



## Employing Intelligent Methods to Estimate the Parameters of the Proposed Generalized Goel Process

Sulaiman, M.S.

Department of Statistics and Informatics, College of Computer Science and Mathematics University of Mosul, Mosul, Iraq

### Article information

#### Article history:

Received April 19, 2024

Revised: November 13, 2024

Accepted November 18, 2024

Available online December 1, 2024

#### Keywords:

Generalized Goel process

Parameters Estimation

Maximum Likelihood Estimator

PSO

Simulation

#### Correspondence:

Sulaiman, M.S.

[muthanna.sulaiman@uomosul.edu.iq](mailto:muthanna.sulaiman@uomosul.edu.iq)

### Abstract

This paper will present the Goel distribution as the occurrence rate of the non-homogeneous Poisson process (NHPP) to improve its occurrence rate, it is proposed to be called the Generalized Goel Process (GGP). As for the estimation of parameters of this process, a number of methods were discussed, the maximum likelihood estimator (MLE) was suggested and after that – a modification to this method was necessary due to the fact that it was impossible to find estimators using it. An intelligent algorithm of the likelihood function was added with the parameter and was known as the Modified Maximum Likelihood Estimator (MMLE). MMLE was then compared with another intelligent method the Particle Swarm Optimization (PSO) in estimating occurrence rate of the proposed Goel process to determine the best estimator of the process. Besides, the paper contains the simulation of the mentioned process and an example of its practical usage. The simulation and application results showed that the MMLE approach gave higher accuracy estimates than the PSO algorithm for the majority of the studied sample sizes, especially for the larger sizes.

DOI: [10.33899/ijqjoss.2024.185239](https://doi.org/10.33899/ijqjoss.2024.185239) , ©Authors, 2024, College of Computer Science and Mathematical, University of Mosul.

This is an open access article under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

### 1. Introduction

The Non-homogeneous Poisson Process (NHPP) analyses and models failure data in repairable systems. This process is one of the important stochastic processes, which is a generalization of the homogeneous Poisson process. NHPP assumes that the number of events that occur in a process period is a Poisson distribution with independent increments. Furthermore, the occurrence of events in NHPP is monotonic over a period, which is a function of time called the intensity function, and is denoted by  $\lambda(t)$  [4,15].

Goel process (GP) is a well-known reliability engineering model used to analyse the failure data of repairable systems. It is a non-homogeneous Poisson process that assumes that the failure rate of a system decreases over time due to the presence of some latent failures that are gradually eliminated during the system's operation. This process was introduced in 1979 by Jitendra Goel and Kiyoshi Okumoto [3], and it has been widely used in the reliability analysis of complex systems in various fields such as engineering, finance, and healthcare. It depends on the following assumptions [10]:

- The probability of failures at time  $t$  is Poisson distributed with mean rate function  $m(t)$ .
- If  $\Delta t$  is approaching 0 the total count of failures taking place in the interval  $(t, t + \Delta t]$  is related to the expected number of non-suspected errors.
- That means for any given finite subset of times  $(t_1 < t_2 < \dots < t_n)$  the number of failures that occur in each of the interval's  $t_i - t_{i-1}$  are independent.

In conclusion, the Goel process is a useful tool to assess the reliability of the complicated systems, and the method is applied in practice to enhance the reliability and safety of systems in many fields.

**2. Material and methods**

In this section, the theoretical background of the generalized Goel process (GGP) shall be presented as well as two parameter estimation methods, namely the Modified Maximum Likelihood Estimation (MMLE) and Particle Swarm Optimization (PSO) algorithms. Furthermore, a presentation of the proposed algorithms for the assessment of the rate of occurrence of GGP will be given.

