mead and Pattern-Search algorithms from MATLAB's Optimization Toolbox. However, in our previous work Pattern-Search procedure was not implemented because it is not available from the Python's Lmfit library we used. Therefore, the SIRD model fitting had attained better accuracy, measured by the least Root Mean Square Error (RMSE), compared to ours [5]. This motivates the reproduction of SEIRD fitting using both Nelder-Mead and Pattern-Search under the Pymoo framework, to investigate whether Pymoo can match the performance of MATLAB's commercial optimization tools.

The impact of COVID-19 to Malaysian population was so severe as 2,346,303 individuals were infected including local and imported cases. The death toll stands at 27,422 cases, including those classified as Brought in Dead (BID), indicating individuals who passed away before reaching the hospital. These figures were reported by the Ministry of Health (MOH) Malaysia as of 11th October 2021. The Malaysian government, particularly the MOH Malaysia, has been closely involved in responding to this outbreak. Various phases of movement restrictions and Standard Operating Procedures (SOPs) have been implemented to break the chain of COVID-19 transmission among Malaysian citizens. These efforts have led to a significant decline in daily COVID-19 cases, effectively ending the second wave of the pandemic on 8th July 2020. However, the emergence of new variants of SARS-CoV-2 during the third wave has presented more critical and challenging circumstances.

The introduction of time dependent function for the evolution kinetics in the models is to consider the impact of intervention measures such as lockdown, healthcare system and the community behavior. These factors were translated into piecewise function based on Malaysian government action in three phases, first; before Movement Control Order (MCO), second; during MCO and Conditional MCO and lastly; during Recovery MCO. Expressing SIRD parameters in particular explicit function of time has practiced by various other researchers such as [6] and [7]. Our SEIRD model, despite heavily inspired by [8] and [9], the formulation for the evolution's kinetics is highly influenced by [4]. Note that the SEIRD model parameters in [8] and [9] are also considered as time varying but were implicitly determined by some different optimization techniques.

The basic reproduction number R_0 is one of the key values in epidemiology as it is conventionally used to predict whether the infectious disease will spread further or vanish. In most epidemiological contexts, the epidemic is controlled when $R_0 < 1$, however for complex epidemics, where $R_0(t)$ fluctuates, a proper dynamic threshold definition can be set as the upper bound. Despite the success in fitting their modified models to the data where the best optimized parameters were obtained, the estimation of R_0 is not reported in [4] and [10]. Usually, the explicit formula for R_0 can be derived from the model's ordinary differential equations (ODEs) and one of the popular methods is known as the next generation matrix. However, the R_0 formulation may not be unique as it depends on how one classifies the infected compartments from the ODEs during the derivation [11]. Once the R_0 explicit formula is derived then the sensitivity analysis can be carried out easily.

In this paper, first under the Methodology section, we present the SEIRD model that was proposed by [3] to investigate the dynamics of COVID-19 cases during the early outbreak period prior to the national vaccination program. Some justifications related to the construction of this compartmental model are also discussed. Then it is followed by the explanation on the solving the ODEs and optimization of the parameters to obtain the best fitted solution to the data. Next, the techniques used in the next generation matrix for deriving the $R_0(t)$ formula, its corresponding dynamic threshold and local sensitivity analysis will be presented. Secondly, under the Results and Discussion section, we demonstrate the improvement of our fitting results compared to the two main cited works. Additionally, the estimated $R_0(t)$ and its percentage degree of local sensitivity are also discussed. Finally, in the Conclusion part, we summarize our findings from this study.

2 MATERIAL AND METHODS

2.1 Model and Data

The modified SEIRD model is described in terms of non-linear ODE system as in Equation (1) – Equation (5). Additionally, Equation (6) represents the constant total population, denoted by N.

$$\frac{dS}{dt} = -\frac{\beta_I(t)SI}{N} - \frac{\beta_E(t)SE}{N} \quad (1)$$

$$\frac{dE}{dt} = \frac{\beta_I(t)SI}{N} + \frac{\beta_E(t)SE}{N} - \sigma E \quad (2)$$

$$\frac{dI}{dt} = \sigma E - \gamma(t)I - \mu(t)I \quad (3)$$

$$\frac{dR}{dt} = \gamma(t)I \quad (4)$$

$$\frac{dD}{dt} = \mu(t)I \quad (5)$$

$$N = S(t) + E(t) + I(t) + R(t) + D(t) \quad (6)$$

where S, E, I, R and D represent the susceptible, exposed, infected, recovered and death individuals in the population, $\beta_I(t)$ is the infection rate upon contact with I, $\beta_E(t)$ is the infection rate upon contact with E, σ is the incubation rate, $\gamma(t)$ is the recovery death and $\mu(t)$ is the death rate. The detail explanation on the model formulation can be referred to [3] and [5].

Note that the epidemiological parameters are changing over time. Inspired by [4], time-varying infection rate $\beta_I(t)$ and $\beta_E(t)$, recovery death $\gamma(t)$ and death rate $\mu(t)$ are formulated as piecewise-defined functions as in Equation (7) – Equation (10). The time interval of these piecewise functions is basically divided into three phases:

- Phase I: Before MCO, $t < t_{lock}$ (27/2/2020-17/3/2020)
- Phase II: During MCO, $t_{lock} \leq t < t_{lift}$ (18/3/2020-9/6/2020)
- Phase III: During RMCO, $t \geq t_{lift}$ (10/6/2020-23/2/21)

$$\beta_I(t) = \begin{cases} \beta_1 t + \beta_2, & t < t_{lock} \\ \beta_0 e^{-\frac{t-t_{lock}}{\tau_\beta}}, & t_{lock} \leq t < t_{lift} \\ (1-r)\left(\beta_1\left(t - t_{lift}\right) + \beta_2\right), & t \geq t_{lift} \end{cases} \tag{7}$$

$$\beta_E(t) = p\beta_I(t) \tag{8}$$

$$\gamma(t) = \begin{cases} \gamma_2(t) + \gamma_3, & t < t_{lock} \\ \gamma_0 + \dfrac{\gamma_1}{1 + e^{(-t + t_{lock} + \tau_\gamma)}}, & t \geq t_{lock} \end{cases} \tag{9}$$

$$\mu(t) = \begin{cases} \mu_2(t) + \mu_3, & t < t_{lock} \\ \mu_0 e^{-\frac{t-t_{lock}}{\tau_\mu}} + \mu_1, & t_{lock} \leq t < t_{lift} \\ \mu_2\left(t - t_{lift}\right) + \mu_3, & t \geq t_{lift} \end{cases} \tag{10}$$

Each phase is correlated to the specific time dependent function based on the population behavior. Phase I represents the situation before the MCO, where people were free to move without restriction, therefore the infection rates β_I and β_E, recovery rate γ and mortality rate μ are assumed as linear functions. Phase II considers the occurrence during the MCO, where the infection and mortality rates should decay due to the isolation and the implementation of standard operation procedures (SOP). Hence, those two rates could be described as exponential functions with decay rate. Meanwhile, in this phase the infected individuals should receive better treatment and healthcare, so it is expected that the recovery rate will increase following a logistic function. Phase III justifies the condition after MCO, when the lockdown was lifted but social activities were allowed under the SOP. Hence, again the infection and mortality rates can be considered as linear function, however the infection rate is also assumed to be proportional to the percentage of population disobedient to the SOP.

The original data consists in two spreadsheet files, namely Data Set 1 and 2. If combined, it spans from 25th January 2020 to 23rd February 2021, but we intentionally select the first 238 days of each dataset to fairly compare our fitting results with the previous works [4], [5] and [10]. Moreover, the chosen 238-day span sufficiently covers all three phases considered in the model. The dataset is clean and well-structured, and since our fitting methodology is based on a deterministic model rather than statistical inference, additional preprocessing steps such as smoothing or filtering are deemed unnecessary. Furthermore, the optimization procedure employs numerical derivative-free methods, which do not rely on gradient-based information. While these methods are less sensitive to data distribution assumptions, potential issues such as underreporting or inconsistencies are not explicitly modeled in this approach. However, given the nature of our dataset and problem context, we assume that such factors do not significantly impact the results.

2.2 The Optimization and Fitting Procedure

This problem has only one objective which is to minimize the difference between the numerical solutions from the model and the data i.e., the error. Hence, a single objective function needs to be constructed and the parameters that need to be optimized are corresponding coefficients from the system of ordinary differential SEIRD equations. Since the model is a system of non-linear ODEs, the objective function should include an ODE solver where the solver will require initial conditions for the five populations denoted by $S(0)$, $E(0)$, $I(0)$, $R(0)$ and $D(0)$. The optimization problem is defined as follows:

$$\min \sum_{i=1}^{m} \left(Y(t_i) - U(t_i, \mathbf{x})\right)^2, \quad \mathbf{x}_L \le \mathbf{x} \le \mathbf{x}_U \tag{11}$$

where t_i is the i-th day, Y is a case from the recorded data, U is the solution from the model for that particular Y and $\mathbf{x} \in \mathbb{R}^{\boldsymbol{n}}$ is the vector that contains the parameters for the optimization, $\boldsymbol{x}_L, \boldsymbol{x}_U$ are respectively the lower and upper bound of $\mathbf{x}$ and n is the number of those extremums. Based on the ODEs, $n = 16$ and we set the corresponding vectors as:

$$\mathbf{x} = [\beta_0, \beta_1, \beta_2, \gamma_0, \gamma_1, \gamma_2, \gamma_3, \mu_0, \mu_1, \mu_2, \mu_3, \tau_\beta, \tau_\gamma, \tau_\mu, p, \sigma],$$

$$\mathbf{x}_L = [\epsilon, \epsilon, \epsilon, \epsilon, \epsilon, \epsilon, \epsilon, \epsilon, \epsilon, \epsilon, \epsilon, 1,1,1,1,0.07] \text{ where } \epsilon = 10^{-6}, \tag{12}$$

$$\mathbf{x}_U = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,30,30,30,5,0.5].$$

The bounds corresponding to the parameters β_0 until μ_3 in Equation (12) are between $[0, 1]$ but we substitute 0 with $\epsilon = 10^{-6}$ merely for computational efficiency. The rational is that these parameters should be small since t will increase linearly in Equations (7 – 10). Meanwhile for characteristic time of transmission τ_β, characteristic time of recovery τ_γ and characteristic time of death τ_μ, the bounds are between [1,30] represent 30 days limit based on our previous study [5]. Parameter p, the proportion of β_E over β_I is bounded between $[1, 5]$ based on [9] and parameter σ, incubation rate is based on the general formula $1/T_c$ where T_c is the average of latency period which ranges from 5 to 6 days [5], therefore $[0.07, 0.5]$ is the assumed bound.

2.2.1 Nelder-Mead Method

The Nelder-Mead (NM) technique is a direct method or derivative-free minimization algorithm that generates a sequence of simplexes that run after or circumscribe the minimizer of the objective function. It uses the evaluations of the objective function at the simplexes' vertices and simple geometrical transformations such as reflections, expansions, and contractions repeatedly until it satisfies the stopping criteria for convergence and returns associate vertices as the solution. This algorithm is quite robust and efficient for small dimensional problems. Its rate of convergence is severely affected by the choice of the initial simplex [12].

