STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–22.
Published online in International Academic Press (www.IAPress.org)

Advanced Parameter Estimation for the Gompertz-Makeham Process: A Comparative Study of MMLE, PSO, CS, and Bayesian Methods

Adel S. Hussain [1], Muthanna Subhi Sulaiman [2], Sura Mohamed Hussein [2], Emad A. Az-Zo'bi [3], Mohammad A. Tashtoush [4,5,*]

[1] *IT Department, Amedi Technical Institutes, University of Duhok Polytechnic, Duhok, Iraq*
[2] *Department Statistics and Informatics, College of Computer and Mathematics, University of Mosul, Mosul, Iraq*
[3] *Department of Mathematics and Statistics, Faculty Sciences, Mutah University, Karak 61710, Jordan*
[4] *Depatrment of Basic Sciences, AL-Huson University College, AL-Balqa Applied University, Salt 19117, Jordan*
[5] *Faculty of Education and Arts, Sohar University, Sohar 311, Oman*

Abstract A research study investigates how to estimate Gompertz-Make ham Process (GMP) parameters within non-homogeneous Poisson processes (NHPP). Authorities have developed Modified Maximum Likelihood Estimation (MMLE) as an improvement over standard Maximum Likelihood Estimation (MLE) to resolve parameter estimation accuracy issues. The study utilizes combination artificial intelligence optimizations through particle swarm optimization (PSO) and cockoo search (CS) alongside Bayesian estimation to assess different methods. This study evaluates MMLE and PSO and CS with Bayesian methods through Root Mean Square Error (RMSE) and Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) statistical accuracy measurements during a simulation analysis. The MMLE estimation technique delivers better estimation precision than PSO, CS and Bayesian methods during the performance assessment. The methodology is validated through its use in modeling operational failures at the Badoush Cement Factory and COVID-19 case occurrences in Italy, showing its capability to model failure rates alongside event occurrences. The research generates progress in NHPP statistical estimation methods which gives a stronger analytical platform for reliability monitoring and survival model prediction and epidemiological projection. Research into the GMP needs to focus on including time-dependent elements and structural dependency mechanisms to enhance the model's capability and guess making power.

Keywords Gompertzian-Make ham Process; PSO Algorithm; Cuckoo Search (CS); Bayesian Estimation; Modified Maximum Likelihood Estimation; Simulation; Non-homogeneous Poisson Processes

AMS 2010 subject classifications 60J80, 60J85, 60K10

DOI: 10.19139/soic-2310-5070-2167

1. Introduction

The NHPP is a stochastic process that models the occurrence of events over time, where the rate of occurrence of events is not constant but varies with time. In other words, the NHPP is a type of Poisson Process in which the rate of occurrence of events is not constant but is a function of time. This process is characterized by a time-varying intensity function, which describes the rate of occurrence of events at any given time. The intensity function can be modelled using various parametric or non-parametric models, such as exponential, Weibull, or log-normal models [1]. The NHPP has several important applications in various fields, including reliability engineering, finance, and telecommunications. In reliability engineering, the NHPP is commonly used to model the failure rate of systems over time, where the failure rate varies with time due to aging or other factors [2]. In finance,

*Correspondence to: Mohammad A. Tashtoush (Email: tashtoushzz@su.edu.om). Depatrment of Basic Sciences, AL-Huson University College, AL-Balqa Applied University, Salt 19117, Jordan, and Faculty of Education and Arts, Sohar University, Sohar 311, Oman.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

the NHPP is used to model the arrival rate of financial transactions or the occurrence of market events over time [3].

In telecommunications, the NHPP is used to model the arrival rate of calls or messages in communication networks [4, 5]. One important property of the NHPP is that it satisfies the Markov property, which means that the future occurrence of events depends only on the current [6]. This property makes the NHPP particularly useful for modelling complex systems and processes. Although, the time-varying nature of the NHPP and the complexity of the intensity function make the parameter estimation for the intensity function challenging, but various methods for estimating such parameters have been proposed; these includes Maximum Likelihood Estimation, Bayesian Inference, and Intelligent techniques such as Genetic Algorithms (GA) and Neural Networks [7, 8].

In continuation of the reference review of some previous research and studies the NHPP continues to be a vital tool in various fields, with ongoing research enhancing its applicability and accuracy. In this study, we proposing an NHPP model characterized by the Gompertz-Make ham distribution [9, 10]. By comparing different classical and intelligent estimation methods, we aim to identify the most accurate approach for parameter estimation in this context. Our results are validated against real-world data, providing practical insights into the effectiveness of these methods [11].

2. Gompertz-Make ham Process (GMP)

The number of events that occur within a time interval $(0, t]$ follows a Poisson distribution with a probability density function assuming that the Poisson process $\{X(t), t = 0\}$[12]:

$$p[N(t) = n] = \frac{(\lambda\,(t))^{n} e^{-m(t_0)}}{n!}, \;\; n = 1, 2, 3. \tag{1}$$

The cumulative density function for the time rate of occurrence of events $m(t)$ can be found by integrating the intensity function as follows:

$$m(t) = \int_0^t \lambda\,(u)\,du, \;\; 0 < t < \infty, \tag{2}$$

where $\lambda(u)$ denotes the intensity function or time rate of occurrence. Following is a description of the Gompertz-Make ham process, which appears as a nonhomogeneous Poisson process with the time rate of occurrence:

$$\lambda\,(t) = a + b\,e^{c\,t},\; t \geq 0,\; a, b, c, d, > 0, \tag{3}$$

The parameters a, b, and c govern the Gompertzian-Make ham Process's time rate of event occurrences. A wide range of strategies have been presented as a consequence of extensive research on the estimate of parameters for processes of this kind.

3. Performance of Estimation Accuracy

When different estimates for a parameter are obtained, the comparison of their accuracy is an essential process. The measure the accuracy of these estimates, various techniques exist in literature; the root mean squared error (RMSE) is one of the most popular tools for measuring the accuracy. RMSE determines the differences between the estimated and the actual parameter values; it is defined as the square root of the average squared difference between the estimated and actual parameter values [13, 14].

The Root Mean Squared Error (RMSE) is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{Q} \left(\widehat{y}_i - y\right)^2}{Q}}. \quad (4)$$

Where
$\overline{y_i}$: Reflects the parameters predicted value for iteration i.
y: Reflects the actual value of the parameter.
Q: The total number of iterations.

4. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based and nature-inspired stochastic optimization technique utilized for solving diverse computational optimization problems. This method's inception stems from the concept of fostering social information exchange among individuals within a population. The crux of PSO lies in its elegant simplicity coupled with its formidable algorithmic strength a resilient search algorithm drawing inspiration from the intelligent and social conduct exhibited by various living organisms, ranging from swarms of insects like bees, wasps, and ants to collective assemblies observed in the natural world, such as flocks of birds or schools of fish. The algorithm was developed by simulating the mechanism of the natural phenomenon that depends on the movement and intelligence of individuals in swarms. In addition, the algorithm mimics the social behavior of animals, where individuals in a swarm cooperate to adapt to foraging, and each member adjusts its foraging pattern based on its own and other members' experiences. [15] invented PSO in 2015. Their goal was to create a model that might explain how animals, including fish and birds in flocks, behave in social situations. They created the idea of particle swarm optimization and put forth a brand-new optimizer known as PSO after realizing that their model might be used to optimization problems. Since the invention of this method, scientists have created new iterations to accommodate various needs, released theoretical analyses of the impacts of various factors, and unveiled several algorithmic variations.
Consequently, the method has gained popularity and proven efficacy in resolving optimization issues across several scientific and industrial domains. In PSO, a dynamic swarm, which is similar to a population, is composed of distinct particles that function as individual agents. These particles engage in communication, be it through direct or indirect means, utilizing search directions commonly referred to as gradients to navigate the optimization landscape effectively. In PSO, when a particle identifies a promising trajectory—such as a favorable path towards a food source—other particles within the swarm rapidly adopt this advantageous course, regardless of their initial distance from the group or cluster. This intrinsic cooperation mechanism results in swift convergence towards promising solutions. Hence, each individual particle in PSO is characterized by a triad of vector components [15, 16]:
1. The X-vector denotes the ongoing position (location) of the particle within the search space.
2. The V-vector encapsulates the gradient (direction) the particle intends to traverse.
3. The P-vector (P-best) records the coordinates of the particle's best-known solution thus far.

Therefore, each particle has a velocity vector, a position vector and a p-best that represents its best possible solution. The PSO algorithm is initialized with a series of random particles that search for the optimal value by updating generations (iterations). In each iteration, each particle is updated on two best values, one local and one global. The fitness function (objective function) used for optimization is then defined and a fitness value is determined for each particle [17]. Using the current velocity and the distance from P_best to g_best as follows:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 \left(P_{best} - X_i^t\right) + c_2 r_2 \left(g_{best} - X_i^t\right), \quad (5)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1}, \quad (6)$$

with $X_i^0 = U(X_{Min}, X_{Max})$
i.e. $X_i^0 = X_{Min} + r_i(X_{Max} - X_{Min}), r_i = U(0,1)$.

