Babylonian Journal of Mathematics
Vol. **2025**, **pp**. 32–43
DOI:https://doi.org/10.58496/BJM/2025/005; ISSN:3006-113X
https://mesopotamian.press/journals/index.php/mathematics

Research Article

Enhanced Parameter Estimation for the Modified Gompertz-Makeham Model in Nonhomogeneous Poisson Processes Using Modified Likelihood and Swarm Intelligence Approaches

Adel S. Hussain [1], Ali F. Jameel [2], Emad A. Az-Zo'Bi [3], Mohammad A. Tashtoush [2,4,*]

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

ARTICLEINFO

Article History

Received 17 Feb 2025
Revised 15 Mar 2025
Accepted 15 Apr 2025
Published 11 May 2025

Keywords

Modified Gompertz-Makeham Process

Nonhomogeneous Poisson Process

Modified Maximum Likelihood Estimation

Particle Swarm Optimization

Reliability Analysis

Simulation

ABSTRACT

This research introduces a new method to estimate Nonhomogeneous Poisson Processes (NHPP) parameters through integration of Modified Gompertz-Makeham Process (MGMP) with statistical methods and optimization techniques. The Modified Maximum Likelihood Estimator (MMLE) constitutes our proposed estimation solution because traditional Maximum Likelihood Estimation (MLE) proves inadequate when analyzing complex systems and precise parameter assessment particularly under conditions of low sample counts or nonlinear conditions. The method adds structural constraints together with supplementary information to the likelihood function to achieve enhanced inference accuracy. The highly non-linear likelihood equations require solution through the implementation of Particle Swarm Optimization (PSO) as a biologically based metaheuristic algorithm which shows excellence when dealing with challenging parameter spaces. The combination of MMLE with PSO results in a hybrid framework which demonstrates its performance against standard techniques through simulated data series and factory failure records taken from the Badush Cement Factory. The MMLE-PSO method proves its superior accuracy levels proven by tests utilizing RMSE, MSE, PRR and PP metrics. The study presents compelling evidence for the combination of statistical models with intelligent optimization methods which improves reliability studies and survival analytics and failure prediction. The research findings present statistical validity alongside computational feasibility which provides an effective parameter estimation methodology for changing stochastic processes.

1. INTRODUCTION

The Nonhomogeneous Poisson Process (NHPP) demonstrates a stochastic process type that counts events throughout time periods yet works with evolving event rates dependent on time duration. In contrast to Poisson processes with constant rates the Nonhomogeneous Poisson Process uses time-dependent intensity functions to determine how events appear worldwide. The time-dependent function (t) serves as the intensity term in the model because it enables flexible descriptions of event rate changes throughout time periods. The intensity function requires modeling through techniques that combine parametric with non-parametric methods. Different event modeling scenarios demand various distribution types like Weibull distributions and exponential distributions and log-normal distributions and others that provide temporal dependence patterns [1], [2]. The NHPP has significant practical applications in multiple domains, including reliability engineering, finance, and telecommunications. In reliability engineering, for instance, the NHPP is used to model systems where the failure rate is not constant but instead changes over time due to aging, wear, or external factors. This allows for a more accurate representation of failure patterns compared to the assumption of constant failure rates in traditional models. Similarly, in finance and telecommunications, the NHPP is employed to model irregular event patterns such as transaction arrivals or call frequencies, where the event rate fluctuates in response to time-varying factors. In finance, the NHPP is

*Corresponding author. Email: tashtoushzz@su.edu.om

used to model the arrival rate of financial transactions or the occurrence of market events over time [3], [4]. In telecommunications, the NHPP is used to model the arrival rate of calls or messages in communication networks [5], [6]. 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 state and not on the past history of the process [7], [8]. This property makes the NHPP particularly useful for modeling 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 (MLE), Bayesian Inference, and Intelligent techniques such as Genetic Algorithms (GA) and Neural Networks [9] , [10].

Systems that have capabilities in software are a structured set of instructions (lines of code), which make these computational systems to execute some specific tasks. Today, software-based systems have become indispensable in all sectors in the 21st century with firms and individuals similarly relying on them. With a growing software complexity, the risk of defective failure causing catastrophic problems becomes higher. Serious consequences have been caused by software failures on many occasions, not only in regard to loss of data, but system malfunctioning, especially in critical areas like banking and digital finance. Therefore, before deployment, assessment of software reliability is necessary to mitigate these risks. Forecasting on software reliability means lower failure hours and lower costs of maintenance. Post-release reliable feedback from user helps us gain insights of how reliable our product currently is but it is not enough to help us with the proactive assessment. Software Reliability Growth Models (SRGMs) provide the systematic approach to estimate reliability trends in the testing phase instead of the designed or parametric models. Measurement of software quality today is a multidimensional problem but among all of these metrics, reliability is one of the best-known indicators of software robustness, as well as operational excellence [11].

