International Journal of Reliability and Applications
Vol. 13, No. 2, pp. 91-104, 2012

# Parameters estimation of the generalized linear failure rate distribution using simulated annealing algorithm

**Ammar M. Sarhan***
*Department of Mathematics and Statistics,*
*Faculty of Science, Dalhousie University, Halifax NS B3H 4R2, Canada*

**A. A. Karawia**
*Computer Science Unit, Deanship of Educational Services,*
*Qassim University, Buraidah 51452, Saudi Arabia*

**Abstract.** Sarhan and Kundu (2009) introduced a new distribution named as the generalized linear failure rate distribution. This distribution generalizes several well known distributions. The probability density function of the generalized linear failure rate distribution can be right skewed or unimodal and its hazard function can be increasing, decreasing or bathtub shaped. This distribution can be used quite effectively to analyze lifetime data in place of linear failure rate, generalized exponential and generalized Rayleigh distributions. In this paper, we apply the simulated annealing algorithm to obtain the maximum likelihood point estimates of the parameters of the generalized linear failure rate distribution. Simulated annealing algorithm can not only find the global optimum; it is also less likely to fail because it is a very robust algorithm. The estimators obtained using simulated annealing algorithm have been compared with the corresponding traditional maximum likelihood estimators for their risks.

**Key Words:** *Maximum likelihood method, Monte Carlo simulation method, statistical inference*

## 1. INTRODUCTION

Recently, Sarhan and Kundu (2009) proposed a new three-parameter distribution named as generalized linear failure rate distribution (GLFR). The GLFR distribution generalizes several well known distributions such exponential, Rayleigh, linear failure rate, generalized exponential and generalized Rayleigh distributions. The hazard rate function of the GLFR distribution takes different shapes which makes it flexible and can fit very well a wide range of survival data sets.

*Corresponding Author.
E-mail address*: asarhan@mathstat.dal.ca*

Successful application of the generalized linear failure rate distribution requires having acceptable point estimates of its three unknown parameters. In estimation theory and data analysis, global optimization of the log-likelihood function is a cornerstone in the maximum likelihood method. The maximum likelihood estimates (MLE) of the parameters are those values that maximize the likelihood function. The maximum likelihood method has very desirable properties. In some cases when the number of unknown parameters is bigger than or equal two, the optimization problem becomes difficult to solve analytically. In such cases, numerical techniques are required. Sarhan and Kundu (2009) used the traditional gradient technique to derive the maximum likelihood estimation (MLE) of the different parameters of the GLFR distribution and discussed some of the testing of hypothesis problems. The traditional gradient technique might only work under special case where the gradients of the functions to be maximized are defined at all points. This means that they could be gradient sensitive. The most important fact regarding the traditional gradient technique is that it is a local optimization algorithm and therefore, when we use it to find the global optimization, we would most likely end up getting a local optimum depending on the initial guess for the distribution parameters at the beginning of the algorithm execution. That was what Sarhan and Kundu (2009) faced when they used the traditional gradient technique to get the MLEs of the parameters. Cramer (1986) and Finch et al. (1989) list some of ”unpleasant possibilities” for the traditional gradient technique of the maximum likelihood method. Since the traditional gradient technique is highly sensitive to the initial guess and is capable only of local optimization, it is necessary to look out for other technique which is capable of finding the global optimization and is relatively insensitive to the user’s initial guess.

One of the more useful algorithms, easy to implement and a robust technique for global optimization of the log-likelihood function (as a multi-dimensional function) is Simulated Annealing Algorithm (SAA), Brooks et al. (1995). SAA is capable of efficiently finding the global optimum of any *m*-dimensional log-likelihood function ($m \geq 1$). Also, SAA is an easy approach to use to optimize the log-likelihood function. It explores the function’s entire surface and tries to optimize the function while moving both uphill and downhill. Thus, it is largely independent of the starting values, often a critical input in conventional algorithms. Further, it can escape from local optima and go on to find the global optimum by the uphill and downhill moves. Simulated annealing algorithm also makes less stringent assumptions regarding the function than do conventional algorithms (it does not need even to be continuous). Because of the relaxed assumptions, it can more easily deal with functions that have ridges and plateaus. It is worthwhile to mention that SAA can be used to determine the optimal set of parameters for any distribution given its ability to perform global optimization for any multidimensional log-likelihood function. This means that SAA works well for any distribution. Tan and Raghavan (2008) used SAA to globally maximize the multidimensional log-likelihood function to determine the MLEs of a mixture distribution parameters. Finally, it can optimize functions that are not defined for some parameter values. In optimization literatures, the function to be optimized is usually referred to as the objective function (Brooks and Morgan, 1995). The same convention will be followed here and our log-likelihood function will be called as the objective function.

The main aim of this paper is to use the SAA to globally maximize the three-

dimensional log-likelihood function which will in turn help determine the optimal values (MLEs) of the GLFR distribution parameters. The obtained maximum likelihood estimates are compared with the traditional ones obtained using gradient method through numerical simulations.

