Int J Syst Assur Eng Manag
DOI 10.1007/s13198-016-0541-0

ORIGINAL ARTICLE

An efficient parameter estimation of software reliability growth models using gravitational search algorithm

Ankur Choudhary[1] · Anurag Singh Baghel[2] · Om Prakash Sangwan[3]

Received: 19 February 2016 / Revised: 24 July 2016
© The Society for Reliability Engineering, Quality and Operations Management (SREQOM), India and The Division of Operation and Maintenance, Lulea University of Technology, Sweden 2016

Abstract This paper presents an effective parameter estimation approach for software reliability growth models using gravitational search algorithm. A software reliability growth model is imperfect, if model parameters are unknown and are not validated on real-time software datasets. There exist several efficient numerical estimation techniques for parameter estimation of software reliability growth models. But they are not panacea. Sample size, biasing and initialization etc. always remain a constraint for best parameter estimation. Results indicate that gravitational search algorithm based technique for parameter estimation overcomes these problems and does superior quality parameter estimation. In this paper, extensive experiments on nine real-time datasets were conducted and results were analyzed to compare the proposed approach. The analysis results point towards the superiority of proposed approach over existing numerical estimation, genetic algorithm and cuckoo search methods.

Keywords Gravitational search · Parameter estimation · Software reliability growth model · Metaheuristics

✉ Ankur Choudhary
achoudhary@amity.edu

1 Department of Computer Science and Engineering, Amity University Uttar Pradesh, Noida, India

2 Department of Computer Science and Engineering, School of ICT, Gautam Buddha University, Greater Noida, India

3 Department of Computer Science and Engineering, Guru Jambheshwar University, Hisar, India

1 Introduction

In today's environment, the demand for software systems and software-controlled systems has increased rapidly. This demand of safety-critical, business-critical or system software's can lead to a serious injury, data loss or financial loss if any defect causes a serious failure.

Software reliability is an utmost important quality attribute among others, which describes the failure free operation of a software system under the stated conditions for a specified time period. In general terms, the reliability of software quantifies—how well the software provides the services intended by users. Through out the development lifecycle of any software, industry always faces challenges of resource management, effective coordination, better scheduling, and budgeting. The software reliability prediction and estimation provides confidence to the software designers about not only the challenges during the life cycle but also about the quality of software. The various phases of software development life cycle help for modelling of software reliability (Kapoor et al. 2011). Figure 1 below shows the relationship between phases of Software development life cycle (SDLC) and software reliability modelling techniques.

The software reliability growth models (SRGMs) are the most explored type of software reliability modelling approach. More than hundreds of software reliability growth models have been proposed by various researchers in the last three decades. These models statistically relate software failure data with well-known functions such as an exponential function, non-homogenous possion process (NHPP) etc. (Wood 1996a). The better will be the mapping of failure data with functions of well-known models, the better will be the accuracy and model.

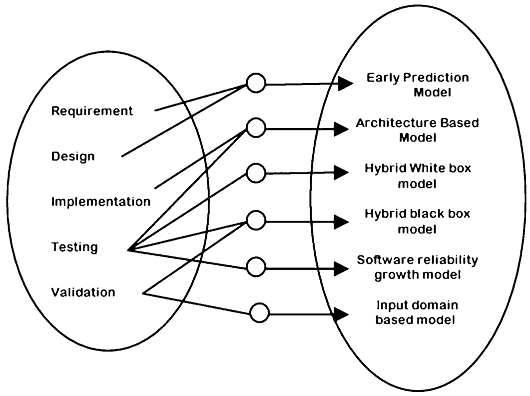

Fig. 1 Software reliability model classification based on Software development life cycle

SRGMs are further classified into failure rate models and non-homogenous Poisson process (NHPP) based models and has been explored in greater depth. Failure rate models were found better than NHPP models in some cases, but components of NHPP model are found more significant and effective (Kapoor et al. 2011; Williams 2007; Xie 1991; Goševa-Popstojanova and Trivedi 2001).

The NHPP based reliability models depend on time, error content function and error detection rate. A generalized and analytical software reliability model uses the mean value function to describe the stochastic software failure process. These models provide mathematical model framework to describe the software failure phenomena during the testing phase. The NHPP models face issue in estimation of mean value function from the experienced cumulative failures at a point of time. Equation parameters of these models are usually estimated using techniques like least square estimation or maximum likelihood estimation. Any mathematical modelling remains incomplete without applying parameter estimation. Software reliability modelling also requires a good parameter estimation technique, which provides the optimal mapping of cumulative failures with time. Basically, it turns into an optimization problem in which the objective function relates the known output variable with unknown parameters. The main aim is the estimation of unknown variables to achieve the minimum error in the output variable. Previous research says that the numerical estimation is most widely used parameter estimation approach for SRGMs. Least square estimation (LSE) and maximum likelihood estimation (MLE) are the most popular parameter estimation approaches among others but due to non-linearity of failure data, these two approaches do not always provide the optimal solution. These techniques also suffer from limitations such as: likelihood function maximization and error minimization and the existence of derivatives of the evaluation method (Hsu and Huang 2010; Tohma 1995).