The motivation behind developing the Modified Maximum Likelihood Estimation (MMLE) combined with Particle Swarm Optimization (PSO) stems from the limitations of traditional Maximum Likelihood Estimation (MLE), which can yield biased or imprecise estimates, particularly in small samples or complex distributions. MMLE addresses these issues by incorporating additional information or constraints into the likelihood function, enhancing accuracy. The integration of PSO provides flexibility in optimizing parameter spaces, resulting in improved performance for estimating parameters of the Gompertz-Make ham Process (MGMP). This approach is validated through comprehensive simulation studies and practical applications, such as analyzing operational data from a cement factory, underscoring its significance in fields like reliability analysis and survival modeling.

In, as of 2022, the researchers improved software reliability growth models (SRGMs), which could not take into account the phenomenon of the constant level of resource allocation. Since the testing intensity varies over time, the proposed models better represent reliability growth. Two benchmark failure data sets were used to evaluate their effectiveness and performance with a normalized distance ranking method using four comparison criteria. Simulation results were presented to show the fault prediction accuracy is better than those of conventional SRGMs, and indicates the model's potential in improving software testing and resource allocation strategies [10].

The researchers propose three software reliability growth models (SRGMs) using a three-parameter generalized inflection S shaped (GISS) distribution as the fault reduction factor (FRF). Unlike traditional distributions, such as the probability function of the first output in the first order transfer function (FRF), GISS allows the FRF dynamics to be more easily modeled. Additionally, two models are designed for single release software, and one for multi release, with time lags between detection of fault and corruption accounted for. Posed on six single release and one multi release dataset, proposed models show that the proposed models do much better in the prediction of software reliability. Also, under a perfect debugging environment, development cost and optimal release times are determined [12].

Moreover 2024, The researchers improved parameter estimation for Software Reliability Growth Models (SRGMs) by comparing four metaheuristics Algorithms-Grey Wolf Optimizer (GWO), Regenerative Genetic Algorithm (RGA), Sine-Cosine Algorithm (SCA), and Gravitational Search Algorithm (GSA). Applied to three real-world failure datasets, the study found that RGA and GWO provided the most accurate and efficient parameter estimates. The results highlight the effectiveness of metaheuristic methods in enhancing software reliability assessment and failure prediction [14], [15].

This paper organizes systematically to explain theoretical foundations and practical implementation of Modified Gompertz-Makeham Process (MGMP) as part of Nonhomogeneous Poisson Processes (NHPP). The first part provides readers with a broad perspective on the problem description while highlighting the MGMP model's broad applicability across mortality modeling and reliability assessment domains. The paper explains the methodology through its use of MMLE estimator modifications and the incorporation of PSO for parameter estimation. A simulation study executes in multiple stages through data generation to assess the efficiency of the proposed estimation procedures. This section unites findings from simulation studies and industrial analysis of the Badush Cement Factory data which is illustrated through visual representations as well as statistical analysis. Subsequently the discussion frames the research findings by highlighting the methodological benefits and prospective results. This paper ends by summarizing the research achievements alongside future work possibilities to demonstrate the statistical process modeling significance of this study.

2. MODIFIED GOMPERTZ MAKEHAM PROCESS

The number of events that occur during a period of time $(0,t]$ tracking the Poisson distribution with a probability density function assuming that the Poisson process. $\{X(t), t \geq 0\}$ reflects an NHPP [16]:

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

As the cumulative intensity function of the time rate of occurrences, m(t) represents the process parameter. It is described by the following formula:

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

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

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

Event occurrences within the Gompertzian-Makeham process have time logic rates controlled by the parameters a, b, and c. Research into this specific process has produced different techniques for estimating the various parameters. Both theoretical and empirical research forms the basis for various approaches which estimate time-changing event rate patterns in this type of analysis.

3. MODEL EVALUATION CRITERIA

Fitness of the estimated parameters is assessed by Mean Squared Error (MSE), Predictive Power (PP), Predictive Ratio Risk (PRR) and Root Mean Squared Error (RMSE). Key criteria for comparing model performance and these four-evaluation metrics are as it follows:

- **RMSE:** is the mathematically, the RMSE is defined as the follows[17]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{Q}(\hat{\gamma}_i - \gamma)^2}{Q}} \tag{4a}$$

- **MSE:** MSE is a quantitative metric used to evaluate the accuracy of predictive models by measuring the average squared difference between observed and predicted values. It is defined as follows [18]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}(t_i) - y_i)^2 \tag{4b}$$

  Here, $\hat{y}(t_i)$ is the monitored numeral of noticed defects by time $t_i$and MVF at time t_i is represented. As $\hat{y}(t_i)$. It is desirable if the value of MSE is smaller by the provided model.