Apart from the initial random positioning of particles, their velocities can also be set to zero initially, denoted as $V_i^0 = 0$. However, the parameter ω, a positive constant denoted by the inertia weight, is determined. Takes on a pivotal role in steering local and global search dynamics. Accompanying this, the acceleration coefficients or learning factors, c_1 and c_2 play a crucial role in finetuning each iteration. They govern the extent to which a particle can traverse in a single step c_1 underscores an individual particle's understanding, prompting it to converge toward its own best-known position. On the other hand, c_2, the social or cooperative component, signifies the collective wisdom of the swarm, propelling particles toward a global solution. Choose these variables c_1 and c_2, and ω carries substantial weight in influencing the PSO algorithm's optimization performance. Additionally,r_1 and r_2 denote random numbers drawn from a uniform distribution $(0, 1)$, contributing to the diversity and exploration of the swarm's movement.

Hence, the fundamental steps encapsulating the PSO algorithm can be succinctly summarized as follows [18]:
1. Initialization of Positions: Initialize the positions of each particle by attributing them to random values.
2. Fitness level evaluation: Evaluate the fitness function of each particle individually.
3. Local Best Update: Update the local best if the newly encountered solution surpasses the previous one.
4. Global Best Update: Update the global best if the newly found solution is better than the previously recorded global best.
5. Velocity Calculation: Calculate the particle velocity employing Equation 5.
6. Position Update: Update the particle's position using Equation 6.
7. Iteration Loop: Repeat the sequence of steps (2) through (6) iteratively until the predefined termination criteria are met.

$$V_j^{(i)} = \theta\, V_j^{(i-1)} + C_1 r_1 \left(P_{best,j} - X_j^{(i-1)}\right) + C_1 r_1 \left(G_{best,j} - X_j^{(i-1)}\right),\, j = 1, 2, \ldots, N. \tag{7}$$

5. Cuckoo Search Algorithms (CS)

In this section, a new bio-inspired optimization algorithm, namely Cuckoo Search (CS) algorithm is proposed in (2015) by the authors [19]. It mimics the hierarchal order in the Cuckoo search and the Behavior of the Cuckoo swarm. The Cuckoo Search algorithm is particularly effective for solving complex optimization problems due to its simplicity and efficiency. The algorithm operates based on key principles derived from the natural behaviors of cuckoos.

While optimization can be defined as a branch of knowledge, dealing with the discovery or investigation of optimal solutions to a particular problem within a set of alternatives, or it can be considered one of the key quantitative tools in a decision-making network where decisions must be made to optimize one or more objectives in a specific set of Circumstances. The cuckoo i, a Levy flight is performed [19]:

$$x_i^{(i+1)} = x_i^{(i)} + \beta * Levy\,(\lambda)\,, \tag{8}$$

where $\beta > 0$ is the step size which should be related to the scales. Cuckoo Search has been successfully applied to a wide range of optimization problems, including function optimization, parameter estimation, feature selection, and machine learning model tuning. Its simplicity, effectiveness, and ability to handle multimodal and non-convex optimization problems make it a popular choice for optimization tasks.

Nesting and Replacement: In the Cuckoo Search algorithm, each cuckoo lays its eggs in the nests of other birds. The nests represent potential solutions to the optimization problem. If a host bird discovers an egg that is not its own (i.e., a poor solution), it may abandon that nest, allowing the cuckoo to take over. This process introduces a mechanism for replacing less optimal solutions with better ones.

Selection of Best Solutions: The algorithm iteratively evaluates the quality of the nests (solutions) based on a predefined fitness function. The best solutions are retained, while poorer solutions are replaced, leading to an overall improvement in the search for optimal parameters.

6. Parameters Estimation

The parameters of this process receive estimates through Modified Maximum Likelihood Estimation (MMLE) and through three methods which are Bayesian estimation and the PSO algorithm and CS algorithm.

6.1. Modified Maximum Likelihood Estimation (MMLE)

A typical statistical method for estimating parameters within a probability distribution using a given dataset is the Maximum Likelihood Estimator (MLE). Its goal is to find parameter values that maximize the likelihood function, which is a gauge of the likelihood of getting the observed data under various parameter configurations. However, in certain situations, MLE may yield inaccurate parameter estimates, particularly when the sample size is small or the distribution is complex. To address this limitation, modifications to the MLE approach can be employed to enhance the accuracy of the estimates [20, 21]. Modified Maximum Likelihood Estimator (MMLE) is one such variant of MLE that improves the precision of parameter estimates by incorporating additional information or constraints into the likelihood function. By modifying the likelihood function, MMLE takes advantage of the supplementary data or constraints available, leading to more refined estimates. This adjustment in the likelihood function can be beneficial in situations where the standard MLE may produce biased or imprecise results [22, 23]. The use of MMLE allows researchers to leverage relevant knowledge, prior information, or restrictions on the parameter space to refine the estimation process. By incorporating these supplementary factors, MMLE can overcome the limitations of the standard MLE and produce more accurate estimates, particularly in cases with limited data or complex distributions. In summary, the Modified Maximum Likelihood Estimator (MMLE) is a powerful technique that enhances the accuracy of parameter estimates by incorporating additional information or constraints into the likelihood function [23, 24]. This modification results in an improved likelihood function, leading to more precise parameter estimates, especially in situations where the standard MLE may yield less accurate results. Suppose $\{X(t), t = 0\}$ represents a nonhomogeneous Poisson Process with the time rate of occurrences determined by the Equation 3. In this scenario, the collective probability function for the sequence of event occurrences (t_1,t_2,...,t_n) within the interval $(0 \leq t_1 \leq t_2 \leq ... \leq t_n \leq t_0)$ is expressed as follows [23]:

$$f_n\left(t_1, t_2, \ldots, t_n\right)=\prod_{i=1}^n \lambda\left(t_i\right) e^{-m\left(t_0\right)}. \tag{9}$$

Therefore, be the formula for the cumulative function of the time rate of occurrence, a key variable in the Gompertz-Make ham process, is as follows:

$$m\left(t\right)=m(t)=\int_0^t \lambda\left(u\right) du=\int_0^t a+b\, e^{c\,u} du=a\,t+\frac{a}{b}\left(e^{c\,t}-1\right). \tag{10}$$

Thus, be the Likelihood function for the Gompertz-Make ham process over the period $(0, t]$ with the rate time $\lambda(t)$ is:

$$L=\prod_{i=1}^n\left(a+b\, e^{c\, t_i}\right) e^{a\, t_0+\frac{a}{b}\left(e^{c\, t_0}-1\right)}. \tag{11}$$

The maximum likelihood estimator for a,b,c. It can be calculated using the Equation 11, where:

$$\ln L = \sum \ln \left(a e^{b\, t_i} + c \right) - n\, c\, t - \frac{n\, a}{b} \left(e^{b\, t_0} - 1 \right). \tag{12}$$

The process of finding the logarithm derivative of the maximum likelihood function with respect to the parameter a can be outlined as follows:

$$\left. \frac{\partial \ln L}{\partial a} \right|_{a=\widehat{a}} = 0, \widehat{a} = \frac{1}{t_0 + \frac{1}{b} \left(e^{c\, t_0} - 1 \right)}. \tag{13}$$

To calculate the parameter b, we differentiate Equation 12 with respect to and set it equal to zero, so we get:

$$\widehat{b} = \widehat{a} \left(1 - e^{c\, t_0} \right). \tag{14}$$

To calculate the parameter c, we differentiate Equation 12 with respect to and set it equal to zero, so we get:

$$\frac{\partial \ln L}{\partial c} = \sum t_i - \frac{n\, a\, t_0}{b} e^{\widehat{c}\, t_0}, \; \widehat{c} = \ln \frac{b}{n\, a\, t_0} + \sum \ln t_i. \tag{15}$$

These equations can be solved numerically using iterative methods such as the Newton-Raphson algorithm or the EM algorithm to obtain the estimates for a, b, and c that maximize the likelihood function [18, 19]. We noticed that solving the system of equations obtained from taking the derivatives of Equation 12 with respect to a, b, and c is not feasible using traditional methods due to the high degree of nonlinearity. Therefore, we propose a modified maximum likelihood method by incorporating one of the most significant artificial intelligence methods (PSO).

• Modified Maximum Likelihood Estimator with PSO (MMLE-PSO)

The MMLE-PSO algorithm combines the MMLE approach with the PSO algorithm to evaluate Gompertz-Make ham process parameters. The MMLE incorporates additional information or constraints into the likelihood function to improve the accuracy of parameter estimation. By integrating the PSO algorithm, which is inspired by social behavior, the algorithm iteratively searches the parameter space to find the optimal parameter values that maximize the likelihood function.