**3. Generalized Goel process**

This paper will therefore present the Goel distribution as a rate of occurrence for the NHPP to improve on its rate of occurrence whereby the newly developed process is known as the Generalized Goel process (GGP). As for the estimation of parameters in this process, a number of methods were discussed, the maximum likelihood estimator (MLE) was suggested and after that a modification to this method was necessary due to the fact that it was impossible to find estimators using it. An intelligent algorithm of the likelihood function was added with the parameter and was known as the Modified Maximum Likelihood Estimator (MMLE). MMLE was then compared with another intelligent method the Particle Swarm Optimization (PSO) in estimating occurrence rate of the proposed Goel process to determine the best estimator of the process. Besides, the paper contains the simulation of the mentioned process and an example of its practical usage.

Supposing the process  $\{N(t), t \geq 0\}$  represents the NHPP, number of events occurring in a time interval  $(0, t]$  follows a Poisson distribution with a density function [13]:

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

where  $m(t)$  represents the process parameter, is the mean occurrence rate, and is defined as follows:

$$m(t) = E[N(t)] = \int_0^t \lambda(u) du, \quad 0 \leq t < \infty \tag{2}$$

where  $\lambda(u)$  denotes the rate of occurrence of a non-homogeneous stochastic process or intensity function, which is described by the Generalized Goel distribution as follows:

$$\lambda(t) = abce^{-bt^c} t^{c-1}; \quad t \geq 0, \quad a, b > 0, \quad c > 1 \tag{3}$$

where  $a, b$  and  $c$  are the parameters for generalized Goel distribution, parameter  $a$  is the expected total number of failures, parameters  $b$  and  $c$  represent the quality of the testing process. It can be seen that  $\lambda(t)$  increases at the beginning of the process and then decreases when  $a, b > 0$  and  $c > 1$ ,  $\lambda(t)$  reaches its maximum at  $t = \left(\frac{c-1}{bc}\right)^{\frac{1}{c}}$ .

It is proposed to use a modified maximum likelihood estimator and Particle Swarm Optimization algorithm to estimate the parameters of this process.

**3.1 Modified Maximum Likelihood Estimation (MMLE)**

Another common classification of estimators is the maximum likelihood estimator (MLE) which is a common efficient estimator to estimate the parameters of the distribution given the data. This method also determines the maximum likelihood estimates of the parameter values, that is, the probability of obtaining the observed data when a particular set of parameter values are assumed [7]. Nevertheless, it has been reported that MLE may in some circumstances give non-biased parameter estimates, which could be more realistic should the sample size be small or the distribution of the population be complicated. It is also noted that in such cases, some changes can be made to the MLE to try enhance the quality of the estimates [17].

MMLE was intended as an extension of the MLE method for estimating the GGP parameters since the estimators for this process cannot be easily derived via the MLE method, thus yielding more reliable estimations of the model parameters.

All in all, MMLE is a versatile modification of MLE good for its application in improving the relative error of the estimated parameters when MLE cannot be used.

Suppose that  $\{N(t), t \geq 0\}$  is a GGP with the occurrences rate defined by the formula (3), then the joint probability function of the times of occurrence  $(t_1, t_2, \dots, t_n)$  where  $(0 < t_1 \leq t_2 \leq \dots \leq t_n \leq t_0)$  defined by [2, 3]:

$$f_n(t_1, t_2, \dots, t_n) = \prod_{i=1}^n \lambda(t_i) e^{-m(t_0)} \tag{4}$$

Therefore, the GGP parameter represents the mean occurrences rate time, as:

$$m(t) = \int_0^t \lambda(u) du$$

$$\begin{aligned}
 &= \int_0^t (abce^{-bu^c} u^{c-1}) du \\
 &= a[1 - e^{-bt^c}]
 \end{aligned} \tag{5}$$

