A comparison of continuous genetic algorithm and particle swarm optimization in parameter estimation of Gompertz growth model

Windarto, Eridani, and Utami Dyah Purwati

View Online

Export Citation

ARTICLES YOU MAY BE INTERESTED IN

A Comparison of Continuous Genetic Algorithm and Particle Swarm Optimization in Parameter Estimation of Gompertz Growth Model

Windarto[1,a)], Eridani[1] and Utami Dyah Purwati[1]

[1]*Department of Mathematics, Faculty of Science and Technology, Universitas Airlangga, Indonesia.*

a)Corresponding author: windarto@fst.unair.ac.id

Abstract. Genetic algorithm and Particle Swarm Optimization are heuristic optimization methods inspired by genetic principles and swarm behavior phenomena, respectively. Those two methods are initiated by random generation of initial populations (initial solutions), fitness evaluation of every solution, solution updating until a termination condition are met. It is well known that those two methods are not always converge to an optimal solution. Those methods sometimes converge to suboptimal solutions, solution near the optimal solution. In this paper, continuous genetic algorithm and particle swarm optimization were implemented to estimate parameters in the Gompertz growth model from rooster weight data cited from literature. Although the best results of the two models were not significantly differs, we found that the particle swarm optimization method was more robust than the continuous genetic algorithm. Hence, the particle swarm optimization method is more recommended than the continuous genetic algorithm.

Keywords: Gompertz growth model, rooster weight dynamic, parameter estimation, particle swarm optimization.

INTRODUCTION

Mathematical models are useful tool to describe many real problems. A mathematical model is usually began by identification of a real problem. Then one could construct a suitable mathematical model and determining mathematical solution of the model. Finally, one should interpret mathematical solution of the model into real problem points of view. A mathematical model might occur in either a deterministic model or a probabilistic (stochastic) model. Mathematical model validation could be performed whenever relevant data from real phenomena are available. If the predicted results from a mathematical model fit the real data, then the model is said a good model. When the predicted results from the model differ significantly the real data, then the model should be improved and modified.

Most mathematical models contain one or more parameters. The parameters should be estimated in order to accurately perform model simulation. Parameter estimation of a mathematical model could be considered as an optimization problem. Deterministic optimization methods such as conjugate gradient method, Nelder-Mead method or Newton method could be applied to estimate parameters in a mathematical model whenever analytical solution of the model could be presented in closed form [1]. Unfortunately, deterministic optimization methods such as Nelder-Mead or Newton method fail to converge into global minimum of a function if the function has many local minima [2]. Moreover, some mathematical models occur in non-linear ordinary differential equation systems, so exact solution (closed form solution) of the model could not be determined. In this case, heuristic method such as particle swarm optimization and genetic algorithm method could be implemented to estimate parameter values from the models.

Particle swarm optimization and genetic algorithm are optimization methods based on a population-based stochastic search process [3, 4]. Particle swarm optimization methods and modified particle swarm optimization have been widely applied in many areas, including performance improvement of Artificial Neural Network [5, 6], scheduling problems [7, 8], flowshop scheduling problem [9], traveling salesman problem [10], vehicle routing problem [11, 12] and clustering technique [13]. Genetic algorithm has been in parameter estimation in poultry growth model [14, 15] and parameter estimation for dynamical system model [1, 16].

Some authors compared performance of particle swarm optimization and genetic algorithm in some research are. Yang et al. compared the methods in a Hidden Markov Model training [17]. Wang et al. have been compared performance of genetic algorithm and particle swarm optimization in relativistic backward wave oscillator [18]. Islam et al. have compared performance of some nature inspired algorithms including genetic algorithm and particle swarm optimization in function optimization of some benchmark functions [19]. In this paper, we compared performance of continuous genetic algorithm and particle swarm optimization in parameter estimation of Gompertz growth model.

The remainder of this paper is organized as follows. Section 2 briefly presents particle swarm optimization and genetic algorithm procedure. Comparison of particle swarm optimization and genetic algorithm in parameter estimation of Gompertz growth model will be presented in Section 3. Finally, conclusions are presented in Section 4.