Algorithm (1): MMLE (MLE-PSO)

1. Function Likelihood Function (data, a, b, c)
* Function $LogL$ (data, a, b, c)
* Return log (Likelihood Function (data, a, b, c))

2. Function $LogL$ (data, a, b, c)
* Compute $\partial logL/\partial a$
* Compute $\partial logL/\partial b$
* Compute $\partial logL/\partial c$
* Return system of equations

3. Function $MMLE - PSO$ (data)
* Initialize the PSO
* Population size (N = 50),
* Maximum number of iterations ($i\ max = 100$),
* Acceleration coefficients (C_1 = C_2 = 1)
* Random values (r_1 = r_2 = 0.1).

* Set the minimum ($\theta_m in$ = 0.4) and maximum ($\theta_m ax$= 0.9) values for the inertial weight.

4. For each particle i in $\{1, ..., N\}$
* Initialize random position (a_i,b_i,c_i)
* Initialize random velocity ($v_a i$,$v_b i$,$v_c i$)
* Evaluate fitness $= -LogL$ (data, a_i,b_i,c_i)
* Store personal best position and fitness

5. Set global best position and fitness from the best particle.

6. For iteration = 1 to $i\ max$, for each particle i in $\{1, ..., N\}$

7. Update velocity
* $v_a i$ = θ * $v_a i$ + C_1 * r_1 * (personal best a - a_i) + C_2 * r_2 * (global best a - a_i)
* $v_b i$ = θ * $v_b i$ + C_1 * r_1 * (personal best b - b_i) + C_2 * r_2 * (global best b - b_i)
* $v_c i$ = θ * $v_c i$ + C_1 * r_1 * (personal best c - c_i) + C_2 * r_2 * (global best c - c_i)

8. Update position
* a_i = a_i + $v_a i$
* b_i = b_i + $v_b i$
* c_i = c_i + $v_c i$

9. Evaluate new fitness $= -LogL$ (data, a_i,b_i,c_i)

10. Update personal best
* If fitness ¡ personal best fitness
* Update personal best position (a_i,b_i,c_i,)
* Update personal best fitness

11. Check stopping criterion
* If stopping condition met (max iterations or convergence threshold): Break
* Return global best position (a, b, c) as optimal parameters

6.2. PSO Algorithm for Parameter Estimation

The second algorithm directly applies the PSO algorithm for estimating the parameters of the Gompertz-Make ham process. The PSO algorithm, based on swarm intelligence, allows particles to explore the parameter space and find the parameter values that optimize a fitness function. In this case, the fitness function is defined based on the likelihood of the observed data given the Gompertz-Make ham process parameters. The PSO algorithm iteratively updates the particle positions and velocities to search for the parameter values that provide the best fit to the observed data. Both algorithms offer distinct approaches to parameter estimation of the Gompertz-Make ham process [21, 25]. The MMLE-PSO algorithm combines the advantages of the MMLE method and the optimization capabilities of PSO, leveraging additional information and optimizing the likelihood function. The PSO algorithm directly explores the parameter space to find the optimal parameter values that maximize the fitness function. In the following sections, we provide a detailed explanation of each algorithm, including the steps involved, initialization of parameters and particles, update rules, and convergence criteria. We also compare the performance of both algorithms and discuss their strengths and limitations in estimating the parameters of the Gompertz-Make ham process.

Algorithm (2): PSO Method

1. Initialize PSO Parameters
* Set number of particles N
* Set maximum iterations $i\ max$
* Set PSO constants: acceleration coefficients C_1 and C_2, inertia weight w.
* Define parameter search space: a_{min}, a_{max}, b_{min}, b_{max}, c_{min}, c_{max}, d_{min}, d_{max}.

2. Initialize Particle Population
* Randomly initialize positions a_i,b_i,c_i,d_i within parameter bounds
* Randomly initialize velocities v_a,v_b,v_c,v_d
* Evaluate fitness of each particle using
* the Fitness = Negative Log-Likelihood NLL of observed data. For Equation 8.

3. Set Initial Best Values
* Assign each particle's initial position as its personal best (p best).
* Identify the global best (g best) among all particles

4. Iterate Until Convergence or Max Iterations
* For each particle
a. Update velocity using PSO equation:
v new = w * v old + C_1 * r_1 * (p best - position) + C_2 * r_2 * (g best - position)
b. Update position:
$position$ new = $position$ old + v new
c. Enforce parameter bounds:
If $position$ new exceeds limits, reset within bounds
d. Evaluate fitness of new position
- Update personal best (p best) if the new position has better fitness.
- Update global best (g best) if any particle achieves the best fitness.

5. Stopping Criteria: If max iterations are reached or the convergence threshold is met, stop.

6. Return Optimal Parameters: Best estimated values for (a, b, c, d) minimizing the NLL.

6.3. Cuckoo Search Algorithm for Parameter Estimation

In this section, we estimate the NHPP SRGM's mean value function by using the Cuckoo Search (CS) method, taking into account all evaluation criteria mentioned in Table 4. The suggested algorithm is delineated as follows:

1. Identify each of the number of particles $N = 50$; the number of iterations with $i\ max = 100$.

2. The positions of each particle, representing estimations for all parameters, are randomly determined. Initially, these positions are generated from a uniform distribution within the range $[0, 1]$.

3. We define the objective function based on the models defined in Table 1.

4. The fitness function is set as the RMSE, which is equals to $RMSE = \sqrt{\frac{\sum_{i=1}^{Q} (\hat{\gamma}_i - \gamma)}{Q}}$

5. Generate the initial population randomly.

6. Enter the main loop of the CS.

7. The Cuckoo Search algorithm is based on the following equations:
* Levy Flight:

$$s_i(t+1) = s_i(t) + \alpha * L(\lambda) * (s_i(t) - s_j(t)), \tag{16}$$

* Cuckoo's Nest Seletion:

$$s_j(t+1) = s_i(t+1), \tag{17}$$

* Random Walk:

$$s_i(t+1) = s_i(t) + \alpha \times N(0,1), \tag{18}$$

8. By periodically replacing nests, the algorithm can explore the search space more effectively and potentially find better solutions.

9. Perform Greedy selection.

10. The estimators of parameters are adjusted based on the resultant value of the objective function RMSE.

11. Steps 4 and 7 are repeated until $i\ max$ is reached.

6.4. Bayesian Estimation

In this section, we estimate the parameters of Equation 3 using the algorithm [11]:

Step 1: Model Specification.
* Define the model of Equation 3.
* Function lambda (t,a,b,c): return $a + be^{ct}$,
* Function likelihood (data,a,b,c,t_0)
$L = 1.0$
For i in range (len(data))
t_i = data[i] * time
y_i = data[i] * value
$L =$ (a + b exp(c t_i)) $* exp($at$_0)$ + (a/b)* $(exp($ct$_0)-1)$
Return L

Step 2: Prior Distributions
* Define priors for a, b and c.
* Function prior a (a,α_a,β_a):
Return gamma pdf (a,α_a,β_a).
* Function prior b (b,α_b,β_b):
Return gamma pdf (b,α_b,β_b).
* Function prior c (c,α_c,β_c):
Return normal pdf (c,μ_c,σ_c).

Step 3: Posterior Distribution
* Define the un-normalized posterior distribution

* Function posterior (data, a, b, c, t_0, a, β_a, α_b, β_b, μ_c, σ_c):
$L\ value = L$ (data, a, b, c, t_0)
$prior_a$ = $prior_a$ (a, α_a, β_a)
$prior_b$ = $prior_b$ (b, α_b, β_b)
$Prior_c$ = $prior_c$ (c, μ_a, σ_a)
Return $L\ value$ * $prior_a$ * $prior_b$ * $prior_c$

Step 4: MCMC Sampling (Metropolis-Hastings Algorithm)
* Initialize parameters.
* Initialize storage for samples.
* Set number of iterations $= 10000$.
* Set proposal distribution.
* Run MCMC.
* Compute acceptance probability.
* Accept or reject the proposed values.
* Store the current values.

Step 5: Posterior Summaries.

Step 6: Model Evaluation.