Hence, likelihood function of the GGP for the time period  $(0, t]$  with the intensity function  $\lambda(t)$  is:

$$L = \prod_{i=1}^n (abce^{-bt^c} t^{c-1}) e^{-a[1-e^{-bt^c}]} \tag{6}$$

To estimate the process parameters using the maximum likelihood method, we start by taking the natural logarithm of formula (6) to obtain the log-likelihood function given by equation (7):

$$\ln L = n \ln a + n \ln b + n \ln c - b \sum_{i=1}^n t_i^c + \sum_{i=1}^n \ln t_i^{c-1} - a(1 - e^{-bt_0^c}) \tag{7}$$

To estimate the values of the parameters  $a, b$ , and  $c$ , we derive equation (7) with respect to each parameter and set the resulting derivative equal to zero. This gives us the following system of equations:

$$\begin{aligned}
 \frac{\partial \ln L}{\partial a} &= \frac{n}{a} - (1 - e^{-bt_0^c}) \\
 \hat{a} &= \frac{n}{1 - e^{-\hat{b}t_0^{\hat{c}}}}
 \end{aligned} \tag{8}$$

$$\begin{aligned}
 \frac{\partial \ln L}{\partial b} &= \frac{n}{b} - \sum_{i=1}^n t_i^c - a e^{-bt_0^c} t_0^c \\
 \hat{b} &= \frac{n}{\sum_{i=1}^n t_i^{\hat{c}} + \hat{a} e^{-\hat{b}t_0^{\hat{c}}} t_0^{\hat{c}}}
 \end{aligned} \tag{9}$$

$$\begin{aligned}
 \frac{\partial \ln L}{\partial c} &= \frac{n}{b} - b \sum_{i=1}^n t_i^c \ln t_i + \sum_{i=1}^n \ln t_i^c \ln(\ln t_i) - a e^{-bt_0^c} b t_0^c \ln(\ln t_0) \\
 \hat{c} &= \frac{n}{\hat{b} \sum_{i=1}^n t_i^{\hat{c}} \ln t_i - \sum_{i=1}^n \ln t_i^{\hat{c}} \ln(\ln t_i) + \hat{a} e^{-\hat{b}t_0^{\hat{c}}} b t_0^{\hat{c}} \ln(\ln t_0)}
 \end{aligned} \tag{10}$$

where  $L, n$  and  $t_i$  are respectively the log-likelihood function given by equation (6), the total number of failure and occurrence times. These equations can be estimated by using iterative techniques like Newton-Raphson or Expectation- Maximization algorithms to compute  $a, b$  and  $c$ , which can give maximum likelihood values [8, 12, 14]. We observed that solving the system of equations which is obtained from differentiation of the equation (7) with respect to "a", "b", and "c" is not at all possible using common algebraic solutions because of the fact that the system is heavily nonlinear. Thus, herein, we suggest the new approach based on the maximum likelihood method modified by the incorporation of one of the most effective artificial intelligence methods: particle swarm optimization (PSO).

### 3.2 Particle Swarm Optimization (PSO)

PSO stands for Particle Swarm Optimization and is an optimization technique that was conceptualized to mimic socially engaging behavior. It was identified by Eberhart and Kennedy in 1995, and has been applied in a number of disciplines including civil engineering, finance, and computer science to name but a few, to solve optimization. It begins with a population of candidate solutions called particles and moves in the search space relying on the best particle discovered so far. It is governed by their current positions, velocities as well as by the acceleration coefficients and is updated on each iteration of the algorithm [1].

The PSO algorithm works to find the best solution, which is done by modifying the particle's position and velocity in the whole swarm. Stopping an algorithm is related to the stopping criteria which include reaching to the maximum number of iterations as well as getting a satisfactory solution. The main strengths of PSO include the simplicity of the algorithm, as well as the relative ease with which it can be implemented; also, for PSO, both continuous and discrete optimization problems can be solved.