2.2.2 Pattern-Search Method

The Hooke and Jeeves Pattern-Search (PS) algorithm is also a derivative-free optimization technique. It starts with an initial guess of the parameter space, then chooses a search pattern (a set of directions) and makes a move from the current point. After evaluating the objective function at the new point then the pattern is updated based on the objective function value either accept it if it decreases or backtrack to the previous point and reduce step size in the current direction if no improvement or find a new direction if no improvement at all. These steps are repeated until the convergence criteria are met [13].

2.3 The Optimization Application in Pymoo

Firstly, Pymoo requires the user to define the optimization problem in a class definition as commonly practice in object-oriented programming [14]. We define our optimization problem as follows:

```
from pymoo.core.problem import ElementwiseProblem
...
class MyProblem(ElementwiseProblem):
    def __init__(self, tspan, initz, cdata, rpars):
        xl=1.0e-6*np.ones(16)
        xl[11:16]=[1,1,1,1,0.07]
        xu=np.ones(16)
        xu[11:16]=[30,30,30,5,0.5]
        super().__init__(n_var=16,n_obj=1,xl=xl,xu=xu)
        self.tspan = tspan
        self.initz = initz
        self.cdata = cdata
        self.rpars = rpars

    def _evaluate(self, x, out, *args, **kwargs):
        tspan = self.tspan
        initz = self.initz
        cdata = self.cdata
        rpars = self.rpars
        sol = Ode_Solver(tspan,initz,*(rpars+list(x)))
        f1 = (sol[:, 2:5] - cdata).ravel()
        out["F"] = np.sum(f1**2)
```

Specifically, in this case, the optimization problem is defined by a custom class called `MyProblem`. It is a subclass of Pymoo's base class for single-objective optimization problems, `ElementwiseProblem`. The class constructor `__init__` is responsible for initializing the problem and determining the upper and lower bounds (`xl` and `xu` respectively) for the optimization variables, which are the compartmental model's parameters. There are 16 optimization variables, as indicated by `n_var=16`, `n_obj=1` represents that there is one objective function (single objective) and the `super().__init__` call initializes the base class of `ElementwiseProblem` with specific parameters. The second definition in the class is `_evaluate` function. This function is required by `ElementwiseProblem` to evaluate the objective function. It employs the user defined function `Ode_Solver` where the sixteen parameters are assigned to the array `x` as one of the inputs, to obtain the numerical solution for the ODEs. Next, it computes the objective function value by comparing the model prediction with the actual data and summing the squared differences. The output was then used by the optimization algorithm to guide the search for new optimized parameters.

Parts of our implementation of the defined class are as follows:

```
...
from pymoo.algorithms.soo.nonconvex.nelder import NelderMead
from pymoo.optimize import minimize
from pymoo.algorithms.soo.nonconvex.pattern import PatternSearch
...
def Ode_Model(z,t,tMCO,tRMCO,abdr,deltp,beta0,beta1,beta2,gam0,gam1,
```

```
            gam2,gam3,mu0,mu1,mu2,mu3,taub,taug,taum,prbb,sigmp):
    S, E, I, R, D = z
    N = S + E + I + R + D
    betaI = Beta_I(t,tMCO,tRMCO,beta0,beta1,beta2,taub,abdr)
    betaE = prbb*betaI
    gampt = Gampt(t,tMCO,gam0,gam1,gam2,gam3,taug)
    mupt = Mupt(t,tMCO,tRMCO,mu0,mu1,mu2,mu3,taum)
    dSdt = -(betaI*S*I)/N - (betaE*S*E)/N + deltp*R
    dEdt = (betaI*S*I)/N + (betaE*S*E)/N - sigmp*E
    dIdt = sigmp*E - gampt*I - mupt*I
    dRdt = gampt*I - deltp*R
    dDdt = mupt*I
    dydt = [dSdt, dEdt, dIdt, dRdt, dDdt]
    return dydt

def Ode_Solver(tspan,initz,*pars):
    OdeFun = lambda y,t: Ode_Model(y,t,*pars)
    sol = odeint(OdeFun,initz,tspan)
    return sol
...
problem = MyProblem(tspan,initz,cdata,rpars)
algorithm = NelderMead(x0=np.array())
res = minimize(problem,algorithm,seed=1,verbose=False)
...
algorithm = PatternSearch(x0=optimal_pNM)
res = minimize(problem,algorithm,seed=1,verbose=False)
...
```

The user defined function for the ODEs is `Ode_Model` and to solve the model we use `odeint` function from SciPy module in the `Ode_Solver` procedure. The initial values `initz` can be referred to Table 1. Then, we define the object from `MyProblem` class as `problem`, then one can choose the available optimization algorithms offered in Pymoo, in which `NelderMead` and `PatternSearch` in our case. Finally, `minimize` is called to perform the optimization based on the chosen algorithm. The initial values for the optimization `ipars` can be referred to Table 2.

Table 1 : Initial values for solving the ODEs of the SEIRD model.

Dataset	Initial States	Values (Individuals)	Sources
1	Susceptible, $S(0)$	32657300	Equation (6)
	Exposed, $E(0)$	4	Assumption
	Infected, $I(0)$	3	[2]
	Recovered, $R(0)$	0	[2]
	Death, $D(0)$	0	[2]
2	Susceptible, $S(0)$	32657300	Equation (6)
	Exposed, $E(0)$	100	Assumption
	Infected, $I(0)$	1	[2]
	Recovered, $R(0)$	22	[2]
	Death, $D(0)$	0	[2]

Table 2 : Initial parameters input for the optimization procedure.