The rest of this paper is organized as follows. Section 2 presents a brief prescription of the GLFR distribution. The likelihood function and likelihood equations required to derive the maximum likelihood estimators of the GLFR distribution are presented in Section 3. Section 4 introduces the general simulating annealing algorithm. Applying simulated annealing algorithm to parameter estimation is presented in Section 5. Simulation study and comparisons between the traditional maximum likelihood method and the proposed algorithm are given in Section 6. Finally, Section 7 concludes the paper.

## 2. THE GENERALIZED LINEAR FAILURE RATE DISTRIBUTION

The cumulative distribution function of the generalized linear failure rate distribution with three parameters, denoted by GLFRD(α,β,γ), takes the form, Sarhan and Kundu (2009),

$$F(t;\alpha,\beta,\gamma) = [1 - e^{-\left(\alpha t+\frac{\beta}{2}t^2\right)}]^{\gamma},\ t \geq 0, \gamma > 0. \tag{2.1}$$

where α and β ≥ 0 such that α+β > 0.

The probability density function (pdf) and the failure function of GLFRD(α,β,γ) are given respectively by

$$f(t;\alpha,\beta,\gamma) = \gamma(\alpha+\beta t)\left[1 - e^{-\left(\alpha t+\frac{\beta}{2}t^2\right)}\right]^{\gamma-1} e^{-\left(\alpha t+\frac{\beta}{2}t^2\right)},\ t \geq 0, \tag{2.2}$$

and

$$h(t;\alpha,\beta,\gamma) = \frac{\gamma(\alpha+\beta t)[1-e^{-\left(\alpha t+\frac{\beta}{2}t^2\right)}]^{\gamma-1}e^{-(\alpha t+\frac{\beta}{2}t^2)}}{1-[1-e^{-(\alpha t+\frac{\beta}{2}t^2)}]^{\gamma}}. \tag{2.3}$$

Figures 1 provides the pdf and the failure rate functions of GLFRD(α,β,γ) for different parameter values.

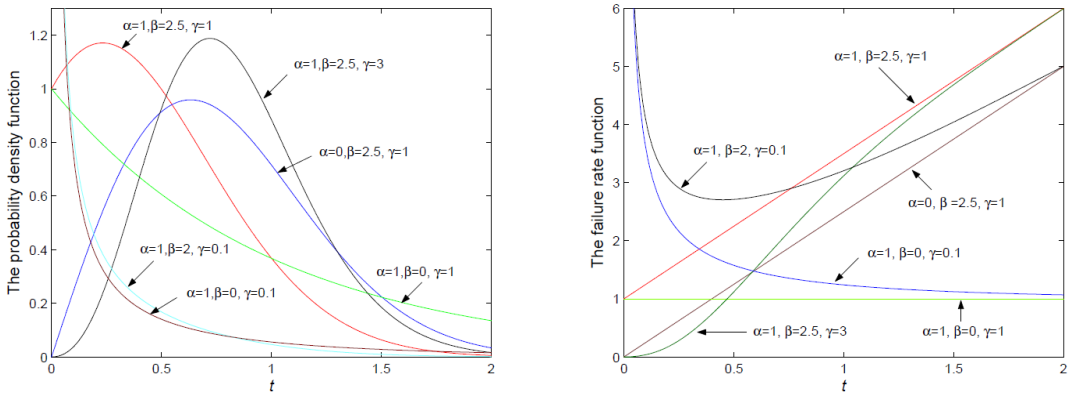


**Figure 1.** The pdf and failure rate function of GLFRD(α,β,γ)

From Figure 1, it is immediate that the pdf can be decreasing (right skewed) or unimodal and the failure rate function can be increasing, decreasing, or bathtub shaped. It is immediate that from GLFRD(α,β,γ), the following special cases can be derived: (*i*) Linear failure distribution LFRD(α,β), when γ = 1; (*ii*) Generalized exponential distribution GE(α,γ), when β= 0; (*iii*) Generalized Rayleigh distribution GRD(β,γ), when α= 0. One can easily verify that: (*i*) if γ = 1, the failure rate function of the GLFR distribution is either increasing (if β> 0) or constant (if β= 0 and α > 0); (*ii*) when γ > 1, the failure rate function should be increasing only; and (*iii*) if γ < 1, then the failure rate function will be either decreasing if β = 0 or a bathtub shaped if β > 0.

Gupta and Gupta (2007) observed that the reversed hazard function plays an important role in the reliability analysis. The reversed hazard function of the GLFRD(α,β,γ) is

$$r(t;\alpha,\beta,\gamma)=\frac{f(t;\alpha,\beta,\gamma)}{F(t;\alpha,\beta,\gamma)}=\gamma\frac{(\alpha+\beta t)e^{-(\alpha t+\frac{\beta}{2}t^2)}}{1-e^{-(\alpha t+\frac{\beta}{2}t^2)}}=\gamma\frac{f(t;\alpha,\beta,1)}{F(t;\alpha,\beta,1)}=\gamma r(t;\alpha,\beta,1). \quad (2.4)$$

It is well known that the hazard function or the reversed hazard function uniquely determines the corresponding probability density function. From (2.4) it is clear that the GLFRD(α,β,γ) is a proportional reversed hazard family.