The PSO algorithm is a metaheuristic optimization technique that can be used to find the optimal solution to an optimization problem. The algorithm starts by initializing a population of candidate solutions called particles. Each particle has a position and a velocity in the search space. The algorithm then evaluates the fitness of each particle using a fitness function that measures how well the particle satisfies the objective of the optimization problem. It updates the position and velocity of each particle at each iteration of the algorithm based on its own experience and the experience of the other particles in the swarm. The updated equations for the position and velocity of a particle are as follows [9]:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (P_{best} - X_i^t) + c_2 r_2 (g_{best} - X_i^t) \quad (11)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (12)$$

with  $X_i^0 \sim U(X_{Min}, X_{Max})$ ,

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

where;  $V_i^{t+1}$  is the velocity of particle  $i$  at iteration  $k$ ,  $\omega$  is a positive constant, referred to as inertia weight,  $c_1$  and  $c_2$  are the acceleration coefficients,  $P_{best}$  is the personal best position of the particle  $i$ ,  $g_{best}$  represents the global best position obtained up. The dimensions of inertia weight determine how much the swarm's search is focused on exploitation of a particular part of the search space or how much the swarm is just looking for new areas to explore. In most cases, it is designed to be defined as reducing from an initial value to a final value in the algorithm. Acceleration coefficients regulate interaction of the personal and global best positions with movements of the particles. PSO algorithm keeps on adjusting the position and velocity of each particle until a stopping condition is reached which may include a maximum number of iterations has elapsed, or a good solution is reached or obtained. Each particle is consists of 3-vector [5]:

- X-vector: is a vector for the particle in the search space to show its current position for the particle.
- V vector: this vector defines the velocity of the particle; this velocity decides the path and movement of the particle in the search space.
- P-vector ( $P_{best}$ ): stores the coordinates of the improving position averagely achieved by the particle while in movement in the space of the search.

In PSO, the fitness function (the objective function) is simultaneously used to assess (evaluated), at the same time particles are initialized and then computed; after which, a loop is begun to search for an optimum solution, in which at that time personal best (this is the best value for each particle) and the global best value (this is the best value for the whole swarm of particles) are determined. When a loop begins the velocities of particles are first updated by the personal and global best values Then the updating of the position of each particle is done by the current velocity of the particle. Finally, the loop is terminated by use of a stopping criterion which is predetermined [16].

The fitness is calculated during the initial phase of PSO where a population of particles are randomly generated in the search space. This position is then used to compute the bead personal best position for each particle and the PSO global best position of the established swarm. Within the iteration process velocity of each particle is proposed taking into consideration the personal as well as the global best position and velocity of the particle. This is followed by a movement of the position of each particle by the steps of its new velocity. This procedure is done for a fixed number of steps or until a stop criterion is met which may include fitness level achievement or reaching a maximum number of steps [6,11].

Therefore, the steps of PSO algorithm can be described as follows:

1. Initiation process should commence with position initialization and sometime the particles are randomly positioned.
2. Assess the quality of a particular particle based on the measures of its fitness function.
3. The best position must be updated until you find a new position that is local and better of the previous one.
4. If the current position is better than the previous best position found so far, then update the global best position.
5. Use the formula given by equation (11) to determine the velocity of each particle.
6. To update the position of the particle, we shall use equation (12).  
Fill steps (2-6) until the termination criteria is attained.

### 3.3 Parameter Estimation for GGP

In this section, we will describe two algorithms to estimate the parameters of the GGP using two different methods. The first method is the MMLE, which incorporates the PSO algorithm with the maximum likelihood method, and the second method is the PSO algorithm directly for estimation. The algorithms are described as follows:

#### Algorithm (1): MMLE (MLE-PSO) method

- 1) For the GGP, we also have the data likelihood function given data set D.
- 2) Take the natural logarithm of the likelihood function of GGP.