Parameters	Values	Sources
Infection rate, β_I	$\beta_0 =$ 0.16100732 $\beta_1 =$ 0.00142347 $\beta_2 =$ 0.07373335	[4]
Recovery rate, γ	$\gamma_0 =$ 0.02590983 $\gamma_1 =$ 0.02670039 $\gamma_2 =$ 0.00008300 $\gamma_3 =$ 0.00606688	[4]
Death rate, μ	$\mu_0 =$ 0.00151062 $\mu_1 =$ 0.00015316 $\mu_2 =$ 0.00008013 $\mu_3 =$ 0.00025064	[4]
Characteristic time of transmission, τ_β	21.73215538	[4]
Characteristic time of recovery, τ_γ	12.35930060	[4]
Characteristic time of death, τ_μ	26.35932269	[4]
Proportion of β_E over β_I, p	1.002	[5]
Incubation rate, σ	0.15	[5]

The flow of our program basically follows what has been implemented in [4], the optimized parameters from Nelder-Mead is impoverished using Pattern-Search, and it is repeated until the current RMSE error is larger than the previous one. However, for each of the methods, Pymoo provides the default setting for terminating the procedure where the maximum number of iterations is 1000 and the numerical tolerance is 10^{-8}. This gives a disadvantage in the computational time, for example using a laptop with Intel-i5 2.38 GHz processor, the elapse time is at least 30 minutes.

2.4 Derivation of Reproduction Number Formula

In the context of compartmental epidemiological modeling, there are a few approaches to estimate the reproduction number such as by direct definition, Jacobian of the ODE and next generation matrix. For a relatively complex model, the next generation matrix approach is preferable [15]. The compartments are divided into two broad categories: infected compartments and noninfected (healthy) compartments. In this study, Equations (2) and (3) are categorized under infected compartments, while the healthy compartments comprise Equations (1), (4) and (5). The ODEs that are assumed under infected compartments can be rewritten as:

$$\frac{dE}{dt} = \mathcal{F}_1 - \mathcal{V}_1\,, \qquad \frac{dI}{dt} = \mathcal{F}_2 - \mathcal{V}_2 \tag{13}$$

where

$$\mathcal{F}_1 = \frac{\beta_I SI}{N} + \frac{\beta_E SE}{N}, \qquad \mathcal{V}_1 = \sigma E\,, \qquad \mathcal{F}_2 = 0, \qquad \mathcal{V}_2 = -\sigma E + \gamma I + \mu I. \tag{14}$$

Then we construct matrices

$$F = \begin{bmatrix} \frac{\partial \mathcal{F}_1}{\partial E} & \frac{\partial \mathcal{F}_1}{\partial I} \\ \frac{\partial \mathcal{F}_2}{\partial E} & \frac{\partial \mathcal{F}_2}{\partial I} \end{bmatrix} = \begin{bmatrix} \frac{\beta_E S}{N} & \frac{\beta_I S}{N} \\ 0 & 0 \end{bmatrix}, \qquad V = \begin{bmatrix} \frac{\partial \mathcal{V}_1}{\partial E} & \frac{\partial \mathcal{V}_1}{\partial I} \\ \frac{\partial \mathcal{V}_2}{\partial E} & \frac{\partial \mathcal{V}_2}{\partial I} \end{bmatrix} = \begin{bmatrix} \sigma & 0 \\ -\sigma & \gamma + \mu \end{bmatrix}. \tag{15}$$

The so-called, next generation matrix is given by

$$K = FV^{-1} = \begin{bmatrix} \frac{\beta_E S}{\sigma N} + \frac{\beta_I S}{(\gamma + \mu)N} & \frac{\beta_I S}{(\gamma + \mu)N} \\ 0 & 0 \end{bmatrix} \tag{16}$$

The effective reproduction number is

$$R_{eff} = \frac{\beta_E S}{\sigma N} + \frac{\beta_I S}{(\gamma + \mu)N} \tag{17}$$

which is the largest eigen-value of K. This number represents the average number of secondary infections produced by a single infected individual at time t in a population that may include both susceptible and non-susceptible (immune) individuals. It reflects the current state of the epidemic. Conventionally, to obtain the basic reproduction R_0 formula, one should replace the dependent variable with solutions from a disease-free equilibrium state and the averaged values for the parameter, in such R_0 is constant over time. This is qualitatively suitable for a study over short period of time span when the values are not too much deviate from their averages.

In our case, we substitute the initial condition in R_{eff} as suggested in [8], to obtain

$$R_0(t) = \frac{p\beta_I(t)S_0}{\sigma N} + \frac{\beta_I(t)S_0}{\big(\gamma(t) + \mu(t)\big)N} \tag{18}$$

where $S_0 = S(0)$ is the initial condition when the ODEs are solved but in [8], they fixed $p = 5$ and $\sigma = 1/7$ and neglect the first term without any explanation. Perhaps, the first term is insignificant in that case. In most epidemiological contexts, the epidemic is under controlled when $R_0(t) < 1$, meaning each infected person spreads the disease to fewer than one other person on average. Despite that, [8] proposed a dynamical threshold $L_0(t)$ to compare with $R_0(t)$, given by

$$L_0(t) = \frac{I(t)}{I(t) + pE(t)} \tag{19}$$

which represents the fraction of individuals who are currently infectious relative to the pool of both infectious and exposed individuals. Using $L_0(t)$ as the control threshold is more realistic for complex epidemics, where the reproduction number can fluctuate, and the exposed population plays a key role. In such cases, simply requiring $R_0(t) < 1$ might be too strict, the condition

$R_0(t) < L_0(t)$ allows for a more nuanced understanding; the epidemic could still be manageable if enough individuals are in the exposed phase (and not infectious yet).