7. Sensitivity Analysis for the Proposed Gompertz-Make ham Process (GMP) Model

The variations in parameters a, b and c along with their precise effects on output $\lambda(t)$ receive evaluation through sensitivity analysis. The analysis helps to determine: which model factors cause the most changes in output behavior because it provides understanding about model stability. The estimation process must achieve high precision rates when determining sensitive model inputs. Model refinement needs guidance to determine parameters that demand adaptive or dynamic changes. For the Gompertz-Make ham process intensity function, we analyze Equation 3. Local sensitivity analysis conducts an instantaneous sensitivity analysis of $\lambda(t)$ parameters by evaluating each derivative of $\lambda(t)$ separately [26].

$$\frac{\partial\lambda\left(\mathrm{t}\right)}{\partial\mathrm{a}} = 1, \tag{19}$$

$$\frac{\partial\lambda\left(\mathrm{t}\right)}{\partial\mathrm{b}} = e^{ct}, \tag{20}$$

$$\frac{\partial\lambda\left(\mathrm{t}\right)}{\partial\mathrm{c}} = \mathrm{bte}^{ct}, \tag{21}$$

The global sensitivity analysis (GSA) analyzes variations in a, b, and c based on their effects on the total variability of $\lambda(t)$. Sobol indices serve to determine parameter influence through variance-based measurement.

$$var\left(\lambda\left(\mathrm{t}\right)\right) = var\left(a\right) + var\left(be^{ct}\right) + Cov\left(a, be^{ct}\right), \tag{22}$$

where the Sobol sensitivity index for each parameter is:

Table 1. Sensitivity Analysis of Parameters for Optimal Release Time Estimation

Parameters	-10%	0	10%
a	0.02921	0	- 0.01534
b	0.01070	0	- 0.00766
c	0.03886	0	0.00066

$$S_a = \frac{var\left(E\left[\lambda|a\right]\right)}{var\left(\lambda\right)}, \tag{23}$$

$$S_b = \frac{var\left(E\left[\lambda|b\right]\right)}{var\left(\lambda\right)}, \tag{24}$$

$$S_c = \frac{var\left(E\left[\lambda|\mathrm{c}\right]\right)}{var\left(\lambda\right)}, \tag{25}$$

Where $E[\lambda|a]$ represents the expected value of $\lambda(t)$ given a fixed a, and similarly for b and c. Then S_a ¿ S_b ¿ S_c. We prove that c is the most sensitive parameter by computing the relative sensitivity function, defined as:

$$RS_\theta = \frac{\frac{\partial \lambda(\mathrm{t})}{\partial \theta}}{\lambda\left(t\right)}, \tag{26}$$

Applying this to each parameter for a:

$$RS_a = \frac{1}{\lambda\left(t\right)}, \tag{27}$$

And for b:

$$RS_c = \frac{e^{ct}}{\lambda\left(t\right)}, \tag{28}$$

Then,

$$RS_c = \frac{bte^{ct}}{\lambda\left(t\right)}, \tag{29}$$

8. Simulation

Simulation is a scenario designed to compare any system with the real world and is defined as the attempt to simulate a particular process under specific circumstances using artificial methods that resemble natural conditions. This includes building a smaller model that is an identical copy of the real model and performing tests on the miniature model examining the results and generalizing them to the original model, or computer simulation by writing a program for the methods to be chosen under realistic programming conditions and then observing the results obtained with the program and drawing a conclusion based on them [27].

There are different simulation methods, namely the (analogy method), the (mixed method) and the (Monte Carlo method). The Monte Carlo method is one of the most important and widely used simulation methods, in which a random sample of the phenomenon is generated that corresponds to the behavior of a certain probability distribution that the phenomenon has. To achieve this, the probability distribution of the phenomenon it has (CDF) is known that the set of samples random in this way possesses the property of independence because random samples in this method are by applying the mathematical method to each sample separately [28].

To put the previously discussed ideas into practice, the practical part of the research focused on the estimators of the suggested model during the fuzzy phase for both of the approaches used, utilizing a simulation method. The objective was to apply the Root Mean Square Error (RMSE) statistical criteria to various sample sizes in order to assess the optimality of these estimators. The purpose of the simulation model was to provide a comparison study of the approaches that were evaluated while accounting for a variety of real data situations. By showing how the estimate techniques affect the following variables, this strategy seeks to determine the best technique for estimating parameters inside the interval of the Gompertz-Make ham process [29].
• Change in sample size.
• Change in model parameter values.

8.1. Stages of Building a Simulation Experience

First stage: It is the most important stage on which the program's steps and procedures depend. Below are the steps for this stage:

Step1. Choose default values for the parameters of the Gompertz-Make ham process. Several default values were chosen for the shape parameter a and the scale parameter b for the Rayleigh Process by reviewing previous studies and experimenting with many default values for the parameters, which led us to choose the best of these values, as follows: ($a = 0.5; 0.6; b = 0.5; 0.6; 0.7$ and $c = 0.5, 0.7$).

Step2. Choose sample sizes. Several different sample sizes (small, medium, large) were chosen as follows: ($n = 20; 50$).

Second stage: Data generation: At this stage, random data is generated using the inverse transformation method and according to the Gompertz-Make ham process, as follows:

Step1. Generating a random variable u_i that follows a uniform distribution with the interval $(0, 1)$ using the cumulative distribution function with the help of the Rand.

$$u_i \sim U\left(0,1\right), i = 0, 1, 2, ..., n, \tag{30}$$

where u_i represents a continuous random variable that follows a uniform distribution.

Convert the data generated in step (first) that follows a uniform distribution into data that follows a Gompertz-Make ham process using the inverse function (CDF) transformation method and according to Equation 11 and as in the following formula.

$$t_i = \sqrt{\frac{2b^2u}{a}}, i = 0, 1, 2, ..., n, \tag{31}$$

Third stage: At this stage, parameters are estimated over the period for the Gompertz-Make ham process Software Reliability Growth Models and for all methods, which are:

Table 2. Simulated RMSE Comparison of MMLE, PSO, CS and BM Estimates for Gompertz-Make ham Process Parameters

Parameters	Sample Size	Methods	RMSE ($\hat{a}$)	RMSE ($\hat{b}$)	RMSE ($\hat{c}$)
{a=0.5; b=0.6; c=0.7}	20	MMLE	0.0843	0.1267	0.1162
		PSO	0.0965	0.1453	0.1475
		CS	0.0868	0.1353	0.1262
		BM	0.1834	0.2243	0.1272
{a=0.6; b=0.5; c=0.7}		MMLE	0.0664	0.1328	0.0411
		PSO	0.1096	0.1668	0.0516
		CS	0.0764	0.1467	0.0510
		BM	0.1554	0.2239	0.0522
{a=0.6; b=0.7; c=0.5}		MMLE	0.1339	0.1516	0.1030
		PSO	0.1549	0.2079	0.1185
		CS	0.1449	0.1516	0.1120
		BM	0.2228	0.1627	0.2020
{a=0.5; b=0.6; c=0.7}	50	MMLE	0.0836	0.0393	0.0227
		PSO	0.0838	0.0541	0.0259
		CS	0.0837	0.0442	0.0239
		BM	0.1827	0.1551	0.1248
{a=0.6; b=0.5; c=0.7}		MMLE	0.0848	0.0676	0.0086
		PSO	0.0978	0.0682	0.0117
		CS	0.0859	0.0677	0.0107
		BM	0.1759	0.1766	0.1106
{a=0.6; b=0.7; c=0.5}		MMLE	0.0847	0.0959	0.0651
		PSO	0.0979	0.1682	0.1117
		CS	0.0949	0.1582	0.1107
		BM	0.1838	0.2671	0.1217

- Particle Swarm Optimization (PSO).
- Cuckoo Search (CS).
- Bayesian estimation.

Fourth stage: Experiment is repeated 1000 times.

The results provided in Table 2 reveal that, in the pursuit of estimating the parameters of the Gompertz-Make ham process, the Maximum Marginal Likelihood Estimation (MMLE) method yields superior outcomes compared to the Particle Swarm Optimization (PSO), Cuckoo Search (CS) and Bayesian estimation methods.

9. Applications of Real Data

9.1. First Data Set

To assess the practical utility of both methods, real-world data collected from the Badoush Cement Factory was employed. The recently established Badoush Cement Factory, located within Nineveh Governorate, holds immense significance as a pivotal unit of the General Cement Company in northern Iraq. It serves as a central hub for cement production, catering not only to the entire nation but specifically to Nineveh Governorate. The dataset encompasses consecutive operational periods represented in daily production figures, spanning from 1st April 2020 to 1st January 2022. Which represent $X = [3\ 8\ 2\ 4\ 1\ 1\ 2\ 3\ 1\ 1\ 1\ 1\ 3\ 2\ 3\ 1\ 1\ 1\ 2\ 3\ 5\ 6\ 5\ 2\ 1\ 1\ 4\ 1\ 4\ 3\ 1\ 3\ 1\ 1\ 7\ 2\ 5\ 1\ 2\ 1\ 1\ 3\ 3\ 1\ 6\ 1\ 2\ 3\ 3\ 1\ 3\ 2\ 1]$. To ensure the suitability of the data for analysis, a goodness of fit test is imperative a process elucidated in the subsequent sections.