- 3) Substituting the resulting equations obtained by differentiating of equation (7) with respect to  $a, b$ , and  $c$  with PSO algorithm.
- 4) Initialize PSO parameters, including the population size ( $N = 50$ ), maximum number of iterations ( $i_{max} = 100$ ), PSO constants, the acceleration coefficients  $C_1 = C_2 = 1$ , ( $r_1 = r_2 = 0.1$ ); note that the maximum and minimum values for inertial weight is: Different with equation (3), the maximum value of transfer entropy is  $\theta_{max} = 0.9$ ,  $\theta_{min} = 0.4$ .
- 5) Randomly position and integrate the particle's velocities at the start of evolution, or generate the first population of particles.
- 6) For each particle in the population, calculate the fitness value which is the negative log-likelihood function.
- 7) Replace personal best positions and velocities of each particle by the result of the fitness function.
- 8) Update the global best position of the population along with the global best velocity of the population.
- 9) Then, apply the PSO algorithm in updating the positions and the velocities of each particle.
- 10) Calculate the health of all the new position in the landscape in terms of the fitness function.

**Algorithm (2): PSO method**

- 1) Initialize PSO parameters, including the population size ( $N = 50$ ), maximum number of iterations ( $i_{max} = 100$ ), PSO constants, the acceleration coefficients  $C_1 = C_2 = 1$ , ( $r_1 = r_2 = 0.1$ ); note that the maximum and minimum values for inertial weight is:  $\theta_{max} = 0.9$ ,  $\theta_{min} = 0.4$ .
- 2) Propose an initial state and, in particular, randomly place and assign velocities to particles.
- 3) Evaluate the fitness function for each particle in the population, which is maximum percentage error (MPE), according to the formula below:

$$MPE = \sum_{1 \leq i \leq n}^{max} \left[ \frac{|S_i - \hat{S}_i|}{S_i} \right] \tag{13}$$

where  $S_i = \sum_{j=1}^i X_j, \hat{S}_i = \sum_{j=1}^i \hat{X}_j$ .

- 4) Update the personal best positions and velocities of each particle according to the fitness function of the used problem.
- 5) Record the global best position and velocity of the population according to an individual's personal best of position and velocity.
- 6) The PSO algorithm has to be used to update the positions and velocities of each particle present in the environment.
- 7) Assessing the new positions means that the fitness of such positions must be evaluated to determine the possibility of accepting those new positions.
- 8) If it is however reached then return the best solution found in the process of carrying out the search. If not go to the next step 4.

**4. Results and Discussion**

**4.1 Simulation**

In this section, a comprehensive simulation study is conducted to compare two estimation methods in order to arrive at the best estimate of the required parameter. The five stages that follow are an explication of the design simulation experiments.

Stage 1 for the simulation is all about generating data. In the course of this work, a simulated data values are created with the use of GGP distribution with known parameters. The obtained data will be employed to assess the degree of the correspondence between the methods of estimation used in the study. The number of observations and values which are employed in the process of simulation can be defined referring to the features of the studied process. It is advised to target an  $n$  sufficiently high so as to have good estimation of the parameters which is largely determined by the computations in the case of a large data sample.

MMLE (MLE-PSO) Algorithm on Stage 2: Estimating Parameters. However, in this stage, no matter how wide the sense is, the authors propose the MMLE method with PSO algorithm to estimate the parameters of the GGP distribution. This is done several times and the final parameter estimates are recorded after the run of the algorithm. The MPE and RMSE are computed for the estimates of all sets of parameters.

Stage 3: Estimation of parameters using PSO algorithm directly. Finally in this stage, we use PSO right on the model to estimate the parameters of the GGP distribution. The algorithm is operated many times, and the parameter estimates that the algorithm produces at each trial run are noted. RMSE values are determined for all kinds of the parameter estimations.

The fourth stage is the comparing of the estimation methods after the four estimation techniques have been applied to a specific problem. In this stage, RMSE values obtained from MMLE and PSO algorithm directly are compared to select the best estimation method, according to the following form:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{\theta}_i - \theta)^2} \tag{14}$$

In general, the simulation study provides researchers with a convenient tool with which they can compare the relative performance of a number of estimation methods in order to select the method that yields the most accurate parameter estimates. The above selected method can then be employed to estimate the parameters of the GGP distribution for real data in organizations, analysis and decision making.