CONTINUOUS GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION

Genetic algorithm is inspired from principles of genetic and natural selection in a life organism. Therefore, many terms such as gene chromosome, individual, parent, selection, mating, crossover, offspring in genetic algorithm are adopted from biology. From mathematical point of view a gene represents a variable, while a chromosome or an individual represents a solution. Genetic algorithm has at least the following elements, namely populations of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutation of new offspring [20]. Genetic algorithm diagram is presented in the Figure 1. The diagram is adapted from Haupt and Haupt [2].
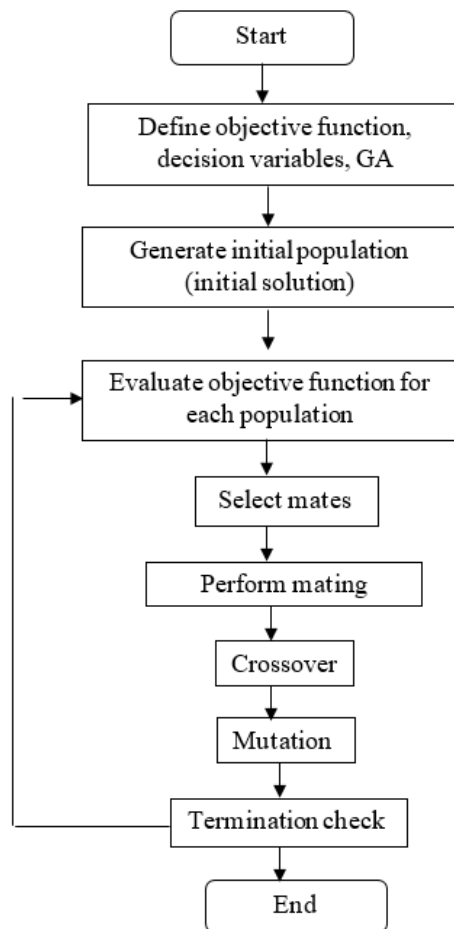

FIGURE 1. Flowchart of continuous genetic algorithm.

We can transform an optimization problem into a minimization problem. Here is genetic algorithm procedure for finding either optimal or suboptimal solution of a minimization problem [2, 16, 20]:

(1) Define the objective function (the cost function) and decision variables related to the optimization problem.
(2) Determine parameter values in genetic algorithm, namely number of generations/number of iteration, crossover probability, and mutation probability/mutation rate.
(3) Generate initial solutions (initial population) from the search space/solution space.
(4) Evaluate cost function of each solution (individual). In a minimization problem, all individuals are ordered from the lowest to the highest of objective function value objectivecost to the highest cost.
(5) Select part of individuals for the next generation as parent individuals. Only the best solutions are maintained for the next generation. The remaining individuals are replaced by better individuals. The selection rate parameter determines the fraction of all population that survives for the next generation. The typical value of selection rate parameter is 50%.
(6) Carry out mating process from parent individuals.
(7) Do crossover process to generate offspring individual.
(8) Perform mutation process to part of solutions to generate solutions.
(9) Test termination condition. If the termination condition did not satisfy yet, then go to the fourth step.

A main problem in the genetic algorithm is premature convergence where the solutions converge to a local optimum. The premature convergence occurs when a high fitness solution (individual) quickly dominate the population. The problem especially occurs in multimodal problems [21, 22]. Hence, genetic algorithm should be implemented many times to obtain the best solution.

Eberhart and Kennedy developed the particle swarm optimization algorithm in 1995. The algorithm has resemblance to genetic algorithm. The algorithm is started by a set random solutions in a solution space. Then the algorithm searches optimal solution by updating the solutions. However, there are no crossover and mutation process in the particle swarm optimization algorithm. In the algorithm, potential solutions (particles) are updated in the solution space by following the current best solution [4].

In the particle swarm optimization method, a solution is represented by position of a particle. We start the particle swarm optimization by randomly selecting initial solutions in a solution space. Then, we evaluate fitness function of current position. We update the local best position of a particle whenever fitness value of current particle is better than the previous best value. We update the global best based on the best fitness value of all particles. Here are the steps of particle swarm optimization algorithm [4, 23]:

(1) Calculate fitness value of every particle. The fitness function is related to the objective function. In a minimization problem, the smaller objective function the the greater fitness value will be.
(2) Update position of local best and global best.
(3) Update particle velocity by using the equation

$$v_i(t + 1) = wv_i(t) + c_1r_1(lbest(t) - x_i(t)) + c_2r_2(gbest(t) - x_i(t)), \qquad (1)$$

Here $v_i(t)$ and $x_i(t)$ are velocity of particle i and position of particle i at time t, while $lbest(t)$ and $gbest(t)$ are local best and global best position at time t. Parameters r_1 and r_2 represent random number between zero and one with uniform distribution.
(4) Update particle position using the following equation

$$x_i(t + 1) = x_i(t) + v_i(t + 1). \qquad (2)$$

The steps are reiterated some termination condition is met.