9.1.1. Goodness of Fit Tests for Gompertz-Make ham Process with Estimated Parameter: In statistical analysis, the goodness-of-fit test holds paramount significance as it aids in the selection of an appropriate distribution that effectively aligns with the given data. This test is particularly crucial for lifetime data, where classical assessments often rely on graphical techniques to unveil the congruity of the dataset in question. This section focuses on a graphic scrutiny of the dataset, aiming to ascertain its suitability for the Gompertz-Make ham function. To achieve this, the distribution of cumulative days of operational periods between two stops is plotted against the logarithmic times of the process. A distinctive linear distribution of these points signifies a fitting match between the data and the function governing the rate of occurrence of a Poisson process is irregular over time. By taking the natural logarithm of the cumulative function dictating the rate at which the Gompertz-Make ham process occurs over time, the subsequent equation is derived:

$$\ln\left[m\left(t\right)\right] = \ln\left(\frac{1}{2\beta^2}\right) + 2, \tag{32}$$

By using the programming language MATLAB/R2019b, the following figure was obtained.

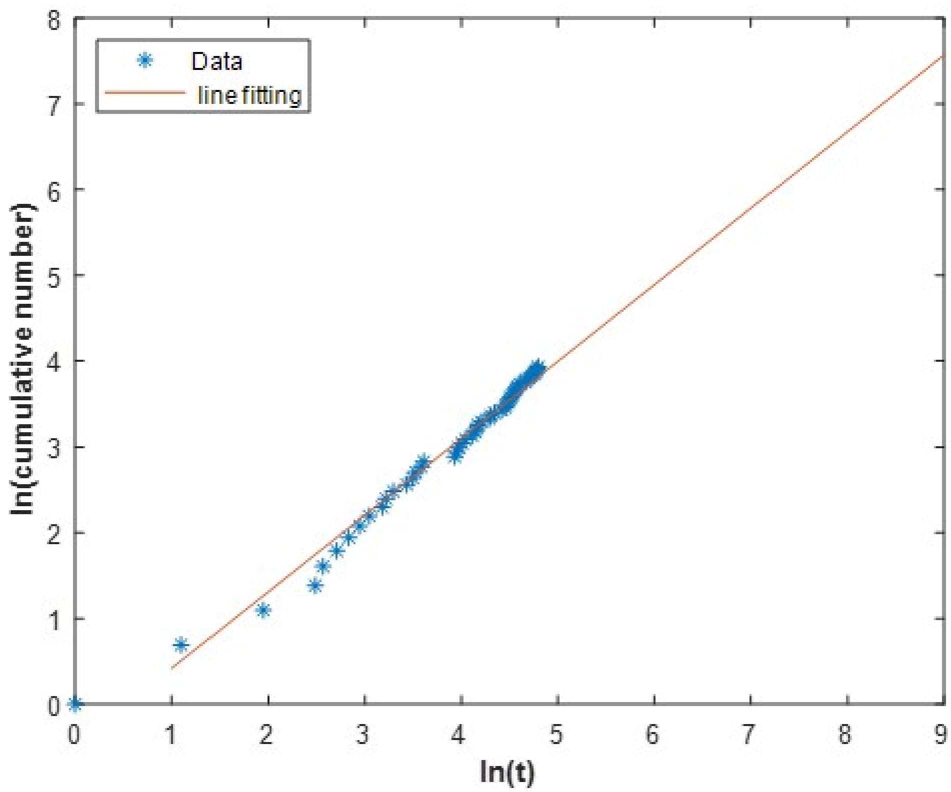

Figure 1. Scatter Plot of Logarithmic Distribution for Cumulative Number of Operating Periods in Days for a Raw Material

Figure 1 depicts the graphical distribution of cumulative days for operating periods and their corresponding occurrence times, presented on a logarithmic scale for the analyzed dataset. Notably, the scatter plot reveals a

Table 3. Value of the methods used in Gompertz-Make ham process parameter estimation.

Methods	RMSE	AIC	BIC
MMLE	0.0789*	114.9172*	134.0623
PSO	0.1044	141.8089	182.5319
CS	0.0889	114.7360	174.5497
BM	0.1085	151.2367	192.0322

discernible linear pattern, suggesting potential compatibility with modeling this data using the Gompertz-Make ham function.

Table 3 presents the Root Mean Square Error (RMSE), Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) values obtained from comparing the performance of the Modified Maximum Likelihood Estimator (MMLE), Particle Swarm Optimization (PSO) and Cuckoo Search (CS) techniques for calculating the Gompertz-Make ham process (GMP) rate. The results demonstrate that the MMLE method yields a lower RMSE, AIC and BIC values compared to the PSO and CS methods, indicating its superior efficiency in parameter estimation for the GMP process. Additionally, the figure below showcases the estimated time rate functions of the GMP obtained through both the MMLE and PSO estimation methods. The cumulative real numbers, which stand for the operational periods in days at the Northern Cement Company's Badoush Cement Plant, are compared to these estimations. The figure provides a visual representation of the accuracy and performance of the estimation methods in capturing the observed data patterns and modeling the rate of occurrence for the GMP process.

The efficiency of the MMLE estimation method compared to the PSO, CS, and Bayesian estimation in estimating the time rate functions of the GMP is demonstrated in Figure 2, where the estimated time rate function using the MMLE method is closer to the real data compared to the PSO, CS, and Bayesian estimation methods. Research into the Gompertz-Mack ham process has important practical implications in various fields such as manufacturing, healthcare, and finance. It improves decision accuracy by providing a reliable framework for predicting failure rates and operational efficiency, which contributes to improved maintenance strategies, resource allocation, and risk management. The flexibility of the model allows it to be customized to the needs of specific operational contexts, increasing reliability and performance. The accurate estimation of parameters in research also contributes to improved resource utilization and risk assessment, leading to more efficient strategies and actions. In addition, research supports progress in academic knowledge and improves practical applications by providing valuable insights for future innovations.

9.2. Data Set II

The second dataset, reported in [22], comprises 61 days of COVID-19 data recorded in Italy between 13 June and 12 August 2021. This dataset captures the daily number of newly reported cases. Figure 3 presents the P-P plots of the fitted distribution for this dataset. Table 3 provides a detailed summary of the model's value obtained using the Modified Maximum Likelihood Estimation (MMLE) method, the Particle Swarm Optimization (PSO), and the Cuckoo Search (CS). Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Root Mean Square Error (RMSE) for distribution.

Table 4 presents the Root Mean Square Error (RMSE), Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC) values obtained from evaluating the performance of the Modified Maximum Likelihood Estimation (MMLE) method, Particle Swarm Optimization (PSO), and Cuckoo Search (CS) in estimating the parameters of the Gompertz-Make ham Process (GMP). The dataset, reported in [22], consists of 61 days of COVID-19 incidence data recorded in Italy between 13 June and 12 August 2021, capturing the daily

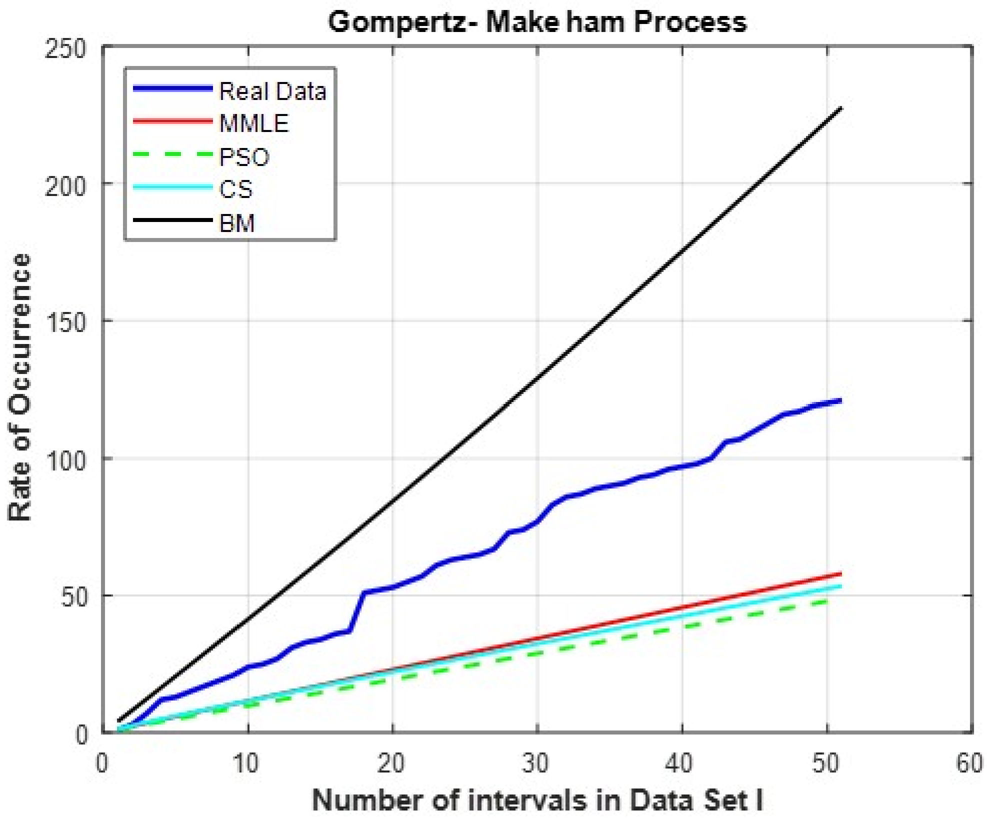

Figure 2. Estimates of the cumulative time rate of power outages for generation units compared to real data.