Table 1: RMSE of estimated parameters for simulated GGP using PSO and MMLE methods with sample sizes of  $n = 50, 100$ .

Sample size	parameters	Method	RMSE ( $\hat{a}$ )	RMSE ( $\hat{b}$ )	RMSE ( $\hat{c}$ )
50	$\{a = 0.75, b = 0.5, c = 0.75\}$	PSO	0.0292	0.0261	0.0659
		MMLE	0.0111	0.0250	0.0654
	$\{a = 0.75, b = 0.75, c = 0.5\}$	PSO	0.0156	0.0603	0.0603
		MMLE	0.0101	0.0433	0.0594
	$\{a = 0.75, b = 0.75, c = 0.75\}$	PSO	0.0620	0.0624	0.0615
		MMLE	0.0582	0.0286	0.0505
100	$\{a = 0.75, b = 0.5, c = 0.75\}$	PSO	0.0207	0.0452	0.0466
		MMLE	0.0079	0.0247	0.0462
	$\{a = 0.75, b = 0.75, c = 0.5\}$	PSO	0.0110	0.0426	0.0427
		MMLE	0.0071	0.0306	0.0420
	$\{a = 0.75, b = 0.75, c = 0.75\}$	PSO	0.0438	0.0441	0.0435
		MMLE	0.0412	0.0202	0.0357

Table 1 reports on a simulation study undertaken to fit random variables to the GGP distribution having different sample sizes ( $n = 50, 100$ ) and a different set of parameters  $a, b$  and  $c$ . Two estimation procedures, the PSO algorithm and the proposed MMLE estimators were compared based on the Root Mean Squared Error (RMSE) of the parameters' estimates. The following value of RMSE was obtained where a small value of RMSE indicates better performance of the estimation method. The table summarizes the results in terms of the RMSE of the estimated parameters of the model by the different methods under different sample size.

For every combination of the samples size and parameters values the table shows the RMSE of the estimated parameters  $a, b$  and  $c$  defined with the PSO and MMLE. The results demonstrate that, on the whole, MMLE yields smaller RMSE than PSO and these improvements augmented with the sample size.

#### 4.2 Application for Real Data

In this paragraph, the application was applied to real data that characterized the outage of electric power generation units within days of Mosul gas station in north Nineveh Governorate for the period from 01/05/2019 to 30/06/2021. The first test implies confirmation that the data used is suitable for the process being observed. This is done by analyzing actual data and by using logarithmic scales to construct the number of cumulative days with power outages. This kind of data distribution configuration means that the majority of these points are arranged in a straight line. Hence, the collection and the details contained therein conform to the rate of occurrence of the GGP. Using MATLAB\R2017b language, the following figure was obtained:

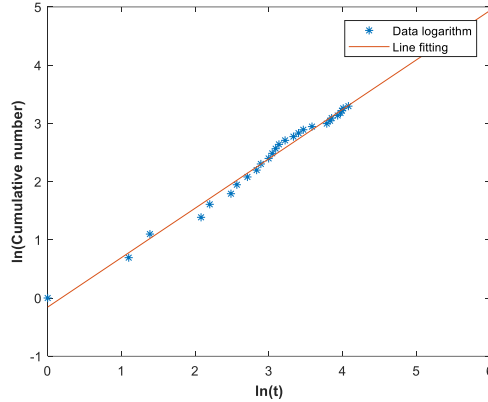


Figure 1: The graph for testing the suitability of the power outage data under study for the GGP.

Figure 1 shows cumulative number of outages in days with times of occurrence on a logarithmic scale for the data under study. It is noted that the scatterplot shows that there is a linear behaviour of the data, and then the possibility of modelling such data with GGP.