To overcome these limitations several nature-inspired approaches have been proposed for parameter estimation of SRGMs. Genetic algorithm, ant colony optimization, particle swarm optimization, simulated annealing, cuckoo search optimization techniques are used in parameter estimation of SRGMs (Zhang et al. 2008; AL-Saati et al. 2013; Kim et al. 2015; Bidhan and Awasthi 2014; Singh et al. 2015).

In this paper, we propose gravitational search algorithm (GSA) based technique for parameters estimation of SRGMs. GSA is a well proven optimization algorithm enthused by the law of gravity and mass interactions. It has been intensively used by many researchers on complex problems as it provides better balance between intensification and diversification for parameter identification (Sahoo 2014). We performed experiments on nine real-time datasets for five well established SRGMs, Goel Okumoto model (GO), Generalized Goel model(GG), Delayed S-shaped model (DS), Logistic growth model (Pai 2013) and generalized fault-detection software reliability model (Pham Model) (Pham 2016).

The rest of the paper is organized as follows. Basic concepts of SRGMs and existing parameter estimation techniques are discussed in Sect. 2. Section 3 explains the basic concept of GSA and proposed approach. Section 4 validates the experimental results and Sect. 5 concludes the paper.

2 Software reliability growth model and parameter estimation techniques

Software reliability growth models based NHPP predict the cumulative failure count of software systems at any point of time based on their past failure behavior at any time t (Kapoor et al. 2011; Williams 2007).

These models also make the following assumptions (Kapoor et al. 2011; Hsu and Huang 2010; Sahoo 2014):

1. Remaining faults are the cause for failure in software systems.
2. At any point of time number of detected faults and remaining faults are proportional to each other.
3. Remaining faults uniformly affect the failure rate of software.
4. Detected faults are repaired immediately without introducing new faults.

The NHPP models can be represented mathematically using mean value function m(t), where a, b etc. are behavior parameters. The models selected for this study are shown below in Table 1.

The parameters in the model can be estimated using failure data by various estimation techniques. Once parameter estimation is over, the model can be used to analyze various performance measures. Parameter estimation problem for the stochastic system can be transmuted to an optimization problem. The aim of this estimation is to identify a set of parameters which is best fitted to the function and can map the failure data precisely. One way to address this parameter estimation problem is the application of numerical estimation such as LSE and MLE. LSE method determines the highest probable accurate set of parameter for a given experimental dataset (Hsu and Huang 2010). Moreover, it uses curve fitting process on the experimental dataset to estimate the unknown parameters (Tohma 1995). LSE also uses the nonlinear regression. LSE is quite simple to apply, as nonlinear regression is available as functionality in most of the commercial statistical packages (Kim et al. 2015). MLE is also an influential parameter estimation technique for statistical models. MLE has several significant statistical properties to be an optimal estimator for big amount of data. But the solution process of MLE is highly complex for evaluation of parameters. They need to be solved numerically (Hsu and Huang 2010; Wood 1996b; Schneidewind 1993; Aljahdali and El-Telbany 2009), which is a practical problem for industry professionals. The SRGM functions are bit complex log likelihood etc., which also makes MLE to very difficult. MLE and LSE were compared by Wood (1996b) assuming that the estimated errors are normally distributed, and he found that confidence intervals for MLE perform better on large samples rather than small samples. It provides asymmetric confidence intervals for the total defect parameter. While the confidence interval for LSE is symmetric.

The LSE becomes the choice, where MLE cannot provide the satisfactory parameter estimation. It offers steady outcomes in broader datasets hence it becomes a highly preferred method used by the software practitioners. The literature shows limitations of numerical estimation such as it is usually non-trivial. Sometimes the estimation shows heavy biases for small size samples. The selection of initial parameter values is also a sensitive issue. Nature has always been a source of inspiration. Few authors have proposed neural network based approach to map the non linearity of software failures (RajKiran and Ravi 2007; Mohanty et al. 2013; Lo 2009; Su et al. 2005). But, the major drawback of these techniques is that they require a very large training data set to train the system, which produces a high computational cost and time. These techniques also consume high computation cost for prediction of no of failure per iteration.

To overcome such issues, Nature-inspired approaches have been applied in various areas of software engineering, software testing and software reliability (Arora and Baghel 2015). In 1995, Minohara and Tohma (1995) proposed a model using the genetic algorithms for SRGMs and found that it is a more stable approach for getting an estimate. Zhang et al. (Amin et al. 2013) proposed particle swarm optimization (PSO) technique as a new parameter estimation approach for SRGM, but observed that this approach required high search range and low convergence speed. Aljahdali and El-Telbany (2009) proposed the use of a multi-objective genetic algorithm for SRGM. Hsu et al. (2010) proposed a modified genetic algorithm (MGA) based parameter estimation to improve the performance of basic GA for addressing the parameter estimation problem of SRGM. Al-Saati et al. (2013) proposed a Cuckoo Search (CS) based parameter estimation technique. CS performed better than PSO and ACO techniques.