Table 4. Value of the methods used in Gompertz-Make ham process parameter estimation.

Methods	RMSE	AIC	BIC
MMLE	1.8039*	104.9172*	114.0623
PSO	1.9136	121.8089	162.5319
CS	1.8576	104.9360	154.5497
BM	2.8006	131.1108	165.2318

number of newly reported cases. The results indicate that the MMLE method achieves lower RMSE, AIC, and BIC values compared to PSO and CS, demonstrating its superior accuracy and efficiency in parameter estimation for the GMP model. Furthermore, Figure 3 illustrates the P-P plots of the fitted distribution for this dataset, providing a visual assessment of the goodness-of-fit. These findings highlight the effectiveness of the MMLE method in capturing the underlying data structure and improving the reliability of the estimated parameters in real-world epidemiological modeling.

9.3. Data Set III

The third data set is presented in [4]. It contains 26 observations that indicate the failure times for a specific product. This information has also been used in [4, 30]. Figure 4 presents the P-P plots of the fitted distribution for this dataset. Table 4 provides a detailed summary of the model's value obtained using the Modified Maximum Likelihood Estimation (MMLE) method, the Particle Swarm Optimization (PSO), and Cuckoo Search (CS).

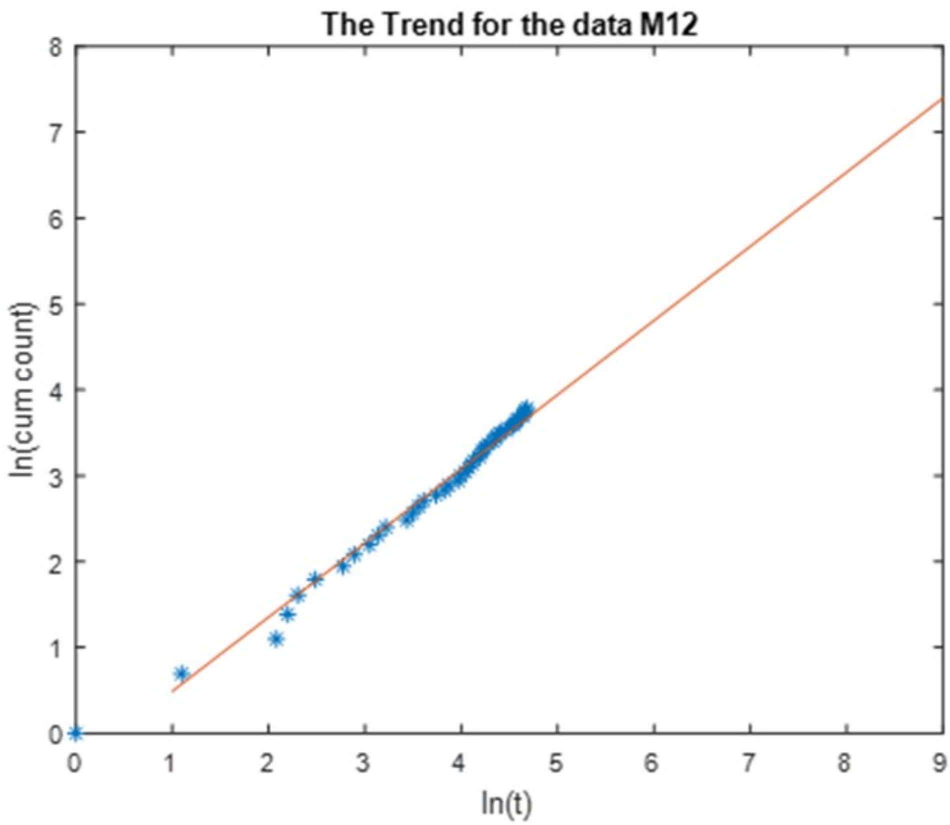

Figure 3. Shows a P-P plot of the Gompertz-Mack ham process with other estimated distributions for the second dataset and the goodness-of-fit test is shown.

Table 5. Value of the methods used in Gompertz-Make ham process parameter estimation.

Methods	RMSE	AIC	BIC
MMLE	1.5508*	100.9172*	104.0623*
PSO	1.6479	111.8089	152.5319
CS	1.5997	100.9360	144.5497
BM	2.9305	121.7079	154.4318

Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and Root Mean Square Error (RMSE) for distribution.

The performance evaluation of the Modified Maximum Likelihood Estimation (MMLE) method together with Particle Swarm Optimization (PSO) and Cuckoo Search (CS) for Gompertz-Make ham Process (GMP) parameter estimation produced results in Table 5 through Root Mean Square Error (RMSE) and Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). The dataset reported in [4]. It contains 26 observations that indicate the failure times for a specific product. The study shows that MMLE produces lower RMSE, AIC, and BIC statistics than PSO and CS indicating its excellence at estimating GMP model parameters. The P-P plots shown in Figure 3 evaluate the distribution match for this dataset by visual presentation. Research finds that MMLE proves effective for data structure identification and it enhances parameter reliability in genuine epidemiological modeling applications [31, 32].

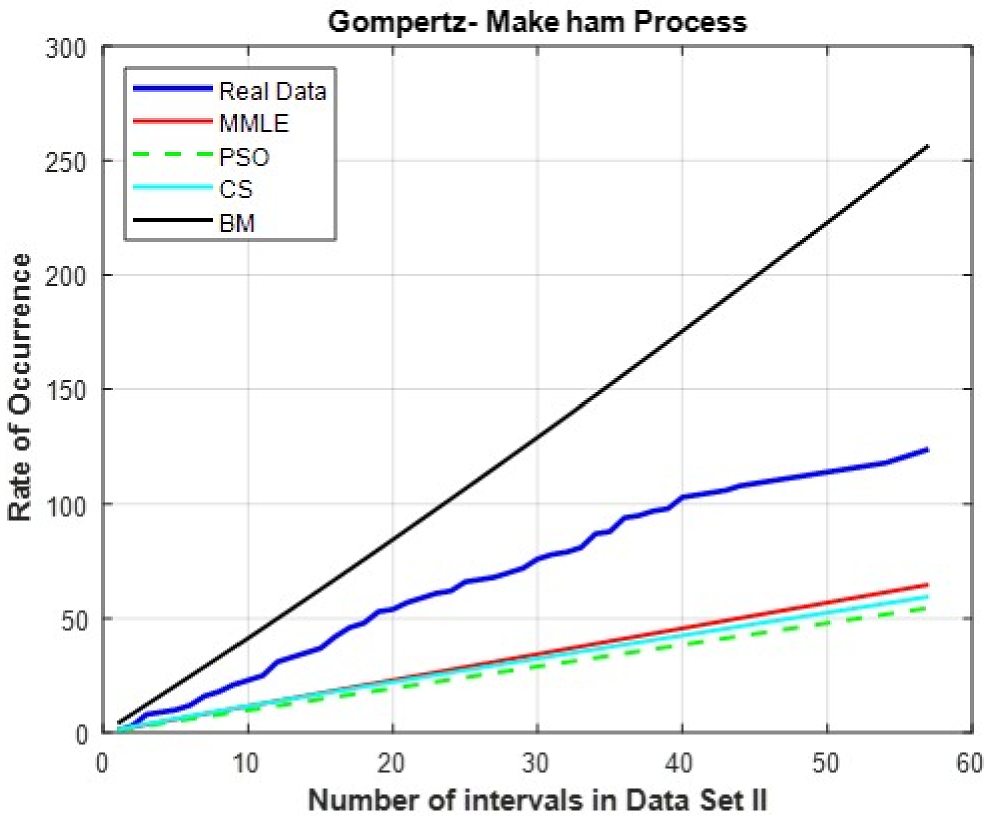

Figure 4. Estimated Cumulative Time Rate of COVID-19 Cases Recorded in Italy.

10. Discussion of Results

To facilitate a comprehensive comparison of the employed parameter estimation methods for the Gompertz-Make ham process, the assessment criterion was based on the formulation from Equation 4. The designed program, coded in MATLAB/R2019b, facilitated the calculation of the projected count of consecutive operational periods for the new Badoush Cement Factory's raw materials mill within the study's designated time frame. By employing this program, the RMSE was computed to gauge the dissimilarity between actual and estimated values of the average duration of plant shutdowns. These outcomes are synthesized in the subsequent table.