## 3. MAXIMUM LIKELIHOOD ESTIMATORS

In this section, we consider the traditional maximum likelihood estimators (MLEs) of GLFRD distribution parameters. Let us assume that we have a simple random sample $\underline{t}=t_1,t_2,\dots,t_n$ from GLFRD(α,β,γ). Using (2.2) the likelihood function of this sample is

$$L(\alpha,\beta,\gamma|\underline{t})=\prod_{i=1}^{n}f(t_i;\alpha,\beta,\gamma)=\gamma^n e^{-(\alpha T_1+\beta T_2)}\prod_{i=1}^{n}\left\{(\alpha+\beta t_i)[1-e^{-(\alpha t_i+\frac{\beta}{2}t_i^2)}]^{\gamma-1}\right\} \quad (3.1)$$

where

$$T_j=\frac{1}{j}\sum_{i=1}^{n}t_i^j,\ j=1,2.$$

The log-likelihood function becomes

$$\mathcal{L}=n\ln\gamma-\alpha T_1-\beta T_2+\sum_{i=1}^{n}\ln(\alpha+\beta t_i)+(\gamma-1)\sum_{i=1}^{n}[1-e^{-\left(\alpha t_i+\frac{\beta}{2}t_i^2\right)}]. \quad (3.2)$$

The maximum likelihood point estimates of α,β,γ can be derived by maximizing the likelihood (3.1) or the log-likelihood function (3.2). This can be done by using ordinary optimization technique. Gradient method depends on setting the partial derivatives of the $\mathcal{L}$ with respect to the three parameters equal zero to get the following system of non-linear likelihood equations in the three parameters

$$\frac{\partial\mathcal{L}}{\partial\alpha}=-T_1+\sum_{i=1}^{n}\frac{1}{(\alpha+\beta t_i)}+(\gamma-1)\sum_{i=1}^{n}\frac{t_i e^{-(\alpha t_i+\frac{\beta}{2}t_i^2)}}{1-e^{-(\alpha t_i+\frac{\beta}{2}t_i^2)}}=0, \quad (3.3)$$

$$\frac{\partial\mathcal{L}}{\partial\beta}=-T_2+\sum_{i=1}^{n}\frac{t_i}{(\alpha+\beta t_i)}+\frac{1}{2}(\gamma-1)\sum_{i=1}^{n}\frac{t_i^2 e^{-(\alpha t_i+\frac{\beta}{2}t_i^2)}}{1-e^{-(\alpha t_i+\frac{\beta}{2}t_i^2)}}=0, \quad (3.4)$$

$$\frac{\partial\mathcal{L}}{\partial\gamma}=\frac{n}{\gamma}+\sum_{i=1}^{n}\ln[1-e^{-\left(\alpha t_i+\frac{\beta}{2}t_i^2\right)}]=0. \quad (3.5)$$

The above system of non-linear equations does not have explicit solution and then we have to solve it numerically. This is the most common problem when the number of unknown parameters to be estimated exceeds two (in some cases equals two). To use a numerical technique to solve that system, it will be needed to evaluate the gradient terms and the objective function $\mathcal{L}$ itself at different points the numerical technique needs. This is, not an easy task in addition to that it is quite boring to derive the gradient of this complicated $\mathcal{L}$. This motivates why we looked for another method to be used instead of the traditional numerical techniques which is efficient and easy to implement. In the following sections, we discuss the simulating annealing algorithm and apply it to calculate the MLE of the three unknown parameters of the GLFR distribution.

## 4. GENERAL SIMULATING ANNEALING ALGORITHM

Annealing is the physical process of heating and then cooling a substance in a controlled manner. The desired result is a strong crystalline structure. The structure in question is our encoded solution, and the temperature is used to determine how and when new solutions are accepted, Kirkpatrick et al. (1983).

The simulated annealing algorithm is very simple and can be defined in five steps (see Figure 2).

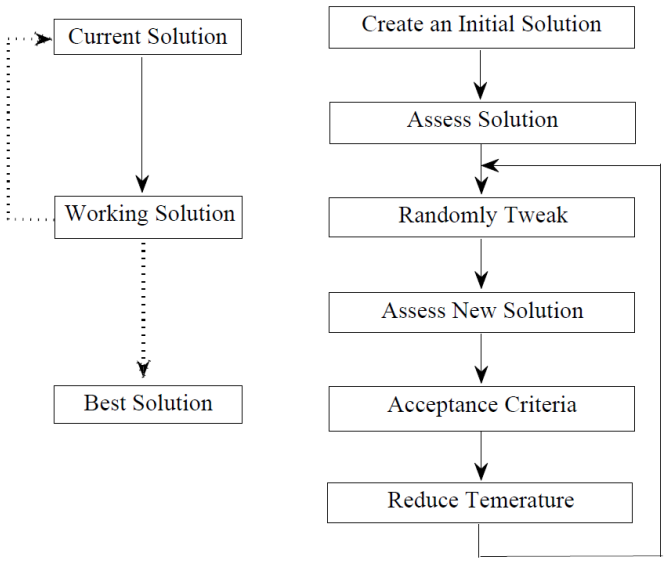


**Figure 2.** Simulating annealing algorithm

**Step 1: Initial Solution**

For most problems, the initial solution will be a random one. This is loaded into what is called the current solution. Another alternative is to load the initial solution with an existing solution, possibly one found in a previous run.