Table 1 A summary of selected software reliability growth models

Model detail	Model type	MVF (m(t))	Parameter significance
Goel–Okumoto (GO) model	Concave	$m(t) = a\left(1 - e^{-bt}\right)$ $a > 0, b > 0$	The scale is determined by 'a' and shape of the mean-value function is determined by 'b'
Generalized Goel model	Concave	$m(t) = a(1 - e^{(-bt^{c})})$ $a > 0, b > 0, c > 0$	Expected total faults count on event is 'a', the quality of testing is reflected by 'b' and 'c' parameters
Delayed S shaped model	S-shaped	$m(t) = a(1 - (1 + bt)e^{(-bt)}))$ $a > 0, b > 0$	Expected total faults count on event is 'a', the fault detection rate is reflected by 'b'.
Logistic growth model	S-shaped	$m(t) = \frac{a}{(1+ce^{-bt})}$ $a > 0, b > 0, k > 0$	Expected total faults count on event is 'a', 'c' and 'b' parameters is used to fit the failure data.
Generalized fault-detection software reliability (Pham model)	S-shaped	$m(t) = N\left(1 - \frac{\beta}{\beta + \left(\frac{c}{b}\right)\ln\left(\frac{a+e^{bt}}{1+a}\right)}\right)^{\alpha}$	Expected total faults count before testing is "N", Expected total faults count in infinite time is "a" Time dependent Fault-detection rate is 'b' Probability density function is generalized by α and β.

3 Parameter estimation using gravitational search algorithm

Gravitational Search Algorithm (Rashedi et al. 2009) is a memory less, population based optimization algorithm based on Newton's Law of gravity and motion (Rashedi et al. 2009). GSA is successfully applied to various area such as in neural network training, robotics, Optical, Bioinformatics, Software Engineering, Networking, Image Processing (Ojugo et al. 2013; Sheikhan and Rad 2013; Biglari et al. 2013; Seljanko 2011; Saucer and Sih 2013; Bababdani and Mousavi 2013; Amoozegar and Nezamabadi-pour 2012; Han and Chang 2012; Sun and Zhang 2013; Sabri et al. 2013; Biswas et al. 2013), etc.

In GSA, masses are considered as agents. Each agent in GSA has following four parameters such as position of the agent in d^{th} dimension, inertia mass, active and passive gravitational mass. The position of agents is represented as solutions of optimization problem, which is navigated by adjustment of the gravitational and inertia masses. These agents will present an optimum solution in the solution space. The heavy masses correspond to good solutions because their movement is slower than lighter masses. Stopping criteria for the algorithm can be either fixed number of iterations or achievement of desired solution.

The main motive of proposed approach is to get better accuracy of the parameter estimation for SRGMs. The main steps of the proposed approach are shown in Fig. 2 and steps as follows:

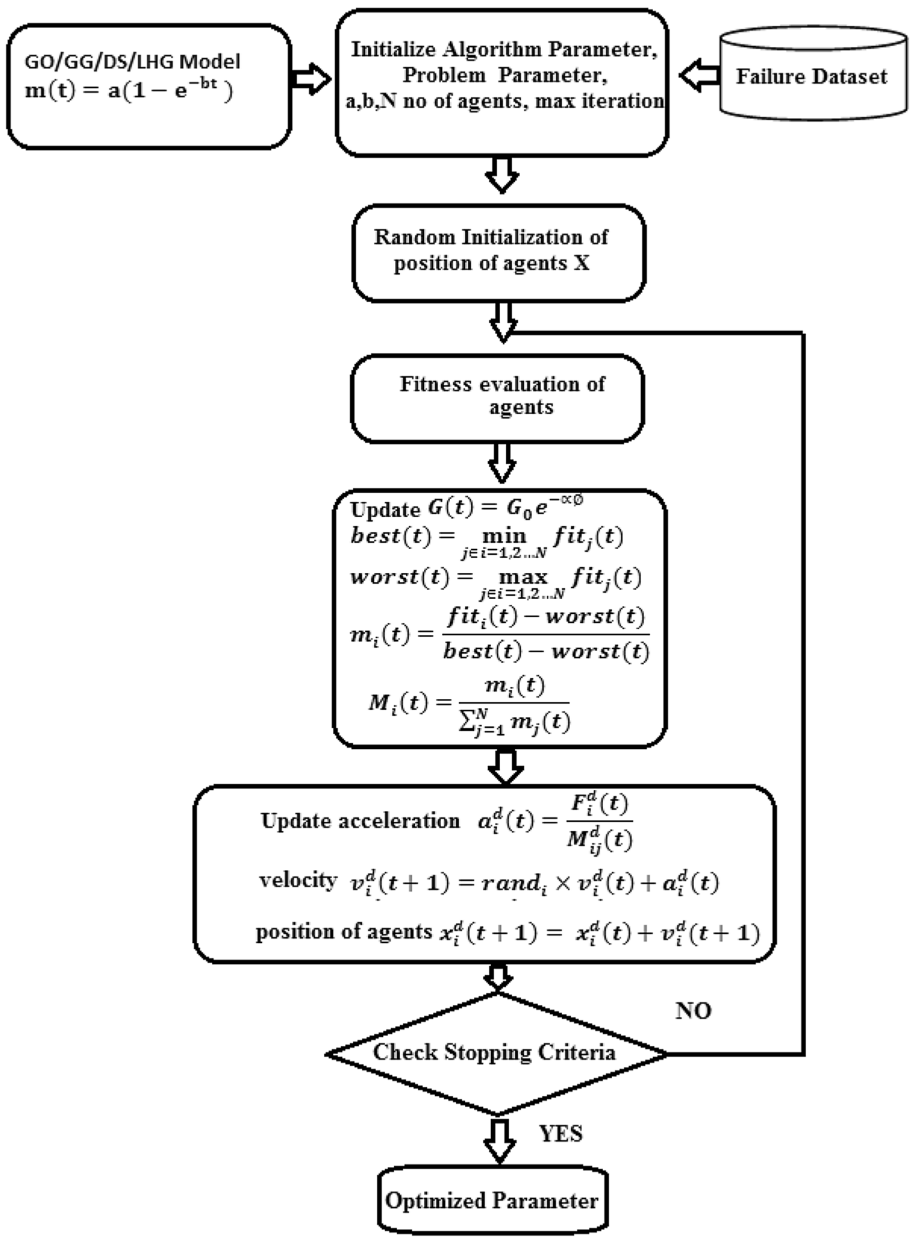

Fig. 2 Proposed approach for parameter estimation using gravitational search algorithm

1. The first step is the formulation of the objective function. Minimize: fit(p), Subject to: $p_i \in \mathrm{P_i} = 1, 2, 3, 4.\ldots$M, Where fit(p) is an objective function for parameter estimation and p is the set of each unknown parameter p_i, M is the number of unknown parameters $\mathrm{p_i}$, is the set of the possible values for each unknown parameter, where $L^{p_i} \leq \mathrm{P_i} \leq U^{p_i}$ and L^{p_i} and U^{p_i} are the lower and upper bounds for each unknown parameter. In our case a, b, c etc. are the unknown parameters according to model choose. The GSA algorithm parameters are also initialized such as N is the number of agents or population size and the number of maximum iteration (T), or termination criterion.
2. Perform random Initialization of position of agents X between the lower and upper bound of parameters.
3. Repeat steps iii to vii until the termination criteria reached (maximum number of iteration). a)Evaluate the fitness of N agents.
4. Update the $\boldsymbol{G(t) = G_0 e^{-\propto\emptyset}}$,$\boldsymbol{best(t)} = \min_{j\in i=1,2\ldots N} \boldsymbol{fit_j(t)}$ and $\boldsymbol{worst(t)} = \max_{j\in i=1,2\ldots N} \boldsymbol{fit_j(t)}$ agents and $\boldsymbol{m_i(t)} = \frac{\boldsymbol{fit_i(t) - worst(t)}}{\boldsymbol{best(t) - worst(t)}}$, $\boldsymbol{M_i(t)} = \frac{\boldsymbol{m_i(t)}}{\sum_{j=1}^{N} \boldsymbol{m_j(t)}}$.
5. Compute the total force $\boldsymbol{F_i^d(t)}$ in different directions.
6. Compute the acceleration $\boldsymbol{a_i^d(t)} = \frac{\boldsymbol{F_i^d(t)}}{\boldsymbol{M_{ij}^d(t)}}$, **velocity** $\boldsymbol{v_i^d(t+1)} = \boldsymbol{rand_i} \times \boldsymbol{v_i^d(t)} + \boldsymbol{a_i^d(t)}$ where $rand_i$ is uniform random variable between [0, 1].
7. Update position of agents $\boldsymbol{x_i^d(t+1)} = \boldsymbol{x_i^d(t)} + \boldsymbol{v_i^d(t+1)}$.

4 Experimental design and result discussions

For the experiment and analysis, nine datasets including open source dataset is selected. Table 2 below shows the details of the selected datasets.

The comparison of SRGMs was used for validation of accuracy of the proposed approach employed for parameter estimation. Five well established SRGMs are selected, GO model, DS model have two unknown parameters, GG Model and LG Model have three unknown parameters and Pham model have six unknown parameters.

As a comparison criterion, the mean square error (MSE) shows the deviation between the predicted and actual data. The smaller values of MSE show the better estimation of the model. The Theil statistic (TS) shows the average deviation percentage, over all periods with respect to the actual data. The closer the value of TS towards zero, better the prediction quality of the model. We selected MSE and TS as our comparison criteria, details of which are shown in Table 3.