To evaluate the performance of the proposed GGP estimation methods, the daily outage rate of the electrical power generation units during the study period was estimated using MATLAB/R2017b language. The expected number of downtimes for the electrical unit was obtained and we calculated the RMSE between the real and estimated values. The results of this analysis are shown in the table below.

Table 2: RMSE values obtained from different methods used to estimate the time rate of occurrence of GGP.

Method	RMSE
MMLE	0.5958*
PSO	0.7454

This has shown in table 2, that of all the two methods that was used in estimating the occurrence rate time for the GGP process that the MMLE method has a lower RMSE value than the PSO method. This implies that the proposed MMLE method is more efficient in estimating the parameters of GGP process than the other methods. The figure below shows the estimated time rate functions of GGP using both MMLE and PSO methods, compared to the cumulative real data values representing the power outage of the electric generation units at the Mosul gas station in northern Nineveh Governorate:

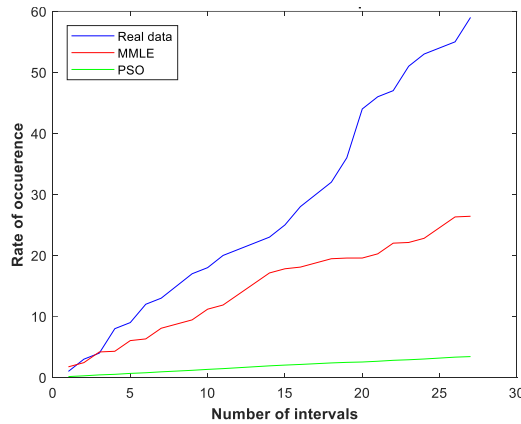


Figure 2: Estimates of the cumulative time rate of power outages for generation units compared to real data.

The evidence shown in figure 2 uses the time rate functions of the GGP where the MMLE outperforms the PSO in the estimation of the time rate functions by being nearer to the actual data.

## **5. Conclusion**

The objective of the study is to evaluate the GGP process parameters using the new MMLE method which integrates the PSO and MLE procedures together with the direct PSO procedure. The two methods were also analyzed with respect to its accuracy through simulation using root mean square error (RMSE). The simulation studies also revealed that use of MMLE approach yielded higher accuracy as compared to the PSO algorithm for most of the sample size under investigation especially for the larger sample size values. Furthermore, the proposed method was also applied on the real cumulative data of power outages of the electricity generation units in Mosul gas station in the north of Nineveh Governorate. From the results in the table, we observed that the results from the applied method which employed the MMLE are also similarly good as the simulated results and therefore we conclude that MMLE gave a relatively better estimation of the GGP parameters when compared to the employed PSO algorithm.

## **Acknowledgment**

The author is very grateful to the University of Mosul, College of Computer Science and Mathematics, for providing the necessary facilities to make this work of a higher standard.

## **Conflict of interest**

The author has no conflict of interest.

## **References**

1. Abd Latiff, I. and M. Tokhi. (2011). Improving particle swarm optimization convergence with spread and momentum factors. *International Journal of Computer Sciences and Engineering Systems*. 5(3), 209-217.
2. Erto, P., M. Giorgio, and A. Lepore. (2018). The generalized inflection S-shaped software reliability growth model. *IEEE Transactions on Reliability*. 69(1), 228-244.
3. Goel, A.L. and K. Okumoto. (1979). Time-dependent error-detection rate model for software reliability and other performance measures. *IEEE transactions on Reliability*. 28(3), 206-211.
4. Hossain, S.A. and R.C. Dahiya. (1993). Estimating the parameters of a non-homogeneous Poisson-process model for software reliability. *IEEE Transactions on Reliability*. 42(4), 604-612.
5. Jain, N., U. Nangia, and J. Jain. (2018). A review of particle swarm optimization. *Journal of The Institution of Engineers (India): Series B*. 99, 407-411.
6. Júnior, S.F.A.X., et al. (2020). An application of Particle Swarm Optimization (PSO) algorithm with daily precipitation data in Campina Grande, Paraíba, Brazil. *Research, Society and Development*. 9(8), e444985841-e444985841.
7. Lavanya, G., et al. (2017). Parameter estimation of the goel-okumoto model by comparing aco with mle method. *International Research Journal of Engineering and Technology*. 4(3), 1605-1615.
8. Nagar, P. and B. Thankachan. (2012). Application of Goel-Okumoto model in software reliability measurement. *Int. J. Comp. Appl. Special Issue ICNICT*. 5, 1-3.
9. Sayah, S. and A. Hamouda. (2013). A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems. *Applied soft computing*. 13(4), 1608-1619.
10. Sheta, A. Reliability growth modeling for software fault detection using particle swarm optimization. in 2006 IEEE International Conference on Evolutionary Computation. 2006. IEEE.