**Step 2: Assess Solution**

Assessing the solution consists of decoding the current solution and then performing whatever action is necessary to evaluate it against the given problem. The encoded solution may simply consist of a set of variables. These variables would be decoded from the current solution and the energy of the solution assessed based upon how well it solved the given problem.

**Step 3: Randomly Tweak Solution**

Tweaking the solution begins by copying the current solution into what is called the working solution. We then randomly modified the working solution. Once the working solution has been tweaked, we assess the solution as defined in the previous step. This random trail is based upon the Metropolis algorithm, Jones (2003).

**Step 4: Acceptance Criteria**

Now, we have two solutions. The first is our original solution called the current solution and the second is tweaked version called the working solution. Each has an associated energy, which is the strength of the solution. Our working solution is then compared to the current solution. If the working solution has less energy than the current solution then we copy the working solution to the current solution and move to temperature reduction. On the other hand, if the working solution is worse than the current solution, we evaluate the acceptance criteria to figure out what to do with the current working solution. The probability of acceptance is given by:

$$P = \exp\left(-\frac{\Delta}{T}\right) \tag{4.1}$$

where $\Delta$ is the increase in the objective function $g$ and $T$ is a control parameter, which by analogy with the original application is known as the system 'temperature' irrespective of the objective function involved.

**Step 5: Reduce Temperature**

After some number of iterations through the algorithm at this temperature, we reduce the temperature by a small amount. Large varieties of cooling schedules exist, but in this paper we will use a simple geometric function

$$T_{i+1} = \rho T_i \tag{4.2}$$

where$\rho < 0$ and constant. Typically, $0.75 \leq \rho \leq 0.95$.

A number of iterations will be performed at a single temperature. When that set of iterations is complete, the temperature is reduced and the process continues until the temperature reaches zero.

Ease of use and provision of good solutions to real-world problems makes this algorithm be one of the most powerful and popular meta-heuristics to solve many optimization problems.

The basic structure of simulated annealing algorithm is presented in Table 1, where the following notation is used:

$S$ = the current solution,

$S^*$= the best solution.

$S_i$ = neighboring solution,

$g(S)$ = the value of objective function at solution $S$,

$i$ = repetition counter,
$T_0$= initial temperature,
$T_f$= final temperature,
$I$ = number of repetition allowed at each temperature level,
$p$ = probability of accepting $Si$ when it is not better than $S$.

**Table 1.** Simulated annealing algorithm for minimization problem

Initialize the SAA control parameter $(T_0, I)$
Select an initial solution, $S_0$
Set $T = T_0$; Set $S = S_0$; Set $S^* = S_0$; Calculate $g(S_0)$;
While the stop criterion is not reached do:
    Set $i = 1$;
    While $i < I$ do:
        Generate solution $S_i$ in the neighborhood of $S_0$; Calculate $\Delta = g(S_i) - g(S)$;
        if $\Delta \leq 0$
            $S = S_i$
        else
            generate a random number, $r \in (0, 1)$
            if $(r \leq p = \exp(-\Delta/T)$;
                $S = S_i; i = i + 1$;
            end
        end
        if $(g(S) < g(S^*))$
            $S^* = S_i$;
        end
    end
    reduce the temperature $T$;
end

It is obvious that this procedure just takes into account the minimization problems, hence while performing a maximization problem, the objective function is multiplied by (−1) to obtain a capable form.

The algorithm starts with an initial solution for the problem. As it is obvious from Table 1, SAA has two cycles, inner and outer. In the inner cycle of the SA, repeated while $n < I$, a neighboring solution $S_i$of the current solution $S$ is generated. If $\Delta \leq 0$ ($S_i$ is better than $S$), then the generated solution replaces the current solution, otherwise the solution is accepted with a criterion probability. The value of the temperature, $T$, decreases in each iteration of the outer cycle of the algorithm.

As a meta-heuristic algorithm, the most important feature of this algorithm is the possibility of accepting a worse solution, hence allowing it to prevent falling into a local optimum trap. Obviously, the probability of accepting a worse solution decreases as the temperature decreases in each outer cycle. The performance of SAA depends on the definition of the several control parameters:

- The initial temperature $T_0$ should be high enough that in the first iteration of the algorithm, the probability of accepting a worse solution is, at least, of 80%, Kirkpatrick et al. (1983).
- The most commonly used temperature reducing function is geometric.
- The length of each temperature level *I* determines the number of solutions generated at a certain temperature *T*.
- The stopping criterion defines when the system has reached a desired energy level.