Following experiments has been performed to evaluation the GSA based model. First, the performance

Table 3 Comparison criteria

S. No	Criteria	Formula
1	Mean square error (Pham 2007)	$\mathrm{MSE} = \frac{\sum_{i=1}^{k}\left(m(t_i)' - m(t_i)\right)^2}{k-p}$
2	Theil statistic (Sharma et al. 2010)	$\mathrm{TS} = \frac{\sum_{i=1}^{k}\left(m(t_i)' - m(t_i)\right)^2}{\sum_{i=1}^{k}\left(m(t_i)'\right)^2}$

Table 2 Software reliability failure data set

Dataset	Value	# of Failures	# of weeks/months
DACS Datasets (Musa 1980)			
FC2	Real time command and control	54	11
FC6	Commercial sub system	73	9
FC40	Military	101	50
Tandem computers software failure (Wood (1996b)			
DS1	Computers software failure	100	20
DS2	Computers software failure	176	18
Misra's Space Shuttle Software Failure Data (Misra 1983)			
STS	Space Shuttle flights STS2, STS3, STS4	231	38
Tohma's Software failure data (Tohma et al. 1989)			
TS	Monitoring & real-time control	86	22
WebERP (06/2003 to 07/2008) (SourceForge.net 2008; Hsu et al. 2011)			
DS3	Web-based integrated accounting ERP system	146	60
OpenProj (08/2007 to 07/2008) (SourceForge.net 2008; Hsu et al. 2011)			
DS4	Open source project management software	94	49

Table 4 MSE values comparison of CASRE tool and GSA

Dataset	GO model		GG model		DS model	
	CASRE tool	GSA	CASRE tool	GSA	CASRE tool	GSA
FC2	27.56	12.35	29.37	15.73	11.85	7.49
FC6	12.81	13.48	12.24	39.6	12.3	14.03
DS1	207.81	41.64	199.95	94.78	168.67	59.26

Table 5 Parameter Initialization for gravitational search algorithm

Parameter	Value
Population Size	50
A	20
G_0	3%
Number of iterations	1000

validation of GSA based model with numerical estimation techniques was done. Second, the performance validation with the existing CS and GA based approach.

The validation of effectiveness for proposed approach is done by comparing the existing numerical methods using CASRE tool. CASRE tool is one of the well-proven software reliability modelling tools. For comparison, LSE based approach is selected in CASRE tool for the parameter estimation of SRGMs. Table 4 shows the result of comparison between CASRE tool and GSA base approach.

The three traditional models GO Model and GG Model and DS model were selected for comparison because LG and Pham model is not available in CASRE tool. The results indicate that MSE values of proposed model are less than the existing values in most of the observations. For DS1, it shows that MSE values are much smaller than CASRE tool values. The results of all the selected models are better in case of FC2 and DS1 datasets but not in case of FC6. We have also compared and evaluated the performance of proposed techniques with CS based technique and GA based technique (AL-Saati et al. 2013) using nine real world datasets. The number of maximum iterations was set to 1000 for all the approaches as stopping criteria. All the techniques are executed 15 times repeatedly on each model for each dataset, and the MSE and TS are used for this validation.

Starting with experiment design, first of all the Gravitational search algorithm parameters were set as shown in Table 5.

Results indicate that MSE and TS values of GSA based approach are smaller than the values of the GA based and CS based approach for most of the models datasets. In particular, the MSE and TS values of GSA for all models in FC2, FC6, FC40, DS2 and Thoma dataset are significantly different from GA and CS based approach. Figure 3 shows the MSE values of all the models on selected datasets. The value of MSE in all the cases except Pham Model on FC6 shows better or comparable results with CS and GA based approach. These results indicate the superiority of GSA approach (Fig. 3; Table 6).