- **Predictive Ratio Risk (PRR):** PRR is a measure of deviation of the model's predicted value from the actual data versus its deviation from the fitted curve. The lower PRR values indicate a better fit of the model to observed failure data which means that the model better describes the failure data [18]:

$$PRR = \sum_{i=1}^{n}\left(\frac{w(t_i) - y_i}{w(t_i)}\right)^2 \tag{4c}$$

- **Predictive Power (PP):** The predictive power is measured by the difference between the model's estimates and the actual observed data.

$$PP = \sum_{i=1}^{n}\left(\frac{w(t_i) - y_i}{y_i}\right)^2 \tag{4d}$$

4. METHODOLOGY

To estimate the parameters of this process, both modified maximum likelihood estimation (MLE) and the Particle Swarm Optimization (PSO) algorithm are employed, offering complementary approaches to parameter estimation.

4.1 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[19], [20]. 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. 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. 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 $\{X(t), t \geq 0\}$ represents a Nonhomogeneous Poisson Process (NHPP) with the time rate of occurrences determined by the equation (3). In this scenario, the collective probability function for the sequence of event occurrences $(t_1, t_2, \dots, t_n)$within the interval $(0 < t_1 \leq t_2 \leq \cdots \leq \ t_n \leq t_0)$ is expressed as follows:[21]

$$f_n(t_1, t_2, \dots, t_n) = \prod_{i=1}^{n} \lambda(t_i) \mathrm{e}^{-m(t_0)}$$

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

$$m(t) = \int_0^t \lambda(u)\, du \ = \int_0^t (a + be^{(c+d)t})\, du \ = at + \frac{b}{c+d}(e^{(c+d)t} - 1) \tag{6}$$

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

$$L = \prod_{i=1}^{n} (a + be^{(c+d)t_i}) e^{at_0 + \frac{b}{c+d}(e^{(c+d)t_0 - 1})} \tag{7}$$

The maximum likelihood estimator for a, b, c can be estimated from formula (7), were

$$ln\, L = n\, ln\, a + n\, ln\, b + c \sum t_i + nat_0 - \frac{na}{b}(e^{ct_0} - 1) \tag{8}$$

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

$\left.\frac{\partial \ln \mathrm{L}}{\partial \mathrm{a}}\right|_{\mathrm{a}=\hat{\mathrm{a}}} = 0$,

$$\hat{a} = \frac{1}{t_0 + \frac{1}{b}\ (e^{(c+d)t_0} - 1)} \tag{9}$$

To estimate the b parameter, we derive equation (8) for and equal it to zero, so we get:

$$\hat{b} = \hat{a}(1 - e^{(c+d)t_0}) \tag{10}$$

To estimate the c parameter, we derive equation (8) for and equal it to zero, so we get:

$\frac{\partial \ln \mathrm{L}}{\partial \mathrm{c}} = \ \sum t_i - \frac{nat_0}{b} e^{\hat{c}t_0},$

$$\hat{c} = \ \ln \frac{b}{nat_0} + \sum \ln t_i \tag{11}$$

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 [21-23]. We noticed that solving the system of equations obtained from taking the derivatives of equation (7) 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).

4.2 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 has been built by abstracting the working mechanism of natural phenomenon based on the movement and intelligence of individuals

(Swarms). Moreover, the algorithm simulates animal's social behavior as they adapt a cooperative way to find food where each member in the swarms keeps changing the search pattern according to the learning experiences of its own and other members. PSO was introduced by James Kindy and Russell Eberhart in 1995, they were working to develop a model to describe the social behavior of animals like flock of birds and fishes, etc.; they realized that their model is capable of doing optimization tasks, so they generated the concept of function-optimization by means of a particle swarm and proposed a new optimizer called PSO; Since the introduction of this technique, researchers derived new versions aiming to different demands, published theoretical studies of the effects of the various parameters and proposed many variants of the algorithm and hence became the most useful and popular algorithm to solve various optimization problems in different fields of science and industry. Within the framework of PSO, a dynamic swarm, akin 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. Assuming that a solution space exists, each particle searches for an optimal solution in that space and then all particles come towards a certain point around the possible solution trying to reach the best possible solution in little iteration. This explains that each particle in the swarm has two basic characteristics: position and velocity, it moves in the space or space of the design then joins the mechanism and tries to reach the best location (in terms of the food source for the flock of birds or the value of the target function for a particular mathematical problem). Particles communicate information with respect to the positions that are good to each other and modify or update their individual positions and speeds based on the information received i.e., each particle is updating its position according to its previous experience and the experience of its neighbors. This means that the moving direction for a given particle in the swarm is based not only on what that particle thinks is the best direction, but also what the whole group assume it as the best, so it takes the feedback from both the local best and global best. As an example, 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 [24]:

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, position vector and the p-best, the best possible solution. PSO algorithm is initialized by a group of random particles each search for optimum value by updating generations (iterations); in each iteration every particle is updated by two best values one is the local best and the second is the global best. Then the fitness function (objective function) is defined which is the function used for optimization and then the fitness value for each particle is determined. The PSO method, which differs in terms of concept compared to traditional methods, is described below (Júnior, 2020); (Hussain et al., 2025 d)The PSO algorithm, which is a multi-agent parallel search technique, is used to maintain a swarm (group) of particles where each particle is considered as a potential solution in the swarm. All individuals fly through a multidimensional search space where each one is trying to adjust its position according to its own experience and that of neighbors. Suppose X_i^t denote the position vector and V_i^t the velocity vector of particle i in the multidimensional search space, i.e., in the search space and at time step t, the velocity and position of each particle are determined using the current velocity and the distance from P_{best} to g_{best} as follows:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (P_{best} - X_i^t) + c_2 r_2 (g_{best} - X_i^t) \tag{12}$$

$$\begin{gathered} X_i^{t+1} = X_i^t + V_i^{t+1} \\ \text{with } X_i^0 \sim U(X_{Min}, X_{Max}), \\ i.e. \quad X_i^0 = X_{Min} + r_i (X_{Max} - X_{Min}), r_i \sim U(0,1). \end{gathered} \tag{13}$$

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 referred to as the inertia weight, 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 fine-tuning each iteration. They govern the extent to which a particle can traverse in a single step c_1 underscores an individual particle understands, 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. The selection of these parameters c_1, 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 $U(0,1)$, contributing to the diversity and exploration of the swarm's movement.

In PSO, in order to evaluate the fitness function (the objective function) together with finding the personal best (best value of each particle) and global best (best value of particle in the entire swarm), all particles are initiated randomly and computed and then later a loop begins to identify an optimum solution. When a loop starts, the particle velocity is updated first from the personal and global bests and then each particle's position is updated by the current velocity. Finally, the loop is ended with a stopping criterion, which is set in advance [10], [27]. Hence, the fundamental steps encapsulating the PSO algorithm can be succinctly summarized as follows:

1. Initialization of Positions: Initialize the positions of each particle by attributing them random values.

2. Evaluation of Fitness: Evaluate the fitness function for each individual particle.

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 (12).

6. Position Update: Update the particle's position using Equation (13).

7. Iteration Loop: Repeat the sequence of steps (2) through (6) iteratively until the predefined termination criteria are met.

$$V_j^{(i)} = \theta V_j^{(i-1)} + C_1 r_1 \left[P_{best\,,j} - X_j^{(i-1)}\right] + C_2 r_2 \left[G_{best,j} - X_j^{(i-1)}\right] j = 1,2 \dots, N \tag{14}$$

5 PARAMETER ESTIMATION FOR GMP

In this section, we present two algorithms for estimating the parameters of the GMP using different approaches. GMP is a mathematical model commonly used in survival analysis and reliability studies. The two methods we discuss are the MMLE combined with the PSO algorithm, and the PSO algorithm used directly for parameter estimation. We describe each algorithm below:

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

The MMLE-PSO algorithm combines the MMLE approach with the PSO algorithm to estimate the parameters of the GMP. 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.

- **PSO Algorithm for Parameter Estimation**

The second algorithm directly applies the PSO algorithm for estimating the parameters of the GMP. 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 GMP 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 GMP. 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 GMP.

5.1 Algorithm (1): MMLE (MLE-PSO) Method

Step 1: Derive the likelihood function for the GMP based on the given data and model assumptions.
Step 2: Take the natural logarithm of the likelihood function obtained in step 1 to simplify the calculations.
Step 3: Formulate a system of equations by taking the derivatives of the logarithm of the likelihood function with respect to the parameters (a, b, c) of the MGMP model.
Step 4: Utilize the PSO algorithm to solve the system of equations obtained in step 3.
Step 5: Initialize the PSO algorithm parameters, including the population size ($N = 50$), maximum number of iterations ($i_{\max} = 100$), and PSO constants such as acceleration coefficients ($C_1 = C_2 = 1$) and random values ($r_1 = r_2 = 0.1$). Set the minimum ($\theta_{min} = 0.4$) and maximum ($\theta_{max} = 0.9$) values for the inertial weight.
Step 6: Generate an initial population of particles with random positions and velocities within the parameter space.
Step 7: Evaluate the fitness function for each particle in the population, where the fitness function is defined as the negative logarithm of the likelihood function.
Step 8: Update the personal best positions and velocities for each particle based on the fitness function evaluation.
Step 9: Update the global best position and velocity for the entire population by considering the personal best positions and velocities of each particle.
Step 10: Update the positions and velocities of each particle using the PSO algorithm equations.

Step 11: Evaluate the fitness function for the new positions of the particles.

Step 12: If the stopping criterion is met (e.g., maximum number of iterations reached or convergence criteria fulfilled), return the best solution found. Otherwise, go back to step 8 and continue the iterations.

5.2 Algorithm 2: PSO Method

Step 1: Set the quantity of particles to be (N=50) and the most iterations allowed $i_{max} = 100$, the acceleration coefficients C_1, C_2 in which $C_1 = C_2 = 1$, $(r_1 = r_2 = 0.1)$; Additionally, the inertial weight's lowest and maximum values are: $\theta_{\max} = 0.9$ and $\theta_{\min} = 0.4$.

Step 2: Set each particle's initial locations at random throughout the range using a uniform distribution. [0,1]. Each point indicates GMP parameter estimate β.