It is obvious that these control parameters are chosen with respect to the specific problem at hand. When adapting this general algorithm to a specific problem, the procedure to generate both initial and neighboring solutions is very important in addition to the control parameter. The details of proposed SAA to parameter estimation problem are presented in the next section.

## 5. APPLYING THE SAA TO PARAMETER ESTIMATION

To estimate the three parameters of generalized linear failure rate distribution, we need to maximize $L$ (or $\mathcal{L}$) using simulated annealing algorithm. In this regard, the steps of this algorithm are briefly given below.

**Algorithm 5.1** *To estimate the three parameters of GLFR distribution using SAA, we follow the following steps:*

---

***step 1:*** *Generate a random sample from the GLFR distribution with a large enough size.*
***step 2:*** *Determine control parameters of SAA, i.e.* $T_0$, $T_f$, $\rho$, *I.*
***step 3:*** *Generate random values for the three parameters* $\alpha, \beta, \gamma$.
***step 4:*** *Compute the log-likelihood function* $\mathcal{L}$ *at this randomly generated solution.*
***step 5:*** *While the stop criterion is not reached do:*
  ***step 5.1:*** *If* $T > T_0$ *then let* $T = \rho * T$
  ***step 5.2:*** *Generate neighboring values, say* $\alpha_0, \beta_0, \gamma_0$ *for* $\alpha, \beta, \gamma$.
  ***step 5.3:*** *Compute the log-likelihood function at this new solution, say* $L_0$.
  ***step 5.4:*** *If* $\mathcal{L}_0 > \mathcal{L}$ *then set* $\alpha = \alpha_0$, $\beta = \beta_0$, $\gamma = \gamma_0$, *and* $\mathcal{L} = \mathcal{L}_0$ .
  ***step 5.5:*** *Else*
    ***step 5.5.1:*** *Calculate* $\Delta = \mathcal{L} - \mathcal{L}_0$.
    ***step 5.5.2:*** *Generate a random value u from Uniform*(0, 1).
    ***step 5.5.3:*** *If* $u < \exp(-\frac{\Delta}{T})$ *then set* $\alpha = \alpha_0$, $\beta = \beta_0$, $\gamma = \gamma_0$.
***step 6:*** *Print* $\alpha, \beta, \gamma$ *and* $\mathcal{L}$.

---

In SAA, the temperature parameter controls the search for the optimum solution. The temperature parameter typically starts off high and is slowly lowered (cooled) in each iteration. At every iteration a new solution (point) is generated and its distance from the current solution is proportional to the temperature. If the new solution has a better function value it replaces the current solution and iteration counter is incremented. It is possible to accept and move forward with a worse solution. The probability of doing so depends on the temperature. This unintuitive step sometime helps identify a new search region in hope of finding a better maximum.

The initial temperature $T_0$ for optimizing the log-likelihood function has to be kept high and it could be set to the approximate range of variation of the log-likelihood function which can be determined by a random space search of the log-likelihood function value for different input parameter combination. It is important to mention here that a precise value for $T_0$ is not required. It is only needed to specify a plausible value for $T_0$ that would guarantee a successful SAA run and a good start for this would be to set $T_0$ to be the range of the log-likelihood function. It is advocated that a high initial temperature is essential for the SAA to achieve the global optimum value. However, setting too high initial temperature makes the SAA inefficient as it will take longer time to attain the global optimum. Very hight initial temperature values cause SAA less preferred as it causes slower processing speed, larger memory requirements and increased computational load, all of these are undesirable. There is no hard and fast rule for setting the initial temperature value. However, good judgment should be exercised when a user sets a value for it (Tan and Raghavan, 2008).

As an overall conclusion, if the control parameters of SAA are defined correctly, unbiased estimations with minimum variance will be straightforward.

## 6. SIMULATION STUDY

To illustrate the new approach, Monte Carlo simulation method is applied to present two examples. Furthermore, to know the effects of the sample size on the performance of the simulated annealing algorithm, samples of size 100*;* 250*;* 500 and 1000 have been considered and used to estimate the three parameters of the GLFR distribution. It is straightforward from the primary estimation theory that the bigger the sample size the better the estimation. However, here as the sample size increases the more complicated will be the log-likelihood function to maximize. Therefore, the selection of the sample size is a matter of compromise. For both two examples, the cooling rate has been considered as 0*:*80, the initial and final temperatures are 1 and 1 _ 10□8, respectively. To see how the algorithm approaches the maximum, and finally the estimations, its performance is illustrated for these two examples.

All codes has been done using Matlab R2008b, and been run on Intel(R) Core(TM) *i*5 CPU *M*430@2.27 GHz processor with 4*:*00 GB of RAM.