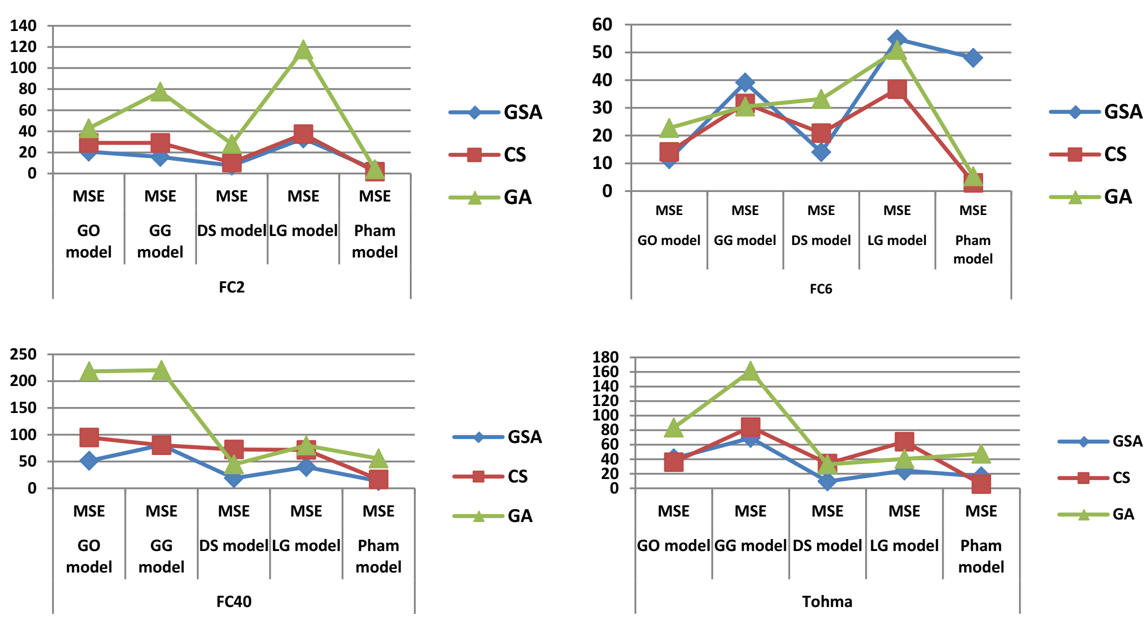

Fig. 3 Performance of GSA, CS and GA based approach on various datasets and selected models

Table 6 Results of comparison criteria of proposed approach on 9 real world datasets

Dataset	Model	Comparison	GSA	CS	GA
FC2	GO model	MSE	20.68638	29.00014	42.82487
		TS	12.35498	14.12895	16.60541
	GG model	MSE	15.73348	29.01508	77.54135
		TS	10.53907	14.09447	23.09338
	DS model	MSE	7.490235	10.39528	28.03524
		TS	7.421363	8.623594	14.26107
	LG model	MSE	33.2019	37.3729	117.5935
		TS	14.06095	16.03998	29.49175
	Pham model	MSE	3.718041	1.826812	3.970186
		TS	5.251653	3.679763	5.380664
FC6	GO model	MSE	11.4848	14.15892	22.6981
		TS	6.332403	6.98499	8.653902
	GG model	MSE	39.14796	31.63734	30.46988
		TS	11.55195	10.30282	9.90179
	DS model	MSE	14.03783	20.89811	33.18888
		TS	7.071327	8.404833	10.8062
	LG model	MSE	54.75683	36.73316	50.86513
		TS	13.68148	11.12131	13.35318
	Pham model	MSE	48.06617	2.972592	5.30133
		TS	12.94844	3.252533	4.245703
FC40	GO model	MSE	51.42722	94.75789	218.223
		TS	10.34132	13.41843	20.38155
	GG model	MSE	80.73764	80.37687	220.6913
		TS	13.03058	12.58527	21.17034
	DS model	MSE	18.85571	72.42391	44.51512
		TS	6.316877	11.82036	9.331651
	LG model	MSE	39.77958	71.75866	79.9226
		TS	9.186928	11.97131	12.86249
	Pham model	MSE	13.48488	16.63494	55.54432
		TS	5.36858	5.965254	10.7983
DS1	GO model	MSE	41.64973	95.27767	65.41596
		TS	8.452943	12.34884	10.52604
	GG model	MSE	94.78755	72.52286	73.8813
		TS	12.6835	10.86018	11.16679
	DS model	MSE	59.26688	70.39161	60.97078
		TS	10.16031	10.99357	10.29254
	LG model	MSE	132.331	103.7217	156.873
		TS	15.03721	13.18369	16.52511
	Pham model	MSE	90.17587	21.52432	126.768
		TS	12.47828	6.127038	14.57453
DS2	GO model	MSE	367.8824	352.9038	1572.502
		TS	15.88208	15.59411	33.0528
	GG model	MSE	537.4242	463.2452	2017.518
		TS	19.34208	17.49565	37.40498
	DS model	MSE	214.7282	228.2654	949.9159
		TS	12.20088	12.53908	25.63977
	LG model	MSE	88.52963	125.8737	479.6042
		TS	7.843258	9.263852	18.22409
	Pham model	MSE	814.4612	1983.276	1399.151
		TS	23.81112	37.15662	31.11373

Table 6 continued

Dataset	Model	Comparison	GSA	CS	GA
Tohma	GO model	MSE	41.19227	35.96293	83.35483
		TS	10.40315	9.806347	14.20342
	GG model	MSE	69.15074	84.29778	161.5447
		TS	13.55064	14.58231	19.76426
	DS model	MSE	9.607635	34.02993	32.82586
		TS	5.079425	8.847071	8.46516
	LG model	MSE	24.37432	63.93187	40.60642
		TS	8.072933	12.69521	10.27679
	Pham model	MSE	16.38813	5.80683	47.4159
		TS	6.6634	3.968783	10.89881
Mishra	GO model	MSE	206.8974	139.8348	2885.724
		TS	10.01629	9.482933	38.8301
	GG model	MSE	586.6731	345.5955	3314.718
		TS	16.28824	12.22388	41.63012
	DS model	MSE	170.5113	178.7255	2320.349
		TS	9.374149	9.588577	34.83043
	LG model	MSE	439.9000	178.7255	1531.936
		TS	15.14547	9.588577	28.29123
	Pham model	MSE	981.3232	3392.587	3258.426
		TS	22.42963	42.13612	41.27886
DS3	GO model	MSE	237.002	109.3179	875.3999
		TS	22.3457	15.35764	41.43447
	GG model	MSE	376.7438	135.2681	515.9911
		TS	27.69044	17.08349	33.18529
	DS model	MSE	189.9687	175.082	252.7855
		TS	20.17422	19.43568	22.90287
	LG model	MSE	301.5138	92.70592	311.1407
		TS	25.46817	14.14248	25.85014
	Pham model	MSE	180.3941	173.7133	461.133
		TS	19.70681	19.35892	31.48576
DS4	GO model	MSE	18.26567	7.837463	34.65955
		TS	6.279295	4.207208	7.266888
	GG model	MSE	147.0641	1.699153	130.1825
		TS	17.19493	1.958926	16.28536
	DS model	MSE	47.9433	47.89308	116.4523
		TS	10.39466	10.40022	15.76252
	LG model	MSE	94.00700	6.652776	155.7647
		TS	14.47982	3.876213	18.62972
	Pham model	MSE	46.80663	2.865497	85.99036
		TS	10.24052	2.477513	13.74052