10.1. The Model's Adaptability

In summary, discussing the model's adaptability involves highlighting its relevance across various fields, addressing the need for context-specific parameter adjustments, validating its performance with real data, and recognizing limitations while suggesting possible modifications. This comprehensive approach provides a clear understanding of how the Gompertz-Make ham process can be effectively utilized in diverse applications.

10.2. Limitations of the Proposed Model

This study which analyzes Gompertz-Make ham Process (GMP) parameter estimation contains various restrictions which diminish the research findings' reliability and flexibility. The model depends on independent event occurrence together with time-unvarying parameter assumptions while real-world conditions feature dependence

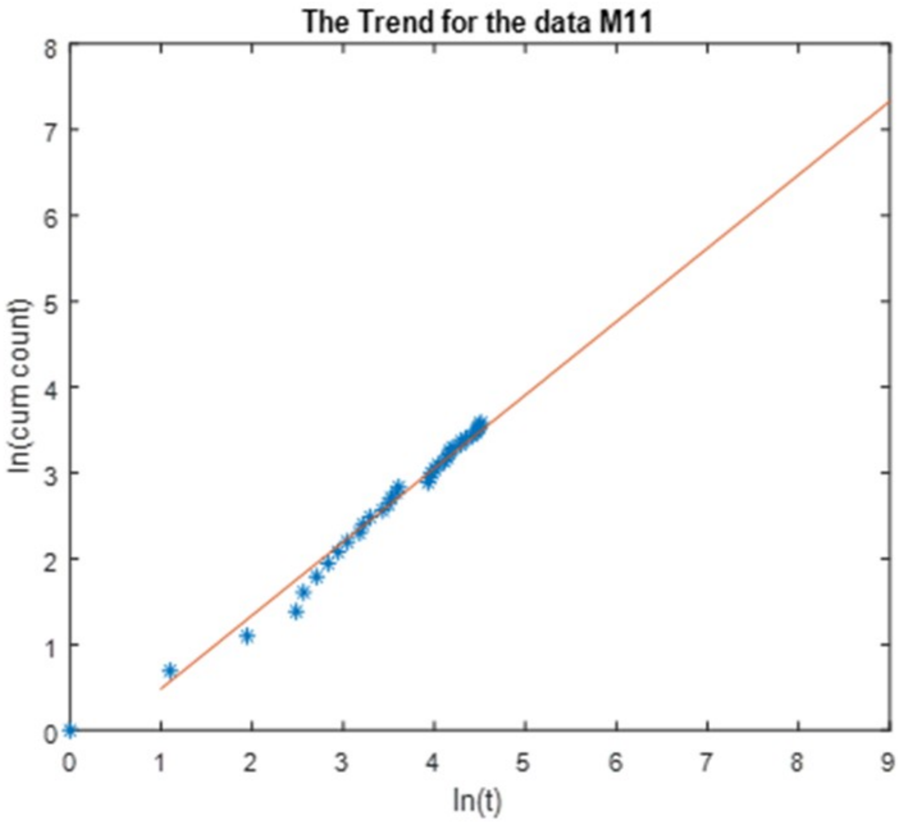

Figure 5. Illustrates the P-P plots are displayed for the Gompertz-Mack ham process for the third dataset, as well as the goodness-of-fit results of other evaluated distributions.

between events and variable parameters across time. The dataset obtained from the Badoush Cement Factory might fail to demonstrate sufficient variability between different contexts causing potential bias effects. The dataset quality issues generate doubts about how well parameters will be estimated. The Modified Maximum Likelihood Estimation (MMLE) demonstrates better performance than Particle Swarm Optimization (PSO) in numerous cases but both techniques become sensitive to particular parameter choices thus affecting the outcome reproducibility. The study findings require careful application to diverse situations because they depend on event dependencies while researchers should use alternative modeling techniques that maintain interpretability together with complexity management.

10.3. Properties of the GMP

A fundamental feature of the GMP is its ability to model varying hazard rates over time through the intensity function: $\lambda(t) = a + be^{cdt}, t \geq 0, a, b, c, d > 0$. This formulation captures both a baseline rate a and exponential growth dynamics b and c, allowing for flexible event modeling. The GMP effectively represents aging processes and accommodates non-linear increases in event occurrence, making it applicable to diverse real-world phenomena.

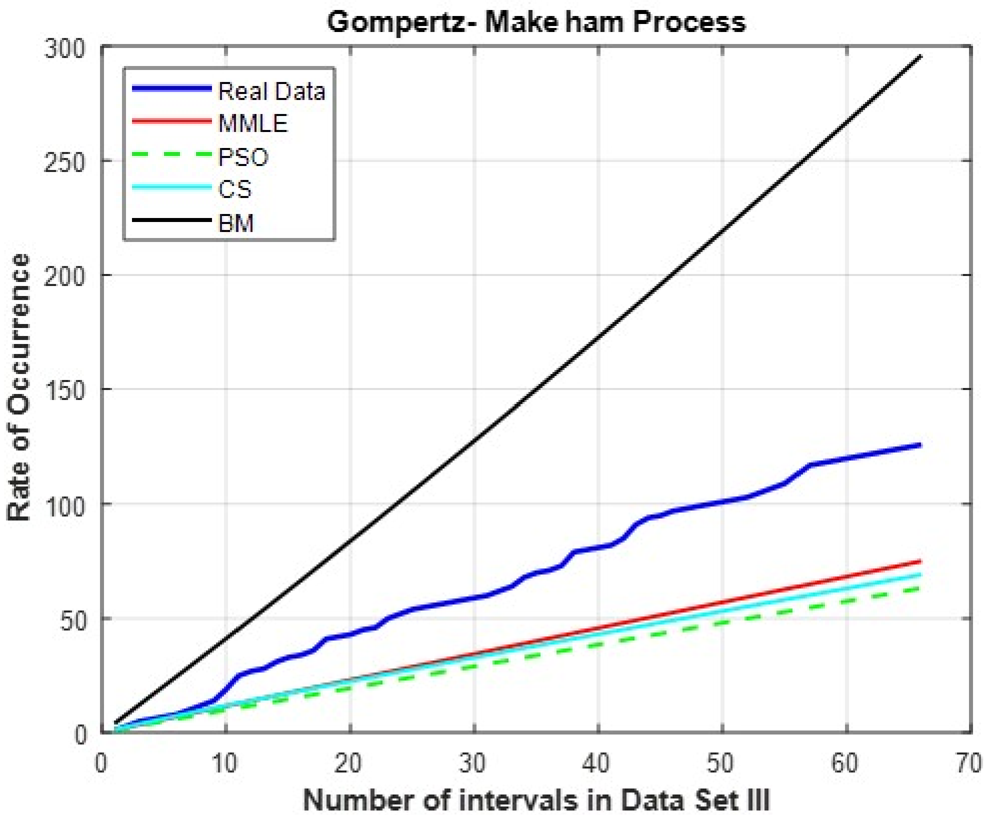

Figure 6. . Estimated Cumulative Time Rate of indicate the failure times for a specific product.

10.4. Assumptions

Despite its strengths, the GMP relies on key assumptions that may limit its applicability:
• Independence of Events: The assumption that events occur independently disregards dependencies present in interconnected systems, potentially leading to biased estimates.
• Time-Invariant Parameters: Assuming constant parameters may obscure underlying trends, limiting the model's adaptability to changing conditions in dynamic environments.
• Continuous Time Framework: The GMP models events in continuous time, which may not align with discretely recorded data, introducing practical inconsistencies.

10.5. Suggestions for Future Model

Time-dependent parameters that evolve with changing conditions need to be researched to achieve better simulations of event occurrences. The capability of the model to detect complex event interrelationships would increase by implementing either hierarchical models or copula-based methods for constructing dependence structures. New methodological approaches would make the model perform more accurately with actual data and expand its usefulness across multiple domains which would result in improved predictive efficiency in real-world applications.

11. Conclusions

This study presents a comprehensive approach to estimating the parameters of the Gompertz-Make ham Process (GMP), a non-homogeneous Poisson process widely used in reliability analysis, survival modeling, and event forecasting. By employing Modified Maximum Likelihood Estimation (MMLE), Particle Swarm Optimization (PSO), Cuckoo Search (CS), and Bayesian estimation, we compare the effectiveness of these methods in capturing the underlying structure of real-world event data. The findings indicate that MMLE consistently outperforms PSO and CS in terms of accuracy, as evaluated using Root Mean Square Error (RMSE), Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC). Bayesian estimation further provides uncertainty quantification, enhancing the robustness of parameter inference. The application of these methods to COVID-19 incidence data in Italy and failure times from the Badoush Cement Factory demonstrates the GMP's capability to model complex stochastic processes in diverse fields. The results highlight the significance of hybrid estimation techniques, particularly when handling non-homogeneous event intensities. The study also underscores the limitations of static parameter assumptions, suggesting the need for more adaptive modeling frameworks to improve predictive performance.