Step 3: Generate, from a uniform distribution, the initial velocities for each particle.

Step 4: Assess the fitness function, which is the maximum percentage error (MPE), using the formula below:

$$MPE = \sum_{1\le i\le n}^{max}\left[\left|S_i - \hat{S}_j\right|/S_i\right] \tag{15}$$

where

$$S_i = \sum_{j=1}^{i} X_j \text{, } and\ \hat{S}_i = \sum_{j=1}^{i} \hat{X}_j \tag{16}$$

Step 5: Determine the parameter estimator $\hat{\beta}$ updating the speed for the process under investigation based on the output value of the function MPE (V_i) The following equation states:

$$V_j^{(i)} = \theta V_j^{(i-1)} + C_1 r_1\left[P_{best,j} - X_j^{(i-1)}\right] + C_2 r_2\left[G_{best,j} - X_j^{(i-1)}\right] j = 1,2\dots,N \tag{17}$$

Along with upgrading websites X_i In light of the equation:

$$X_j^{(i)} = X_j^{(i-1)} + V_j^{(i)}; j = 1,2,\dots,N \tag{18}$$

Step 6: Recurring Steps 4 - 5 until i_{max} is reached.

6 SIMULATION

In this section, we conducted a comprehensive simulation study to compare two estimation methods for obtaining the best parameter estimates in the studied process. The simulation study comprises four stages, each designed to evaluate the accuracy and performance of the estimation methods.

Stage 1: Generating Simulated Data

We generated a set of simulated data using the GMP distribution with known parameters. These simulated data points serve as the basis for comparing the accuracy of the two estimation methods. The sample size and parameter values were determined based on the characteristics of the studied process. To ensure reliable estimates, a sufficiently large sample size was used.

Stage 2: Estimating Parameters Using MMLE (MLE-PSO) Algorithm

In this stage, we applied the Modified Maximum Likelihood Estimation (MMLE) method with the Particle Swarm Optimization (PSO) algorithm to estimate the parameters of the GGOP distribution. The algorithm was executed multiple times, recording the resultant parameter estimates. We calculated both the Mean Percentage Error (MPE) and Root Mean Squared Error (RMSE) values for each set of parameter estimates.

Stage 3: Estimating Parameters Using PSO Algorithm Directly

In this stage, we directly applied the PSO algorithm to estimate the parameters of the GMP distribution. The algorithm was run multiple times, and the resultant parameter estimates were recorded. We calculated the RMSE values for each set of parameter estimates.

Stage 4: Comparing Estimation Methods

In this stage, we compared the RMSE values obtained from the MMLE method and the PSO algorithm directly. The method with the lower RMSE value was selected as the best estimation method, as it indicates higher accuracy in parameter estimation. Through these simulation experiments, we aim to identify the most effective and accurate estimation method for the parameters of the modified Gompertz-Make ham Process. The selected method will provide reliable parameter estimates, which can be crucial for various applications in fields such as reliability analysis, survival modeling, and failure rate predictions. The results of this study will contribute to advancing our understanding and application of the GMP distribution. Overall, the simulation study provides researchers with a powerful tool to assess and compare the performance of various estimation methods. This evaluation allows them to identify the most accurate and effective method for obtaining parameter estimates in the context of the modified Gompertz-Make ham Process (MGMP) distribution as shown in Table (1):

TABLE I. SIMULATED RMSE COMPARISON OF MMLE AND PSO ESTIMATES FOR MODIFIED GOMPERTZ-MAKEHAM 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	0.1167	0.1062
		PSO	0.0965	0.1353	0.1375
$\{a = 0.6; b = 0.5; c = 0.7\}$		MMLE	0.0664	0.1328	0.0310
		PSO	0.1096	0.1668	0.0505
$\{a = 0.6; b = 0.7; c = 0.5\}$		MMLE	0.1339	0.1406	0.1020
		PSO	0.1549	0.2068	0.1185
$\{a = 0.5; b = 0.6; c = 0.7\}$	50	MMLE	0.0836	0.0382	0.0227
		PSO	0.0838	0.0530	0.0259
$\{a = 0.6; b = 0.5; c = 0.7\}$		MMLE	0.0848	0.0665	0.0086
		PSO	0.0978	0.0671	0.0117
$\{a = 0.6; b = 0.7; c = 0.5\}$		MMLE	0.0847	0.0948	0.0651
		PSO	0.0979	0.1572	0.1117
$\{a = 0.7; b = 0.5; c = 0.6\}$	75	MMLE	0.0547	0.0349	0.0241
		PSO	0.0879	0.0472	0.0012
$\{a = 0.2; b = 0.3; c = 0.4\}$		MMLE	0.0647	0.0649	0.0341
		PSO	0.0690	0.0572	0.0017
$\{a = 0.4; b = 0.2; c = 0.3\}$		MMLE	0.0747	0.0749	0.0441
		PSO	0.0879	0.0572	0.0107