In Eq. (1), parameters w, c_1, c_2 are inertia weight, cognitive coefficient and social coefficient respectively. The typical value of w between 0.8 and 1.2, while the typical values of c_1 and c_2 are commonly nearly 2. We can apply velocity clamping to avoid particles from moving very distant outside the solution space. For a solution space restricted by the range $[x_{min}, x_{max}]$, the velocity is limited within the range $[-v_{max}, v_{max}]$ where $v_{max} = m(x_{max} - x_{min})$ for some constant m, $0.1 \leq m \leq 1$. The ending conditions in particle swarm optimization comprises a maximum number of iterations, a number of iterations since the last update of global best solution, or a target fitness value has reached by some particles [23].

In a one dimensional problem, particle swarm optimization converges towards a local optimum for a comparatively wide range of objective functions. In multidimensional problem, it turns out that the swarm might not converge towards a local optimum [24]. Convergence of particle swarm optimization to a local optimum is called premature convergence. The premature convergence is commonly caused by particle velocity decrease in the solution space. Then the particle velocity decrease causes to a total implosion and eventually fitness stagnation of the swarm [25]. Hence, particle swarm optimization method should be implemented many times to obtain the best solution.

A COMPARISON OF CONTINUOUS GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION

In this section, we compared performance of continuous genetic algorithm and particle swarm optimization in parameter estimation of Gompertz growth model. Gompertz growth model is derived from the following Gompertz differential equation

$$\frac{dy}{dt} = ry \ln\left(\frac{K}{y}\right), y(0) = Y_0. \tag{3}$$

Here, $y(t)$ is population size at time t. The exact solution of the Gompertz differential equation in Eq. (3) could be represented as

$$y(t) = \frac{K}{exp(exp(-r(t - t_{inf})))}, \tag{4}$$

where $t_{inf} = \frac{1}{r} \ln\left(\ln\left(\frac{K}{Y_0}\right)\right)$. The Gompertz growth model has three parameters namely intrinsic growth (r), carrying capacity (K), and inflection time (t_{inf}) parameter. From biological point of view, the fastest growth of a population occur at the inflection time.

In this paper, the Gompertz growth model is applied to describe rooster growth where the data is cited from literature [15, 26]. The rooster growth data is shown in the Table 1.

TABLE 1. Means of the rooster weight data (y).

t (days)	y (grams)	t (days)	y (grams)
0	37	42	519.72
3	41.74	45	577.27
6	59.19	48	633.59
9	79.94	51	667.18
12	102.96	54	717.17
15	132.13	57	786.35
18	170.18	71	1069.28
21	206.56	85	1326.49
24	250.71	99	1589.71
27	285.27	113	1859.26
30	324.92	127	2015.44
33	372.83	141	2142.31
36	417.41	155	2220.54
39	469.13	170	2262.63

Since $y(t)$ is the rooster weight at time t, then the carrying capacity parameter (K) could be interpreted as the rooster mature weight or the maximum weight that can be attained by the rooster. Parameters in the Gompertz model (K, r, t_{inf}) are estimated such that the mean absolute percentage error (MAPE)

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left|\frac{y_i - \widehat{y_i}}{y_i}\right| \tag{5}$$

is maximum. Here n is number of observation data.

We applied continuous genetic algorithm and particle swarm optimization to estimate parameters in the Gompertz growth model. Here, optimal parameters in the Gompertz growth model was found from the following search space

$$\Omega = \left\{(K, r, t_{inf}) \in R^3 : K \in [2000, 5000], r \in [0, 0.1], t_{inf} \in [30, 100]\right\}. \tag{6}$$

We applied particle swarm optimization method described in the previous section with the inertia weight parameter $w = 1$, the cognitive coefficient parameter $c_1 = 2$ and the social coefficient parameter $c_2 = 2$ respectively. We also applied continuous genetic algorithm for various mutation rate (m) namely $m = 0.05, 0.1, 0.2, 0.3, 0.4, 0.5$. For

both algorithm, number of population is set to 100 individuals (particles). We applied both methods for 50 trials while for every trial the methods were terminated after 500 iterations. The best estimation results of the particle swarm optimization and the continuous genetic algorithm was presented in the Table 2.