Acknowledgement

The authors are very grateful to the Duhok of Polytechnic University for providing access which allows for more accurate data collection and improved the quality of this work.

Ethical Approval

All the authors demonstrate that they have adhered to the accepted ethical standards of a genuine research study.

Competing Interests

No conflict of interest is declared by authors.

Author contributions

All authors have sufficiently contributed to the study and agreed with the results and conclusions.

Funding

No funding source is reported for this study.

Availability of Data and Materials

The data used to support the findings of this study are included in the article.

REFERENCES

1. A. Hussain, Y. Oraibi, Z. Mashikhin, A. Jameel, M. Tashtoush, E. Az-Zo'Bi, *New Software Reliability Growth Model: Piratical Swarm Optimization-Based Parameter Estimation in Environments with Uncertainty and Dependent Failures*, Statistics, Optimization & Information Computing, vol. 13, no. 1, pp. 209-221, 2025.
2. A. Hussain, K. D. Pati, A. K. Atiyah, M. A. Tashtoush, *Rate of Occurrence Estimation in Geometric Processes with Maxwell Distribution: A Comparative Study between Artificial Intelligence and Classical Methods*, Int. J. Advance Soft Compu. Appl, vol. 17, no. 1, pp. 1-15, 2025. https://doi.org/10.15849/IJASCA.250330.01
3. A. M. Abd El-Bar, *An extended gompertz-makeham distribution with application to lifetime data*, Communications in Statistics-Simulation and Computation, vol. 47, no. 8, pp. 2454-2475, 2018.
4. S. Ahamad, *Performability modeling of safety-critical systems through AADL*, International Journal of Information Technology, vol. 14, no. 5, pp. 2709-2722, 2022.
5. V. A. Chetlapalli, *Performance evaluation of IoT networks: A product density approach*, Computer Communications, vol. 186, pp. 65-79, 2022.
6. S. Chupradit et al., *A Multi-Objective Mathematical Model for the Population-Based Transportation Network Planning*, Industrial Engineering & Management Systems, vol. 21, no. 2, pp. 322-331, 2022.
7. E. Az-Zo'bi et al., *Novel topological, non-topological, and more solitons of the generalized cubic p-system describing isothermal flux*, Optical and Quantum Electronics, vol. 56, no. 1, pp. 1-16, 2024.
8. C. N. Davi, *Pso-pinn: Physics-informed neural networks trained with particle swarm optimization*, arXiv preprint arXiv, :2202.01943, 2022.
9. I. Ibrahim, W., Taha, M., Dawi, A., Jameel, M., Tashtoush, E., Az-Zo'bi, *Various Closed-Form Solitonic Wave Solutions of Conformable Higher-Dimensional Fokas Model in Fluids and Plasma Physics*, Iraqi Journal For Computer Science and Mathematics, vol. 5, no. 3, pp. 401-417, 2024.
10. E. A. Az-Zo'bi, K. A. Dawoud, and M. Marashdeh, *Numeric-analytic solutions of mixed-type systems of balance laws*, Applied Mathematics and Computation, vol. 265, pp. 133–143, 2015.
11. A. Hussain, K. Mahmood, I. Ibrahim, A. Jameel, S. Nawaz, M. Tashtoush, *Parameters Estimation of the Gompertz-Makeham Process in Non-Homogeneous Poisson Processes: Using Modified Maximum Likelihood Estimation and Artificial Intelligence Methods*, Mathematics and Statistics, vol. 13, no. 1, pp. 1-11, 2025. http://doi.org/10.13189/ms.2025.130101.
12. H. Hussein, S. Hussein, A. Hussain, M. Tashtoush, *Estimating Parameters of Software Reliability Growth Models Using Artificial Neural Networks Optimized by the Artificial Bee Colony Algorithm Based on a Novel NHPP*, Mathematical Modelling of Engineering Problems, vol. 12, no. 1, pp. 25-36, 2025. https://doi.org/10.18280/mmep.120104.
13. T. O. Hodson, *Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not*, Geoscientific Model Development, vol. 15, no. 14, pp. 5481-5487, 2022.
14. E. A. Az-Zo'bi, *Construction of solutions for mixed hyperbolic elliptic Riemann initial value system of conservation laws*, Applied Mathematical Modelling, vol. 37, no. 8, pp. 6018–6024, Apr. 2013.
15. Y. W. Zhang, *A comprehensive survey on particle swarm optimization algorithm and its applications*, Mathematical problems in engineering, vol. 2015, pp. 1-38, 2015. http://doi.org/10.1155/2015/931256
16. Y. S. Zhou, *Testing for the Markov property in time series via deep conditional generative learning*, arXiv preprint arXiv: 2305.19244., 2023.
17. S. F. Junior, *An application of Particle Swarm Optimization (PSO) algorithm with daily precipitation data in Campina Grande, Paraíba, Brazil. Research*, Society and Development, vol. 9, no. 8, pp. 325-339, 2020.
18. A. Hussain, Y. Oraibi, M. Sulaiman, A. Abdulghafour, *Parameters Estimation of a Proposed Non-Homogeneous Poisson Process and Estimation of the Reliability Function Using the Gompertz Process: A Comparative Analysis of Artificially Intelligent and Traditional Methods: nonhomogenous poisson process with intelligent*, Iraqi Journal For Computer Science and Mathematics, vol. 5, no. 2, pp. 36-47, 2024.
19. M. Zyphur, F. Oswald, *Bayesian estimation and inference: A user's guide*, Journal of Management, vol. 41, no. 2, pp. 390-420, 2015.
20. G. N. Lavanya, *Parameter estimation of goel-okumoto model by comparing aco with mle method*, International Research Journal of Engineering and Technology, vol. 4, no. 3, pp. 1605-1615, 2017.
21. S. Chupradit et al., *Modeling and Optimizing the Charge of Electric Vehicles with Genetic Algorithm in the Presence of Renewable Energy Sources*, Journal of Operation and Automation in Power Engineering, vol. 11, no. 1, pp. 33-38, 2023.
22. S. A. Jabr, *Gompertz Fréchet stress-strength Reliability Estimation*, Iraqi Journal of Science, vol. 23, no. 3, pp. 4892-4902, 2021.
23. P. G. Erto, *The generalized inflection S-shaped software reliability growth model*, IEEE Transactions on Reliability, vol. 69, no. 1, pp. 228-244, 2018.
24. N. Shirawia, A. Kherd, S. Bamsaoud, M. Tashtoush, A. Jassar, E. Az-Zo'bi, *Dejdumrong Collocation Approach and Operational Matrix for a Class of Second-Order Delay IVPs: Error analysis and Applications*, WSEAS Transactions on Mathematics, vol. 23, pp. 467-479, 2024.
25. H. Zureigat, M. Tashtoush, A. Jameel, E. Az- Zo'bi, M. Alomare, *A solution of the complex fuzzy heat equation in terms of complex Dirichlet conditions using a modified Crank-Nicolson method*, Advances in Mathematical Physics, vol. 2023, 6505227.
26. E. Almetwally, D. Abdo, E. Hafez, T. Jawa, N. Sayed-Ahmed, H. Almongy, *The new discrete distribution with application to COVID-19 Data*, Results Physics, vol. 2022, no. 32, 104987, 2022.
27. N. Shirawia, A. Qasimi, M. Tashtoush, R. Rasheed, M. Khasawneh, E. Az-Zo'bi, *Performance Assessment of the Calculus Students by Using Scoring Rubrics in Composition and Inverse Function*, Applied Mathematics and Information Sciences, vol. 18, no. 5, pp. 1037-1049, 2024.
28. M. Gacula, J. Kubala, *Statistical models for shelf life failures*, Journal of Food Sciences, vol. 40, pp. 404–409, 1975.
29. P. Nagar, *Application of Goel-Okumoto model in software reliability measurement*, Int. J. Comp. Appl. Special Issue ICNICT, vol. 5, pp. 1-3., 2012.

30. K. J. Song, I. H. Chang, H. Pham, *Optimal release time and sensitivity analysis using a new NHPP software re-reliability model with probability of fault removal subject to operating environments*, Applied Sciences, vol. 8, no. 5, pp. 714-729, 2018.
31. A. S. Hussain, *Estimation of Rayleigh Process Parameters Using Classical and Intelligent Methods with Application*, 8th International Conference on Contemporary Information Technology and Mathematics (ICCITM), pp. 447-452, Mosul: IEEE, 2022.
32. S. Ahamad, R. Gupta, *Performability modeling of safety-critical systems through AADL*, International Journal of Information Technology, vol. 14, no. 5, pp. 2709-2722, 2022. https://doi.org/10.1007/s41870-022-01004-w