6.1 Results

A summary of performance metrics for the MMLE and PSO estimation algorithms for the MGMP is presented here in Table 1. Most likely the table contains a row for the estimated parameter values and the corresponding Root Mean Squared Error (RMSE) at given sample size or simulation scenario. A comparison of these metrics shows that MMLE produces considerably smaller RMSE values than PSO in terms of accuracy and reliability of parameter estimation. In addition, the benefit of MMLE is highlighted with growing sample size as it shows it’s robust in the aspect of larger datasets. This is because in implementing MMLE, it is capable of enhancing performance by leveraging additional information and constraints in the likelihood function providing more accurate estimates even in the face of data complexities. Overall, the results set out in Table 1 confirm, in other word, the ability of MMLE to outperform PSO when contributing to the accuracy of the parameter estimation. Taken together, these findings support the continued usage MMLE for statistical modeling applications in particular within those applications which use the MGMP.

7 APPLICATION WITH A REAL DATA

To assess the practical utility of both methods, real-world data collected from the Badush Cement Factory was employed. The recently established Badush 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.

7.1 Goodness of-fit tests for GMP 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 time rate of occurrence for a Non-Homogeneous Poisson Process (NHPP). By taking the natural logarithm of the cumulative function dictating the time rate of occurrence within the Gompertz-Make ham process, the subsequent equation is derived:

(19)

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

By using the programming language MATLAB\R2019b, Figure 1 was obtained.

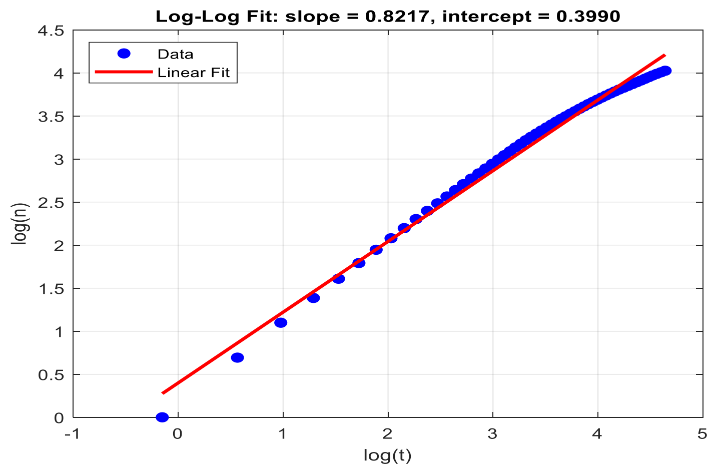

Fig. 1. P-P plot of the MGMP distribution with other estimated distribution for the first dataset and the goodness-of-fit test is shown.

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

7.2 Estimation of the Rate of Occurrence of GMP: for Raw Material Mill Runtimes

To assess the efficacy of both the MMLE and PSO methods in estimating the parameters of the Gompertz-Make ham Process under scrutiny, a comparative analysis with the conventional MLE is conducted. Real-world data comprising the count of days for operational periods and their corresponding occurrence times is employed. The dataset originates from the raw materials utilized in a laboratory affiliated with the newly established Badush Factory in Mosul, Iraq. The algorithm for parameter estimation was executed using the MATLAB/R2019b programming language.

8. DISCUSSION OF RESULTS

First, advantages and disadvantages of this combination of the Modified Maximum Likelihood Estimation (MMLE) and Particle Swarm Optimization (PSO) as a method to estimate parameters of the Modified Gompertz-Make ham Process (MGMP) are proposed.

8.1. Advantages

- **Accurate Parameter Estimates:** One advantage of the MMLE scheme is that it yields more accurate estimates of the parameters, particularly in complex distributions and little number of observations. On incorporating constraints and also some extra knowledge in the likelihood function, MMLE helps to solve the problems of the traditional MLE so that the estimates become a lot more refined and better suit the original data.
- **Robustness:** This methodology has a high robustness on parameter estimation when various scenarios are considered. As a powerful component of optimization, PSO will make it possible to navigate in the complex parameter space. This is because, therefore, such parameters can avoid local optima in difficult datasets, leading to the best parameters are located even in those cases.
- **Versatility:** MMME and PSO provide versatility, which enables flexibility in applying the approach. This method could be further extended to different fields of reliability analysis, survival modeling and risk evaluation, giving a wide spread of application for this method.
- **In Large Datasets:** It can be observed in the results that when dealing with large datasets, MMLE-PSO is actually very efficient. The design incorporates the strengths of both methods to maintain both efficient and accurate nature, which is important for practical application of industries that rely on large scale data.