We developed the following hypothesis to validate the result:

1. Ho: GSA $=$ CS.
2. Ha: GSA $\neq$ CS.

One Way ANOVA was performed to test the hypothesis at 95% confidence interval. Table 7 describes the average MSE values and ANOVA test results. For the FC2 dataset, the p-values are 3.12E–12 for the GO model, 3.68E–10 for the GG model, 5.57E–16 for the DS model, 5.057E–06 for the LG model and similar on other datasets also. The p-values are smaller than the significance level of 0.05 for each selected dataset and models. Therefore, we reject the null hypothesis and accept the alternate hypothesis (Table 7).

Table 7 ANOVA Test Results

Dataset	Techniques	GO model		GG model		DS model		LG model	
		Avg.	*P* value	Avg.	*P* value	Avg.	*P* value	Avg.	*P* value
FC2	GSA	20.69		15.73		7.49		33.20	
	CS	29.00	3.12E–12	29.02	3.68E–10	10.40	5.57E–16	13.27	5.057E–06
FC6	GSA	10.67		39.15		14.04		54.76	
	CS	14.16	1.45E–17	31.64	1.22E–15	20.90	2.22E–13	36.73	8.348E–6
FC40	GSA	51.43		80.74		18.73		39.78	
	CS	94.76	2.24E–08	80.38	6.39E–16	72.42	1.31E–06	71.76	2.78E–11
DS1	GSA	41.65		94.73		59.27		132.33	
	CS	95.28	1.14E–09	69.18	1.68E–16	70.39	2.86E–30	103.72	1.09E–21
DS2	GSA	367.88		655.36		214.73		88.53	
	CS	352.90	4.45E–30	463.25	4.86E–18	228.27	6.93E–36	125.87	2.39E–24
Tohma	GSA	41.19		69.15		9.61		24.37	
	CS	71.48	2.95E–10	84.30	9.86E–07	34.03	1.85E–05	84.30	1.25E–07
Mishra	GSA	206.74		586.67		170.01		439.90	
	CS	139.83	9.16E–12	345.60	4.372E–09	178.73	9.198E–28	345.60	4.701E–13

5 Conclusions and future scope

An effective parameter estimation approach using GSA for SRGMs was discussed in this paper that overcomes the limitations of existing approaches. The evaluation of proposed approach is done by performing extensive experiments on nine well-known datasets for five well established SRGMs. The comparison results of GSA and CASRE tool reflects the effectiveness of GSA based model. The proposed approach is also compared with GA and CS based approach. It is evident from the experimental results that GSA can provide optimal solution more accurately than GA and CS approach. In future, we wish to conduct a comparative experimental study, between GSA and other hybrid meta-heuristics for better parameter estimation approach of SRGMs.

References

Aljahdali SH, El-Telbany ME (2009) Software reliability prediction using multi-objective genetic algorithm. ACS Int Conf 2009:293–300

AL-Saati D, Akram N, Abd-AlKareem M (2013) The use of cuckoo search in estimating the parameters of software reliability growth models. IJCSIS Int J Comput Sci Inf Secur 11(6):39–46

Amin A, Grunske L, Colman A (2013) An approach to software reliability prediction based on time series modeling. J Syst Softw 86(7):1923–1932

Amoozegar M, Nezamabadi-pour H (2012) Software performance optimization based on constrained GSA. In: The 16th CSI international symposium on artificial intelligence and signal processing (AISP 2012), pp 134–139

Arora D, Baghel AS (2015) Application of genetic algorithm and particle swarm optimization in software testing. IOSR J Comput Eng 17(1):75–78

Bababdani BM, Mousavi M (2013) Gravitational search algorithm: a new feature selection method for QSAR study of anticancer potency of imidazo[4,5-b]pyridine derivatives. Chemo Intell Lab Syst 122:1–11

Bidhan K, Awasthi A (2014) Estimation of reliability parameters of software growth models using a variation of particle swarm optimization, confluence—the next generation information technology summit. In: 5th international conference, IEEE, pp 800–805

Biglari M, Assareh E, Poultangari I, Nedaei M (2013) Solving blasius differential equation by using hybrid neural network and gravitational search algorithm (HNNGSA). Glob J Sci Eng Technol 11:29–36