In the following two examples, the following scheme has been adopted:

1. Specify the values of the parameters $\alpha, \beta$, and $\gamma$, say$\alpha_0, \beta_0$, and $\gamma_0$, respectively.
2. Specify the sample size of random sample *n*.
3. Generate a random sample with size *n* from the GLFRD ($\alpha,\beta,\gamma$).
4. Using the generated sample obtained in step 3, estimate the three parameters $\alpha$, $\beta$, and $\gamma$ using traditional maximum likelihood method and Simulated Annealing Algorithm.
5. Repeat steps 3 and 4 *N*-times.
6. Compute the root mean squared error *RMSE* associated with each estimate of Every element of the vector of unknown parameters $\theta = (\theta_1, \theta_2, \theta_3) = (\alpha,\beta,\gamma)$ according to the following formula

$$RMSE_{\theta} = \sqrt{\frac{\sum_{i=1}^{N}(\hat{\theta}_j^{(i)} - \theta_{j0})^2}{N}}$$

where $\hat{\theta}_j^{(i)}$ is the estimated value of the parameter _*j* using the random sample generated in the iteration *i* and $\theta_{j0}$ is the true value of $\theta_j$.

7. Compute the mean of the estimated values obtained from the *N* iterations for every parameter $\theta_j$, $j = 1; 2; 3$, according to the following formula

$$\bar{\hat{\theta}}_j = \frac{\sum_{i=1}^{N} \hat{\theta}_j^{(i)}}{N}$$

8. Repeat the entire process for the sizes *n* = 100,250,500 and 1000 at number of iteration *N* = 10000.

**Example 1:** In this example it is assumed that $\theta = (\alpha,\beta,\gamma) = (2,1,2)$. Table 2 summarizes the mean value and the root mean squared error associated with the every estimator obtained using the traditional maximum likelihood estimators (MLE) and Simulated Annealing Algorithm (SAA) for all the three parameters. Furthermore, Figure 3 gives plots of these results.

**Table 2.** The average point estimate and the RMSE's associated with the estimators of $\theta = (\alpha,\beta,\gamma)$ using maximum likelihood and Simulated Annealing Algorithm

| | The Method | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ML | | | | SAA | | | |
| $n$ | 100 | 250 | 500 | 1000 | 100 | 250 | 500 | 1000 |
| RMSE($\alpha$) | 0.7433 | 0.4539 | 0.3127 | 0.2220 | 0.6044 | 0.4262 | 0.3090 | 0.2219 |
| $\bar{\hat{\alpha}}$ | 1.9484 | 1.9741 | 1.9930 | 1.9946 | 1.9573 | 1.9801 | 1.9968 | 1.9997 |
| RMSE($\beta$) | 0.9758 | 0.5971 | 0.4151 | 0.2927 | 0.8334 | 0.5664 | 0.4105 | 0.2926 |
| $\bar{\hat{\beta}}$ | 1.1558 | 1.0678 | 1.0304 | 1.0180 | 1.1441 | 1.0650 | 1.0302 | 1.0170 |
| RMSE($\gamma$) | 0.5879 | 0.3455 | 0.2366 | 0.1669 | 0.4676 | 0.3250 | 0.2339 | 0.1668 |
| $\bar{\hat{\gamma}}$ | 2.0468 | 2.0135 | 2.0118 | 2.0031 | 1.9857 | 1.9953 | 2.0060 | 2.0030 |

**Example 2:** In this example, it is assumed that $\theta = (\alpha,\beta,\gamma) = (1,2,0{:}5)$. Table 3 summarizes the average point estimate and the root mean squared errors associated with the estimators using maximum likelihood estimators and Simulated Annealing Algorithm. Furthermore, Figure 4 gives plots of these results.