2.5 The Sensitivity Analysis of the Basic Reproduction Number

It is important to know the change generated in a model variable when the values of one of its parameters is changed in the model. A high sensitivity index value for a parameter primarily indicates that the parameter is significant to the model, meaning that variations in this parameter have a substantial impact on the model's output. To compute the sensitivity analysis, we adopt the definition of the normalized sensitivity index from [16]. The sensitivity index of R_0 with respect to p, β_I, σ, γ and μ are given as follows:

$$\Upsilon_p(t) = p\big(\gamma(t) + \mu(t)\big)/\big(p\big(\gamma(t) + \mu(t)\big) + \sigma\big) \tag{20}$$

$$\Upsilon_{\beta_I}(t) = 1 \tag{21}$$

$$\Upsilon_\sigma(t) = -p\big(\gamma(t) + \mu(t)\big)/\big(p\big(\gamma(t) + \mu(t)\big) + \sigma\big) \tag{22}$$

$$\Upsilon_\gamma(t) = -\gamma(t)\sigma/\Big(\big(\gamma(t) + \mu(t)\big)\big(p\big(\gamma(t) + \mu(t)\big) + \sigma\big)\Big) \tag{23}$$

$$\Upsilon_\mu(t) = -\mu(t)\sigma/\Big(\big(\gamma(t) + \mu(t)\big)\big(p\big(\gamma(t) + \mu(t)\big) + \sigma\big)\Big). \tag{24}$$

Except for $\Upsilon_{\beta_I}(t)$ all other sensitivity index expressions are dependent on time. Clearly, as the parameters in our SEIRD model that are also time dependent.

3 RESULTS AND DISCUSSION

First, to compare the current work with [4] and [5], we fit it to the case of COVID-19 outbreaks in Malaysia recorded from 25th January until 13th May 2020. The comparisons in terms of RMSE are depicted in Table 3. The RMSE from [5] is recalculated to match the considered period since in [5], the calculation was carried out for a much wider time span i.e. until 18th September 2020. Our current model has the lowest RMSE compared to the other two. It is about 7.5% RMSE deterioration from Model 1 to Model 2 but 27.5% improvement from Model 1 to Model 3 and 32.6% improvement from Model 2 to Model 3.

Table 3 : Comparison of RMSE from various related models to the case of COVID-19 outbreak in Malaysia.

No	Model	Package / Methods	RMSE
1	SIRD [4]	MATLAB / NM and PS	80.08
2	SEIRD [5]	Lmfit / NM	87.56
3	SEIRD, [this study]	Pymoo / NM and PS	58.35

Next, we show the result of the fitting of the model to the data by choosing two data sets, first data set from 25th January 2020 until 18th September 2020 and second data set from 27th February 2020 until 21st October 2020. Note that the latter data set is just a prolong data of the former, and basically these data are from the pre-vaccination phase, collected from the MOH Malaysia's official GitHub repository. Figure 1 illustrates the fitting of the SEIRD model for three recorded cases from the first data set. In the death case, the fitting seems worse compared to the other two cases but one should look at the proportion of the y-axis to see that the discrepancies are of order as in the other cases. Similarly, for the second set of data, the fitting is depicted in Figure 2 but since it is the extended period of first data set, we use the optimized parameters from the first data set as the initials to compute the optimized parameters for the second data set. Initial values for solving the ODEs are also changed based on the starting point of the second data set.

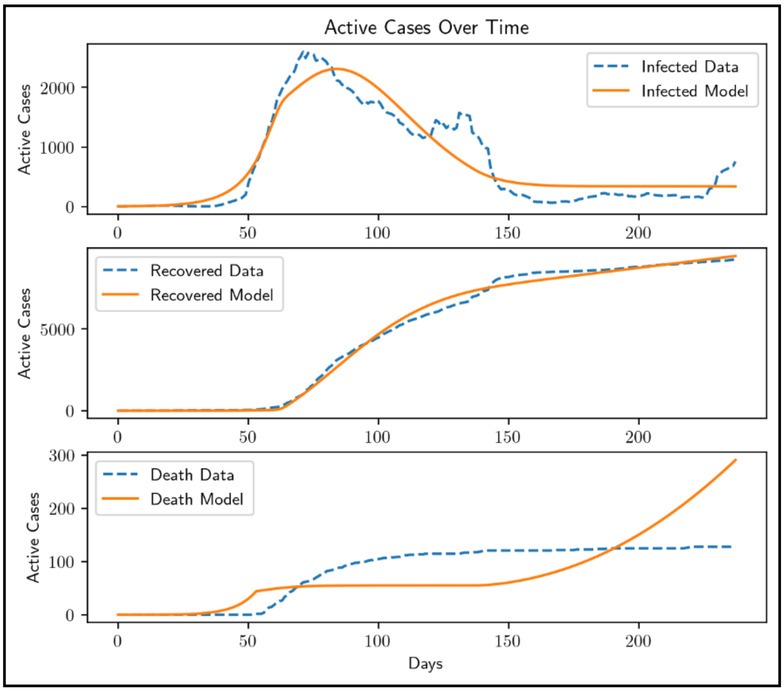

Figure 1 : Fitting to the Dataset 1 from 25th January 2020 to 18th September 2020