Biswas A, Mishra KK, Tiwari S, Misra AK (2013) Physics-inspired optimization algorithms: a survey. J Optim 2013:438152

Goševa-Popstojanova K, Trivedi KS (2001) Architecture-based approach to reliability assessment of software systems. Perform Eval 45(2):179–204

Han X, Chang X (2012) Chaotic secure communication based on a gravitational search algorithm filter. Eng Appl Artif Intell 25(4):766–774

Hsu CJ, Huang CY (2010) A study on the applicability of modified genetic algorithms for the parameter estimation of software reliability modeling. In: Computer software and applications conference (COMPSAC), IEEE 34th Annual, pp 531–540

Hsu C-J, Huang C-Y, Chang J-R (2011) Enhancing software reliability modeling and prediction through the introduction of time-variable fault reduction factor. Appl Math Model 35(1):506–521

Kapoor PK, Pham H, Gupta A et al (2011) Software reliability assessment with OR applications, 1st edn. Springer, London

Kim T, Lee K, Baik J (2015) An effective approach to estimating the parameters of software reliability growth models using a real-valued genetic algorithm. J Syst Softw 102:134–144

Lo J (2009) The implementation of artificial neural networks applying to software reliability modelling. In: 21st annual international

conference on chinese control and decision, IEEE Press, pp 4385–4390
Minohara T, Tohma Y (1995) Parameter estimation of hyper-geometric distribution software reliability growth model by genetic algorithms. In: 6th international symposium, IEEE, pp 324–329
Misra PN (1983) Software reliability analysis. IBM Syst J 22(3):262–270
Mohanty R, Ravi V, Patra MR (2013) Hybrid intelligent systems for predicting software reliability. Appl Soft Comput 13(1):189–200
Musa JD (1980) Software reliability data. Technical report. Cyber security and information systems information analysis center, New York
Ojugo AA, Yoro RE, Okonta EO, Eboka AO (2013) A hybrid artificial neural network gravitational search algorithm for rainfall runoffs modeling and simulation in hydrology. Prog Intell Comput Appl 2:22–33
Pai GJ (2013) A survey of software reliability models. arXiv: 2013, arXiv: 1304.4539
Pham H (2007) System software reliability. Springer, Berlin
Pham H (2016) A generalized fault-detection software reliability model subject to random operating environments. Vietnam J Comput Sci 3(3):145–150
RajKiran N, Ravi V (2007) Software reliability prediction using wavelet neural networks. In: International conference on computational intelligence and multimedia applications, vol 1, IEEE Sivakasi, Tamil Nadu, pp. 195–199
Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inform Sci 179(13):2232–2248
Sabri NM, Puteh M, Mahmood MR (2013) A review of gravitational search algorithm. Int J Adv Soft Comput Appl 5(3):1–39
Sahoo G (2014) A review on gravitational search algorithm and its applications to data clustering & classification. Int J Intell Syst Appl (IJISA) 6(6):79
Saucer TW, Sih V (2013) Optimizing nanophotonic cavity designs with the gravitational search algorithm. Opt Express 21(18):20831–20836
Schneidewind NF (1993) Software reliability model with optimal selection of failure data. IEEE Trans Softw Eng 19(11):1095–1104
Seljanko F(2011) Hexapod walking robot gait generation using genetic gravitational hybrid algorithm. In: The 15th international conference on advanced robotics, pp. 253–258
Sharma K, Garg R, Nagpal CK et al (2010) Selection of optimal software reliability growth models using a distance based approach. Reliab IEEE Trans 59(2):266–276
Sheikhan M, Rad MS (2013) Gravitational search algorithm–optimized neural misuse detector with selected features by fuzzy grids–based association rules mining. Neural Comput Appl 23(7–8):2451–2463
Singh PK, Panda RK, Sangwan OP (2015) A critical analysis on software fault prediction techniques. World Appl 33(3):371–379
SourceForge.net (2008) An open source software website. http://sourceforge.net
Su YS, Huang CY, Chen YS, Chen JX (2005) An artificial neural-network-based approach to software reliability assessment. In: TENCON 2005–2005 IEEE Region 10 conference, IEEE, pp 1–6
Sun G, Zhang A (2013) A hybrid genetic algorithm and gravitational using multilevel thresholding. Pattern Recognit Image Anal 7887:707–714
Tohma Y, Jacoby R, Murata Y et al (1989) Hyper-geometric distribution model to estimate the number of residual software faults. Comput Softw Appl Conf COMPSAC 89:610–617
Williams DP (2007) Study of the warranty cost model for software reliability with an imperfect debugging phenomenon. Turk J Electr Eng 15(3):369–381
Wood A (1996a) Predicting software reliability. Computer 29(11):69–77
Wood A (1996b) Software reliability growth models. Tandem technical report, 96(130056)
Xie M (1991) Software reliability modelling, vol 1. World Scientific, Singapore
Zhang KH, Li AG, Song BW (2008) Estimating parameters of software reliability models using PSO. Comput Eng Appl 44(11):47–49