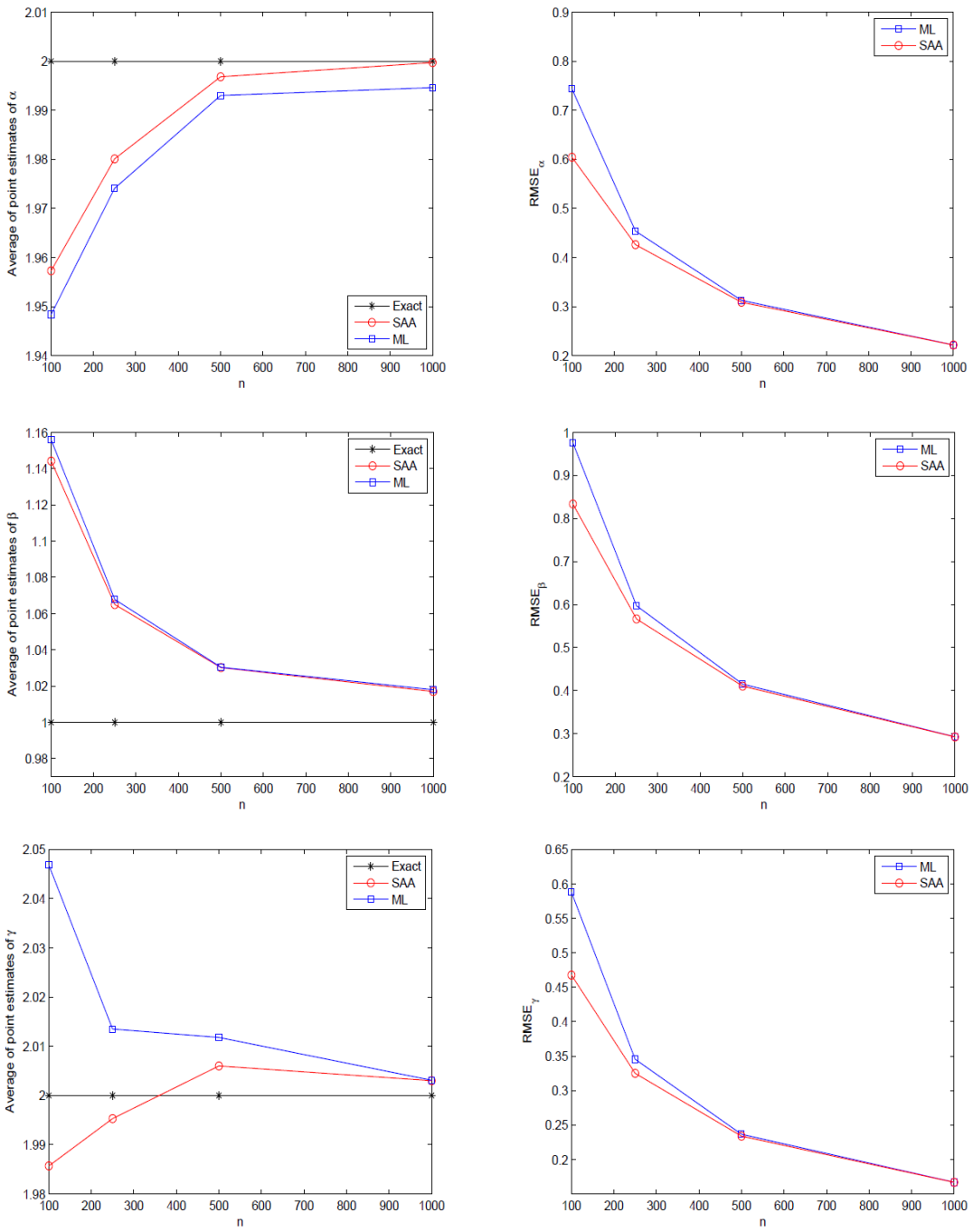


**Figure 3**. The averages of the point estimates and RMSE's for the parameters

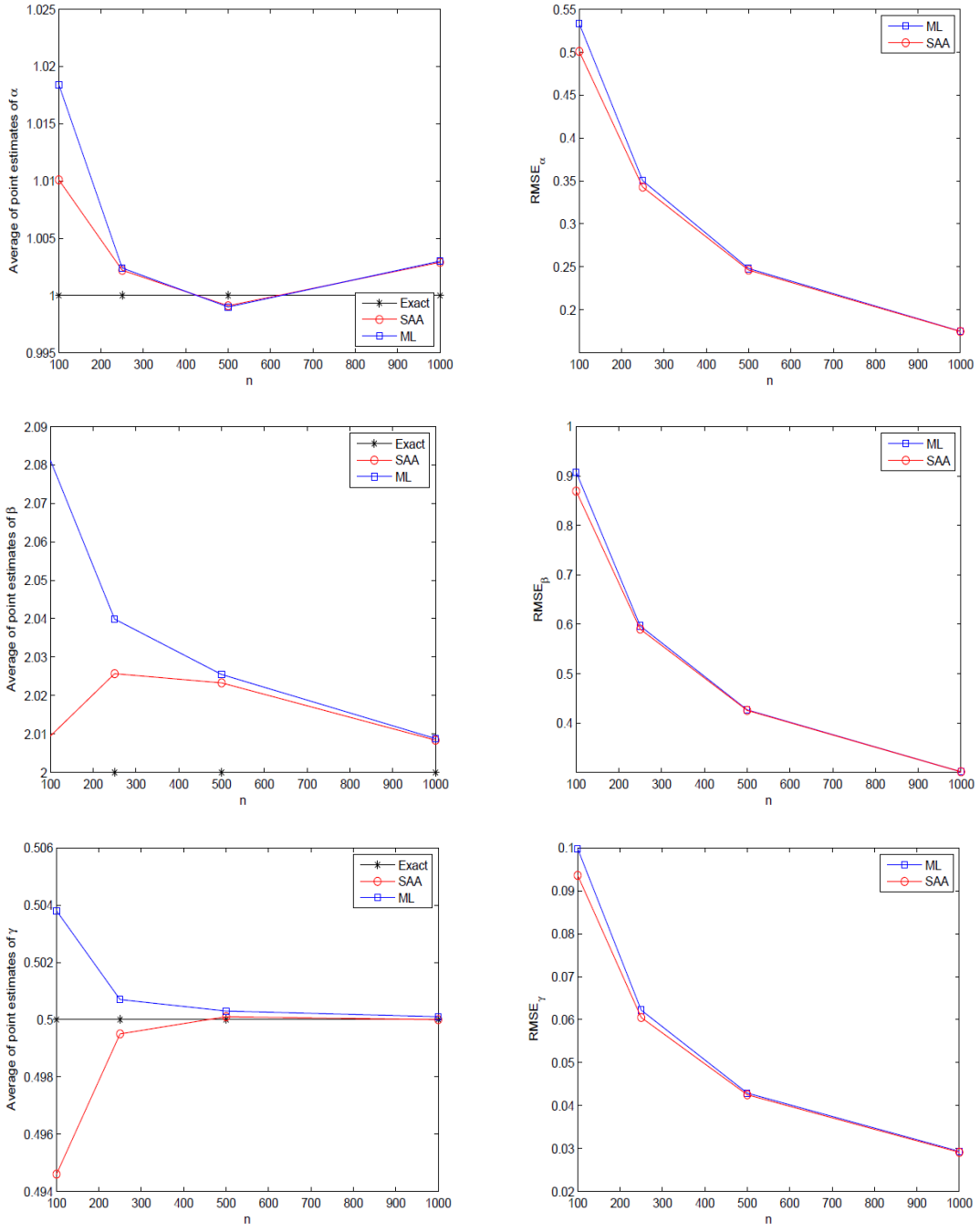


**Figure 4**. The averages of the point estimates and RMSE's for the parameters

**Table 3.** The average point estimate and the RMSE's associated with the estimators of θ = ($\alpha$,$\beta$,$\gamma$) using maximum likelihood and Simulated Annealing Algorithm

| | The Method | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ML | | | | SAA | | | |
| $n$ | 100 | 250 | 500 | 1000 | 100 | 250 | 500 | 1000 |
| RMSE($\alpha$) | 0.5330 | 0.3502 | 0.2481 | 0.1751 | 0.5007 | 0.3428 | 0.2463 | 0.1749 |
| $\bar{\hat{\alpha}}$ | 1.0184 | 1.0024 | 0.9990 | 1.0030 | 1.0101 | 1.0022 | 0.9991 | 1.0029 |
| RMSE($\beta$) | 0.9079 | 0.5964 | 0.4267 | 0.3013 | 0.8692 | 0.5901 | 0.4256 | 0.3012 |
| $\bar{\hat{\beta}}$ | 2.0811 | 2.0399 | 2.0255 | 2.0088 | 2.0095 | 2.0257 | 2.0233 | 2.0084 |
| RMSE($\gamma$) | 0.0998 | 0.0622 | 0.0429 | 0.0293 | 0.0936 | 0.0605 | 0.0425 | 0.0291 |
| $\bar{\hat{\gamma}}$ | 0.5038 | 0.5007 | 0.5003 | 0.5001 | 0.4946 | 0.4995 | 0.5001 | 0.5000 |

Based on the results summarized on tables 2 and 3 and plotted in Figures 3 and 4, we could immediately conclude that: as it was expected, the RMSE's are monotonically decreasing with increasing sample size *n*. The $\text{RMSE}_{ML}$ is greater than $\text{RMSE}_{SAA}$ for all investigated sizes *n*. Also, the average estimate for every parameters using SAA is more closer to the exact value than that obtained by using the traditional ML. From the above analysis, one can conclude that the Simulated Annealing Algorithm provides better estimators than the maximum likelihood method for the case studied in this paper. The performance of the techniques become almost the same when the sample size becomes very large.