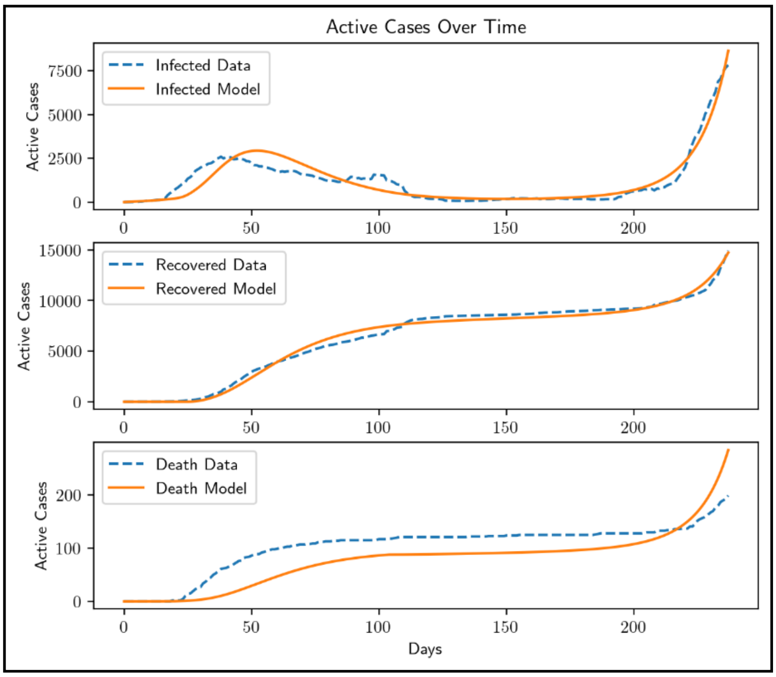

Figure 2 : Fitting to the Dataset 2from 27th February 2020 to 21st October 2020

Further comparisons of error metrics between Lmfit and PyMoo are presented in Table 4 (SEIRD model) and Table 5 (SIRD model). The SIRD model from [4] was reimplemented in Python, where we computed the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Symmetric Mean Absolute Percentage Error (sMAPE) to evaluate model performance. The comparison between the SIRD and SEIRD models shows that SEIRD generally improves absolute error metrics while SIRD sometimes has lower relative errors (sMAPE). For Dataset 1, RMSE decreases ranging by 2.4% to 4.9% when switching from SIRD to SEIRD, while MAE improves ranging by 5.8% to 9.5%, though it slightly increases in one case. In Dataset 2, SEIRD achieves a more significant reduction, lowering RMSE ranging by 4.8% to 22.5% and MAE ranging by 9.4% to 19.0%, demonstrating improved absolute accuracy. Comparing optimization methods, shifting from Lmfit to Pymoo (with NM and PS) improves RMSE by 11.0% to 15.2% and MAE by 7.8% to 8.0% in Dataset 1, while in Dataset 2, Pymoo significantly enhances RMSE by 39.7% and MAE by 42.0%, making it the superior approach for fitting COVID-19 data. However, sMAPE remains consistently high (>20%) for all cases, indicating poor relative accuracy, likely due to the sensitivity of sMAPE to small values or data inconsistencies.

Table 4 : Comparison of error metrics for the Python optimization modules in fitting the SEIRD model to the case of COVID-19 outbreak in Malaysia.

Dataset	Modules / Methods	MAE	RMSE	sMAPE%
1	Lmfit / NM	166.54	265.19	49.37
	Pymoo / NM	153.61	236.12	58.80
	Pymoo / NM and PS	153.27	224.95	66.15
2	Lmfit / NM	412.75	589.24	62.14
	Pymoo / NM	401.21	648.04	59.52
	Pymoo / NM and PS	239.40	355.20	38.11

Table 5 : Comparison of error metrics for the Python optimization modules in fitting the SIRD model to the case of COVID-19 outbreak in Malaysia.

Dataset	Modules / Methods	MAE	RMSE	sMAPE%
1	Lmfit / NM	183.98	278.98	55.02
	Pymoo / NM	163.11	241.84	60.40
	Pymoo / NM and PS	144.37	234.33	53.25
2	Lmfit / NM	391.02	619.23	43.97
	Pymoo / NM	442.88	801.52	47.97
	Pymoo / NM and PS	295.61	458.33	44.12

Further, we compute the basic reproduction number $R_0(t)$ and the dynamical threshold $L_0(t)$ for both data sets. In the first data set, the MCO began on the 53rd day, RMCO started on 137th day and based on Figure 3, the range where $R_0(t) < L_0(t)$ is $114 \le$ days ≤ 142. This means that it took about 61 days after MCO for the COVID 19 to be under control and in just 5 days after RMCO, the disease outbroke again. While in the second data set, the MCO started on the 20th day, RMCO initiated on 104th day and based on Figure 4, the range where $R_0(t) < L_0(t)$ is $87 \le$ days ≤ 112. This gives 67 days after MCO to effectively control the disease and the spreading restarted after 8 days MCO had been lifted. Overall, these figures are qualitatively in agreement to support that the MCO had succeeded in preventing the spread of COVID 19 two months after its implementation and the spread had continued just after one week it was being lifted.