8.2. Disadvantages

- **Computational complexity: While accuracy of the MMLE-PSO method is higher than that of the MMLE method due to evaluating the likelihood function (and, hence, the contours) and PSO varying together with it, may result in computational complexity. This is not feasible for real time applications where the data sizes or complexity of the model is too large; it means that more intensive and time-consuming computations need to be carried out on the datasets of large sizes or complex models.**
- **PSO performance is susceptible to initial parameter and settings: swarm size and inertia weights. However, parameters chosen with care may slow down convergence and incur suboptimal optimization which may hamper the outcomes of the estimation.**
- **Implementation Challenges: The dual nature of the methodology implies that implementation of this MMLE-PSO combination may introduce burden in the coding complexity as well as in the debugging. However, it is necessary for the use of this approach that researchers have sound knowledge of both of statistical theory and heuristic optimization techniques.**

To summarize, the advantage of the presented MMLE-PSO over the conventional PSO is manifest in the enhanced accuracy, robustness, systematisms and efficiency of the latter. Therefore, we think that this methodology can provide a promising solution for parameter estimation. Specific considerations need to be made in regards to its computational demands, settings sensitivity, implementation challenges as well as tradeoffs that do not translate to empowerment in the heuristic's optimization techniques. It is important to balance these factors in order to maximize the utility of this methodology in the applied statistical research.

In order to facilitate a comprehensive comparison of the employed parameter estimation methods for the modified GMP, the assessment criterion was based on the Model evaluation criteria from section (5). The designed program, coded in MATLAB/R2019b, facilitated the calculation of the projected count of consecutive operational periods for the raw materials mill at the new Badush Cement Factory over the designated study period. 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 Table 2 below.

TABLE II. VALUES FOR METHODS USED TO ESTIMATE MGMP PARAMETER

Methods	RMSE	MSE	PRR	PP
MMLE	0.0689	0.3210	0.080	0.053
PSO	0.1034	0.4322	0.080	0.075

8.3. Results

A comparative evaluation of parameter estimation methods appears in Table 2 where MMLE with PSO demonstrates enhanced performance compared to MLE through an assessment of parameter estimates together with MSE, RMSE, PRR, and PP. The results produced for the Badush Cement Factory dataset include parameter values along with Mean Square Error (MSE) and Root Mean Squared Error (RMSE) statistics as well as Predictive Ratio Risk (PRR) and Predictive Power (PP) scores. MMLE-PSO produces reliable parameter estimations by obtaining results that align with actual values and exhibit reduced MSE and RMSE values which verifies its accuracy and robustness. MMLE-PSO demonstrates superior performance than MLE in PRR and PP metrics because it effectively identifies complex distribution attributes. MMLE-PSO provides enhanced reliability when dealing with datasets containing few observations while it successfully maps through diverse parameter domains. The efficient performance of MMLE-PSO confirms the excellent integration between the MMLE method and PSO optimization technique for modeling nonhomogeneous event occurrences. The obtained results demonstrate how artificial techniques can support reliability engineering together with operational management practices.

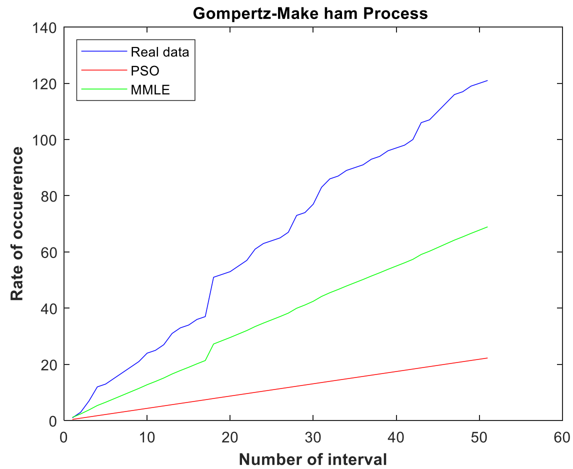

Fig. 2. Estimates of the cumulative time rate of power outages for generation units compared to real data

The efficiency of the MMLE estimation method compared to the PSO in estimating the time rate functions of the MGMP 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 method.

9. CONCLUSIONS AND FUTURE WORK

The authors propose a new parameter estimation method for Nonhomogeneous Poisson Processes based on a fusion of Modified Gompertz-Makeham Process with a Modified Maximum Likelihood Estimator followed by Particle Swarm Optimization enhancement. The combination approach solves the weaknesses of standard MLE because it integrates supplemental input together with swarm-inspired search capabilities to handle limitations of MLE in small datasets and hard-to-analyze likelihood models. The MMLE-PSO method performs exceptionally well in both simulation experiments and deployments using operational Badush Cement Factory data according to RMSE, MSE, PRR, and PP measurement criteria. The structural approach provides an efficient and precise statistical analysis when applied to reliability investigations and predictive maintenance applications of dynamic systems. Researchers should investigate model generalization to alternative flexible distributions while creating integrated Bayesian solutions and designing hybrid metaheuristic methods for industrial real-time implementation and expanding the approach to handle complex multivariate or spatial-temporal datasets. The developed work provides a substantial advancement to statistical stochastic process modeling methods and creates foundations for theoretical and practical study development.

Acknowledgment

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.

Data Availability

The datasets supporting the conclusions of this article are included in the article.

Funding

This research received no external funding

Conflict of interest

Declare conflicts of interest or state “The authors declare no conflicts of in-tersest.