11. Voda, V.G. (2007). A new generalization of Rayleigh distribution. *Reliability: Theory & Applications*. 2(2 (6)), 47-56.
12. Yaghoobi, T. (2020). Parameter optimization of software reliability models using improved differential evolution algorithm. *Mathematics and Computers in Simulation*. 177, 46-62.
13. Yan, T., Nonhomogeneous poisson process models with a generalized bathtub intensity function for repairable systems. 2019, Ohio University.
14. Yin, L. and K.S. Trivedi. Confidence interval estimation of NHPP-based software reliability models. in *Proceedings 10th International Symposium on Software Reliability Engineering (Cat. No. PR00443)*. 1999. IEEE.
15. Yuan, T., T. Yan, and S.J. Bae. (2021). Superposed Poisson process models with a modified bathtub intensity function for repairable systems. *IISE Transactions*. 53(9), 1037-1051.
16. Zhang, Y., S. Wang, and G. Ji. (2015). A comprehensive survey on particle swarm optimization algorithm and its applications. *Mathematical problems in engineering*. 2015.
17. Zhao, M. and M. Xie. (1996). On maximum likelihood estimation for a general non-homogeneous Poisson process. *Scandinavian journal of statistics*. 597-607.

## توظيف طرائق ذكائية لتقدير معالم عملية كول المعممة المقترحة

مثنى صبحي سليمان

قسم الإحصاء والمعلوماتية، كلية علوم الحاسوب والرياضيات، جامعة الموصل، الموصل، العراق.

**الخلاصة:** ستقدم هذه الورقة توزيع Goel كمعدل حدوث لعملية بواسون غير المتجانسة (NHPP) لتحسين معدل حدوثها، وتم اقتراح تسميتها بعملية Goel المعممة (GGP). أما بالنسبة لتقدير معالم هذه العملية، فقد تمت مناقشة عدد من الطرق، واقترح استخدام مقدر الإمكان الأعظم (MLE) وبعد ذلك - كان من الضروري تعديل هذه الطريقة نظرًا لحقيقة أنه كان من المستحيل العثور على مقدرين يستخدمونها. تمت إضافة خوارزمية ذكية لدالة الامكان مع المعلمة وتم تسميتها باسم مقدر الإمكان الأعظم المعدلة (MMLE). ثم تمت مقارنة MMLE بطريقة ذكية أخرى وهي تحسين سرب الجسيمات (PSO) في تقدير معدل حدوث عملية Goel المقترحة لتحديد أفضل مقدر للعملية. بالإضافة إلى ذلك، تحتوي الورقة على محاكاة للعملية المذكورة ومثال على استخدامها العملي. وأظهرت نتائج المحاكاة والتطبيق أن أسلوب MMLE أعطى تقديرات ذات دقة أعلى من خوارزمية PSO لغالبية أحجام العينات المدروسة، وخاصة للأحجام الأكبر.

**الكلمات المفتاحية:** عملية كول المعممة، تقدير المعالم، مقدر الإمكان الأعظم، PSO، المحاكاة.