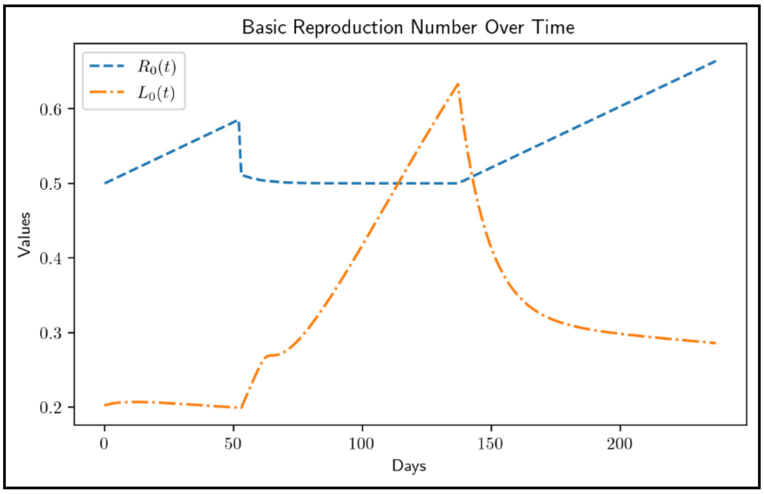

Figure 3 : R_0 and L_0 from 25th January 2020 to 18th September 2020

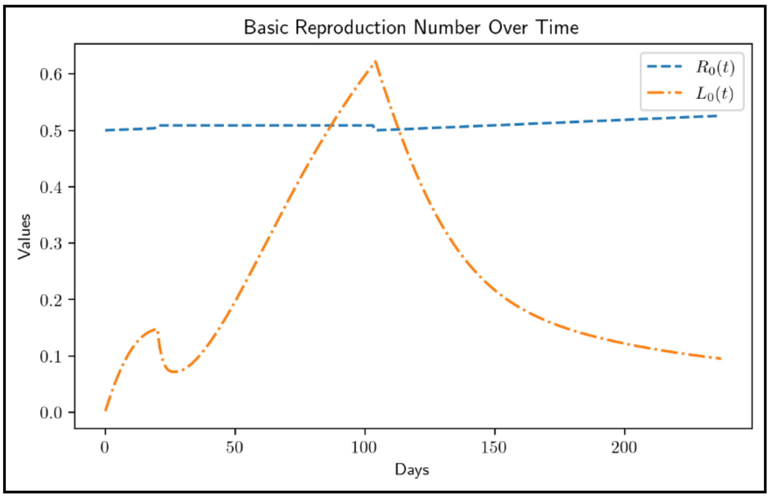

Figure 4 : R_0 and L_0 from 27th February 2020 to 21st October 2020

Finally, the local sensitivity analysis of $R_0(t)$ with respect to its parameters is presented for both datasets. Figures 5 and 6 illustrate the sensitivity index of $R_0(t)$ concerning p, β_I, σ, γ and μ for the first and second datasets, respectively. Notably, only $\Upsilon_{\beta_I}(t)$ remains constant across both time spans. Comparing the maximum absolute values of the sensitivity indices, $\Upsilon_{\beta_I} = 1$ is the highest, indicating that β_I (infection rate) is the most influential parameter in estimating $R_0(t)$. A 1% increase (or decrease) in β_I directly results in a 1% increase (or decrease) in $R_0(t)$. Moreover, examining the maximum absolute sensitivity across both datasets in Figure 5, a 1% change in p leads to a maximum 0.5% change in $R_0(t)$, whereas a 1% increase (or decrease) in σ, γ or μ results in a maximum 0.5% decrease (or increase) in $R_0(t)$.

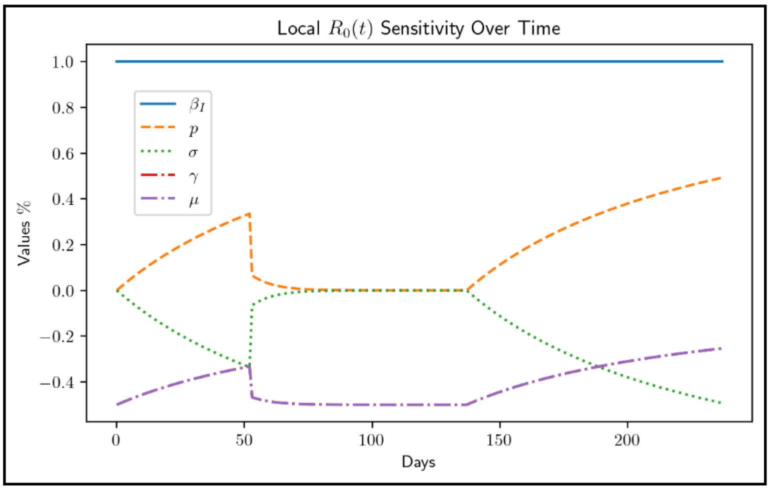

Figure 5 : The sensitivity index of R_0 with respect to p, β_I, σ, γ and μ from 25th January 2020 to 18th September 2020

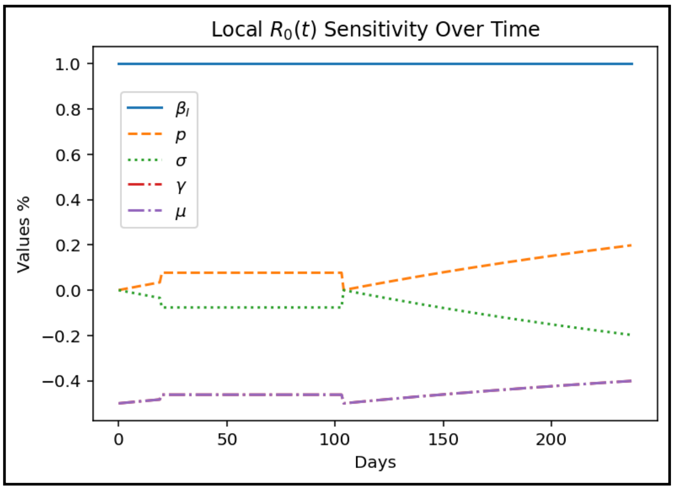

Figure 6 : The sensitivity index of R_0 with respect to p, β_I, σ, γ and μ from 27th February 2020 to 21st October 2020

4 CONCLUSION

In this study, the SEIRD model generally outperforms SIRD in terms of absolute error, as indicated by lower RMSE and MAE. While SIRD sometimes yields a lower sMAPE, suggesting better handling of relative errors, SEIRD remains the more suitable model for fitting COVID-19 data, as absolute errors are more critical than percentage-based errors in this context. The integration of the Pymoo framework to implement a combination of Nelder-Mead and Pattern Search has significantly improved the SEIRD model's fit to COVID-19 case data, achieving up highest about a 40% reduction in both RMSE and MAE. Furthermore, the calculation of R_0 and L_0, along with their dynamic relationship, aligns well with MCO outcomes, indicating that the model parameters have been accurately estimated through the optimization procedure. The sensitivity analysis of R_0 with respect to its corresponding parameters reveals that all sensitivity indices remain below 1%, with the infection rate β_I being the most influential factor.