TABLE 2. The best estimation results of the particle swarm optimization and the continuous genetic algorithm.

Methods	K	r	t_{inf}	MAPE
PSO	2468.54	0.023646	60.68	0.039334
GA (m = 0.05)	2445.65	0.023870	60.15	0.039465
GA (m = 0.1)	2435.97	0.023800	60.21	0.039225
GA (m = 0.2)	2437.81	0.023740	60.34	0.039117
GA (m = 0.3)	2468.65	0.023749	60.67	0.039326
GA (m = 0.4)	2444.90	0.023750	60.41	0.039173
GA (m = 0.5)	2468.70	0.023628	60.77	0.039185

m = mutation rate.

Form the Table 2, we found that best result (minimum of MAPE) of the continuous genetic algorithm and the particle swarm optimization method were not significantly differ. The mean average percentage error for the Gompertz growth model obtained from the methods were around 3.9 %. The results indicated that the Gompertz growth model could be applied to describe rooster growth dynamic. It also indicated that particle swarm optimization and continuous genetic algorithm were successfully implemented in parameter estimation of the Gompertz growth model.

Particle swarm optimization and continuous genetic algorithm are essentially probabilistic methods. Hence, the two methods will generally produce different optimal/sub optimal solution in every trial/calculation/experiment. Statistics of the MAPE of the both methods was presented in the Table 3.

TABLE 3. Statistics of Mean Absolute Percentage Error.

Methods	Number of trials	Average of MAPE	Standard deviation	Minimum	Median	Maximum	p-value
PSO	50	0.045661 [a]	0.003743	0.039334	0.045548	0.058081	
GA (m = 0.05)	50	0.092665 [b]	0.051349	0.039465	0.077008	0.267170	
GA (m = 0.1)	50	0.093638 [b]	0.044163	0.039225	0.087674	0.202533	p-value <
GA (m = 0.2)	50	0.084986 [b]	0.037714	0.039117	0.080257	0.244075	0.0005
GA (m = 0.3)	50	0.098225 [b]	0.054169	0.039326	0.084729	0.270992	
GA (m = 0.4)	50	0.092345 [b]	0.066085	0.039173	0.070386	0.400521	
GA (m = 0.5)	50	0.091441 [b]	0.051345	0.039185	0.077055	0.325060	

m = mutation rate.
[a,b] different superscripts showed significant difference between group at the level 0.05.

From the Table 3, we found that the MAPE average of various mutation rate in the continuous genetic algorithm did not significantly differ. Although the best result (minimum of MAPE) of the continuous genetic algorithm and the particle swarm optimization method were not significantly differ, the MAPE average of the particle swarm optimization method was smaller than the continuous genetic algorithms one. We also found that the MAPE variance of the genetic algorithms were more higher than the particle swarm optimization variance. This results indicate that the particle swarm optimization method was more robust than the continuous genetic algorithm.

CONCLUSIONS

We have implemented particle swarm optimization and continuous genetic algorithm in parameter estimation of the Gompertz growth model. Although the best results of the two models were not significantly differs, we found that the particle swarm optimization method was more robust than the continuous genetic algorithm.

ACKNOWLEDGMENTS

REFERENCES

[1] Tutkun, N., Parameter estimation in mathematical models using the real coded genetic algorithms, Expert Systems with Applications 36, pp. 3342-3345, 2009.

[2] Haupt, R. L. & Haupt, S. E., Practical genetic algorithms, 2nd ed., John Wiley & Sons, 10-13, 2004.

[3] Eberhart R. & Kennedy, J. A new optimizer using particle swarm theory, Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 3943, 1995.

[4] Kuo, R. J., Wang, M. J. & Huang, T. W., An application of particle swarm optimization algorithm to clustering analysis, Soft Computing 15, pp. 533542, 2011.

[5] Salerno, J., Using the particle swarm optimization technique to train a recurrent neural model, Proceedings of the Ninth IEEE International Conference on Tools with Artificial Intelligence, pp 4549, 1997.

[6] Zhang, C., Shao, H. & Li, Y., Particle swarm optimization for evolving artificial neural network, IEEE international conference on systems, man and cybernetics, 24872490, 2000.

[7] Koay, C.A. & Srinivasan, D., Particle swarm optimization-based approach for generator maintenance scheduling. In: Proceedings of the 2003 IEEE swarm intelligence symposium, 167173, 2003.