## 7. CONCLUSION

In this paper, we applied the Simulated Annealing Algorithm to obtain the MLE of the three unknown parameters of the generalized linear failure rate distribution. The MLE's obtained using SAA were compared with those obtained using the gradient method by solving a nonlinear system of three equations numerically. Based on the results obtained, the proposed approach yields better point estimates of the three unknown parameters at all studied sample sizes. As a future work, SAA can be used to derive the MLE of the unknown parameters indexed to a lifetime distribution using different types of data. This work is in progress.

## REFERENCES

Brooks, S. P. and Morgan, B. J. T. (1995). Optimization using Simulated Annealing, *The Statistician*, **44**, 241-257.

Cramer, J. S. (1986). *Econometric Applications of Maximum Likelihood Methods*, New York: Cambridge University Press.

Finch, Stephen J., Nancy R. Mendell and Henry C. Thode, Jr. (1989). Probabilistic Measures of Adequacy of a Numerical Search for a Global Maximum, *Journal of*

*the American Statistical Association*, **84**, 1020-3.

Gupta, R.C., Gupta, R. D. (2007). Proportional reverse hazard rate model and its applications, *Journal of Statistical Planning and Inference*, **137**, 3525-3536.

Jones, M. T. (2003). *Artificial Intelligence Application Programming*, Wiley Dreamtech India Pvt Ltd.

Kirkpatrick, S., Gerlatt Jr., C. D. and Vecchi, M. P. (1983). Optimization by simulated annealing, *Science*, **220**, 671-680.

Sarhan, A. M. and Kundu, D. (2009). Generalized linear failure rate distribution, *Communications in Statistics-Theory and Methods*, **38**, 642 – 660.

Tan, C. M and Raghavan, N. (2008). *Simulated Annealing for mixture distribution analysis and its applications to reliability testing*, Simulated Annealing, Book edited by Chen Tan, I-Tech Education and Publishing, Vienna Austia, 227-256.