The SEIRD model is explicitly constructed for the pre-vaccination period, specifically from January 25, 2020, to February 23, 2021. It does not account for the phase during which Malaysia's vaccination program was implemented. Consequently, the fitted model cannot explain the current phase of COVID-19, where the national immunization program has been completed. Future research could extend this model to incorporate the post-vaccination period. In doing so, it would be crucial to account for vaccination-related factors when developing an updated compartmental model. By integrating vaccination variables and parameters, a more comprehensive understanding of COVID-19 transmission during and after the vaccination phase could be achieved. This would provide valuable insights into the effectiveness of vaccination efforts and aid in designing more effective pandemic control strategies.

ACKNOWLEDGEMENT

The third author acknowledges the Ministry of Higher Education Malaysia (MOHE) for the financial support. This work is an extension from a research project, entitled "Systemic Approach on Book of Life: Methods, Theories and Implications" (Reference Code: FRGS/1/2018/SSI04/ UIAM/02/1).

REFERENCES

[1] J. Blank and K. Deb, "Pymoo: Multi-objective optimization in Python," *IEEE Access*, vol. 8, pp. 89497-89509, 2020. [Online]. Available: https://doi.org/10.1109/ACCESS.2020.2990567

[2] COVID-19 Database in Github MOH [Online]. Available: https://github.com/MoH-Malaysia/covid19public/blob/main/epidemic/cases_malaysia.csv

[3] N. Zulkarnain, N. F. Mohammad and I. Shogar, "Modified SEIRD model: A novel systems dynamic approach in modelling the spread of COVID-19 in Malaysia during the vaccination period," *IIUM Engineering Journal*, vol. 24, no. 2, 2023. [Online]. Available: https://journals.iium.edu.my/ejournal/index.php/iiumej/article/view/2550

[4] N. M. Jamil, N. Rosli and N. Muhammad, "Simulation of COVID-19 outbreaks via graphical user interface (GUI)," *Journal of Public Health Research*, 2022. [Online]. Available: https://journals.sagepub.com/doi/10.4081/jphr.2021.2130

[5] N. Zulkarnain, M. S. A. Hadi, N. F. Mohammad and I. Shogar, "Fitting time-varying coefficients SEIRD model to COVID-19 cases in Malaysia," International Journal of Innovative Computing, vol. 13, no. 1, June 2023. [Online]. Available: https://ijic.utm.my/index.php/ijic/article/view/397

[6] D. Caccavo, "Chinese and Italian COVID-19 outbreaks can be correctly described by a modified SIRD model," *medRxiv*, 2020. [Online]. Available: https://www.medrxiv.org/content/10.1101/2020.03.19.20039388v3

[7] D. Fanelli and F. Piazza, "Analysis and forecast of COVID-19 spreading in China, Italy and France," *Chaos, Solitons & Fractals*, vol. 134, 2020. [Online]. Available: https://doi.org/10.1016/j.chaos.2020.109761

[8] J. He, G. Chen, Y. Jiang, R. Jin, A. Shortridge, S. Agusti, M. He, J. Wu, C. M. Duarte and G. Christacos, "Comparative infection modeling and control of COVID-19 transmission patterns in China, South Korea, Italy and Iran," *Science of The Total Environment*, vol. 747, 2020. [Online]. Available: https://doi.org/10.1016/j.scitotenv.2020.141447

[9] A. M. M. Mukaddes and M. Sannyal, "A simulation approach to predict the transmission of COVID-19 in Bangladesh," *SSRN*, May 2020. [Online]. Available: http://dx.doi.org/10.2139/ssrn.3598172

[10] N. Zulkarnain, M. S. A. Hadi, N. F. Mohammad and I. Shogar, "Modelling transmission dynamics of COVID-19 during pre-vaccination period in Malaysia: A GUI-Based SEIRD predictive model using Streamlit," *Journal of Quality Measurement and Analysis*, vol. 20, is. 1, 2024. [Online]. Available: https://doi.org/10.17576/jqma.2001.2024.03

[11] F. Brauer, C. Castillo-Chavez and Z. Feng, "Mathematical models in epidemiology," New York, NY, USA: Springer-Verlag, 2019.

[12] C. Audet and W. Hare, "Derivative-free and blackbox optimization," Switzerland: Springer Nature, 2017.

[13] M. S. Bazaraa, H. D. Sherali and C. M. Shetty, "Nonlinear Programming: Theory and Algorithms," 3rd ed. Hoboken, NJ, USA: Wiley, 2006.

[14] J. Blank, “Pymoo – Definition,” Pymoo Framework Version 0.6.1.3, 2020. [Online]. Available: https://pymoo.org/problems/definition.html

[15] M. Martcheva, “An Introduction to Mathematical Epidemiology,” New York, NY, USA: Springer Science+Business Media, 2015.

[16] H. Alhamami, “A susceptible-exposed-infected-recovered- vaccinated mathematical model of measles in Madagascar.” MSc Dissertation , Morgan State University, Baltimore, MD, USA, 2019.