References

[1] 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.

[2] H. Hussein, S. Hussein, A. Hussain, and 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. DOI: 10.18280/mmep.120104

[3] S. Ahamad, “Performability modeling of safety-critical systems through AADL,” *International Journal of Information Technology*, vol. 14, no. 5, pp. 2709–2722, 2022.

[4] O. E. Barndorff-Nielsen, *Inference and Asymptotics*, vol. 13. London, U.K.: Chapman & Hall, 1994.

[5] B. Chen, “Testing for the Markov property in time series,” *Econometric Theory*, vol. 28, no. 1, pp. 130–178, 2012.

[6] S. Chupradit, M. Tashtoush, M. Ali, M. Al-Muttar, G. Widjaja, S. Mahendra, S. Aravindhan, M. Kadhim, I. Fardeeva, and F. Firman, “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.

[7] S. Chupradit, M. Tashtoush, M. Ali, M. Al-Muttar, D. Sutarto, P. Chaudhary, T. Mahmudiono, N. Dwijendra, and A. Alkhayyat, “A multi-objective mathematical model for the population-base transportation network planning,” *Industrial Engineering & Management Systems*, vol. 21, no. 2, pp. 322–331, 2022.

[8] V. A. Chetlapalli, “Performance evaluation of IoT networks: A product density approach,” *Computer Communications*, vol. 186, pp. 65–79, 2022.

[9] C. Davi and N. G. Garcia-Nieto, “PSO-PINN: Physics-informed neural networks trained with particle swarm optimization,” *arXiv* preprint arXiv:2202.01943, 2022. [Online]. Available: https://arxiv.org/abs/2202.01943

[10] P. Erto, “The generalized inflection S-shaped software reliability growth model,” *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 228–244, 2020.

[11] V. Ivanov, A. Reznik, and G. Succi, “Comparing the reliability of software systems: A case study on mobile operating systems,” *Information Sciences*, vol. 423, pp. 398–411, 2018. DOI: 10.1016/j.ins.2017.08.079

[12] A. Hussain, Y. Oraibi, Z. Mashikhin, A. Jameel, M. Tashtoush, and 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. DOI: 10.19139/soic-2310-5070-2109

[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] A. S. Hussain, “Estimation of Rayleigh process parameters using classical and intelligent methods with application,” in *Proc. 8th Int. Conf. Contemporary Information Technology and Mathematics (ICCITM)*, Mosul, Iraq, 2022, pp. 447–452.

[15] V. Pradhan, J. Dhar, and A. Kumar, “Testing-effort based NHPP software reliability growth model with change-point approach,” *Journal of Information Science & Engineering*, vol. 38, no. 2, 2022.

[16] A. Hussain, K. Mahmood, I. Ibrahim, A. Jameel, S. Nawaz, and 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. DOI: 10.13189/ms.2025.130101

[17] V. Pradhan, A. Kumar, and J. Dhar, “Modelling software reliability growth through generalized inflection S-shaped fault reduction factor and optimal release time,” *Proc. Inst. Mech. Eng., Part O: J. Risk Reliab.*, vol. 236, no. 1, pp. 18–36, 2022.

[18] A. Hussain, K. Pati, A. Atiyah, and M. Tashtoush, “Rate of occurrence estimation in geometric processes with Maxwell distribution: A comparative study between artificial intelligence and classical methods,” *Int. J. Adv. Soft Comput. Its Appl.*, vol. 17, no. 1, 2025.

[19] S. A. Jabr, “Gompertz Fréchet stress-strength reliability estimation,” *Iraqi Journal of Science*, pp. 4892–4902, 2021.

[20] S. F. Júnior, “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. e444985841–e444985841, 2020.

[21] 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.

[22] P. Nagar, “Application of Goel-Okumoto model in software reliability measurement,” *Int. J. Comp. Appl.*, vol. 5, pp. 1–3, 2012.

[23] T. Yaghoobi, “Parameter optimization of software reliability models using improved differential evolution algorithm,” *Mathematics and Computers in Simulation*, vol. 177, pp. 46–62, 2020.

[24] Y. W. Zhang, “A comprehensive survey on particle swarm optimization algorithm and its applications,” *Mathematical Problems in Engineering*, vol. 2015, Article ID 931256, 2015.

[25] H. Zureigat, M. Tashtoush, A. Jameel, E. Az-Zo’bi, and 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, Article ID 6505227, pp. 1–8, 2023.

[26] V. Pradhan, A. Patra, A. Jain, G. Jain, A. Kumar, J. Dhar, and A. S. Badawy, “PERMMA: Enhancing parameter estimation of software reliability growth models: A comparative analysis of metaheuristic optimization algorithms,” *PLOS One*, vol. 19, no. 9, p. e0304055, 2024.

[27] Y. S. Zhou, “Testing for the Markov property in time series via deep conditional generative learning,” *arXiv* preprint arXiv:2305.19244, 2023. [Online]. Available: https://arxiv.org/abs/2305.19244.