[8] Weijun, X., Zhiming, W., Wei, Z. & Genke, Y., A new hybrid optimization algorithm for the job-shop scheduling problem, Proceedings of the 2004 American Control Conference, 55525557, 2004.

[9] Liao, C. J., Chao-Tang Tseng, Luarn, P., A discrete version of particle swarm optimization for flowshop scheduling problems, Computers and Operations Research Vol. 34, No. 10, pp. 3099-3111, 2017.

[10] Wang, K.P., Huang, L., Zhou, C.G. & Pang, W., Particle swarm optimization for traveling salesman problem, 2003 International Conference on Machine Learning and Cybernetics, 15831585, 2003.

[11] Wu, B., Yanwei, Z., Yaliang, M., Hongzhao, D. & Weian, W., Particle swarm optimization method for vehicle routing problem, Fifth World Congress on Intelligent Control and Automation, 22192221, 2004.

[12] Xiao, J.M., Li, J.J., Wang, X.H., Modified particle swarm optimization algorithm for vehicle routing problem, Jisuanji Jicheng Zhizao Xitong (Computer Integrated Manufacturing Systems) Vol. 11 No. 4, pp. 577-581, 2005.

[13] Niu, B., Duan, Q., Liu, J., Tan, L. , Liu, Y., A population-based clustering technique using particle swarm optimization and k-means, Natural Computing Vol 16. No. 1: 45-59, 2017.

[14] Roush, W.B. & Branton, S. L., A Comparison of Fitting Growth Models with a Genetic Algorithm and Nonlinear Regression, Poultry Science 84, pp. 494502, 2005.

[15] Windarto, Indratno, S. W., Nuraini, N., & Soewono, E., A comparison of binary and continuous genetic algorithm in parameter estimation of a logistic growth model, AIP Conference Proceedings 1587, 139142, 2014.

[16] Windarto, An implementation of continuous genetic algorithm in parameter estimation of predator-prey model, AIP Conference Proceedings 1718, 2016.

[17] Yang, F., Zhang, C., Sun, T., Comparison of Particle Swarm Optimization and Genetic Algorithm for HMM training, Proceedings - International Conference on Pattern Recognition 2008, 2008.

[18] Wang, H., Liu, D., Meng, L., Liu, L., Particle swarm optimization and genetic algorithm for a relativistic backward wave oscillator, Jisuan Wuli (Chinese Journal of Computational Physics) Vol. 31, No. 4, pp. 479-485, 2014.

[19] Islam, M.J., Tanveer, M.S.R., Akhand, M.A.H., A comparative study on prominent nature inspired algorithms for function optimization, 5th International Conference on Informatics, Electronics and Vision (ICIEV) 2016, 7760112, pp. 803-808, 2016.

[20] Mitchell, M., An Introduction to Genetic Algorithms, MIT Press, 1-10, 1999.

[21] Nicoara, E.S., Mechanisms to Avoid the Premature Convergence of Genetic Algorithms, Buletinul Universiţăţii Petrol - Gaze din Ploieşti 61(1):87-96, 2009.

[22] Malik, S., Wadhwa, S., Preventing Premature Convergence in Genetic Algorithm Using DGCA and Elitist Technique, International Journal of Advanced Research in Computer Science and Software Engineering 4(6): 410-418, 2014.

[23] Rini, D.P., Shamsuddin, S.M., Yuhaniz, S.S., Particle Swarm Optimization: Technique, System and Challenges, International Journal of Computer Applications 14(1), 2011.

[24] Schmitt, B.I., Convergence Analysis for Particle Swarm Optimization, FAU University Press, Erlangen, 2015.

[25] Ye, H., Luo, W., Li, Z., Convergence Analysis of Particle Swarm Optimizer and Its Improved Algorithm Based on Velocity Differential Evolution, Computational Intelligence and Neuroscience Volume 2013, Article ID 384125, 2013.

[26] Aggrey, S.E., Comparison of Three Nonlinear and Spline Regression Models for Describing Chicken Growth Curves, Poultry Science 81:1782-1788, 2002.

also developed by scimago: **SCIMAGO INSTITUTIONS RANKINGS**

SJR

Scimago Journal & Country Rank

Enter Journal Title, ISSN or Publisher Name

Home   Journal Rankings   Country Rankings   Viz Tools   Help   About Us

AIP Conference Proceedings

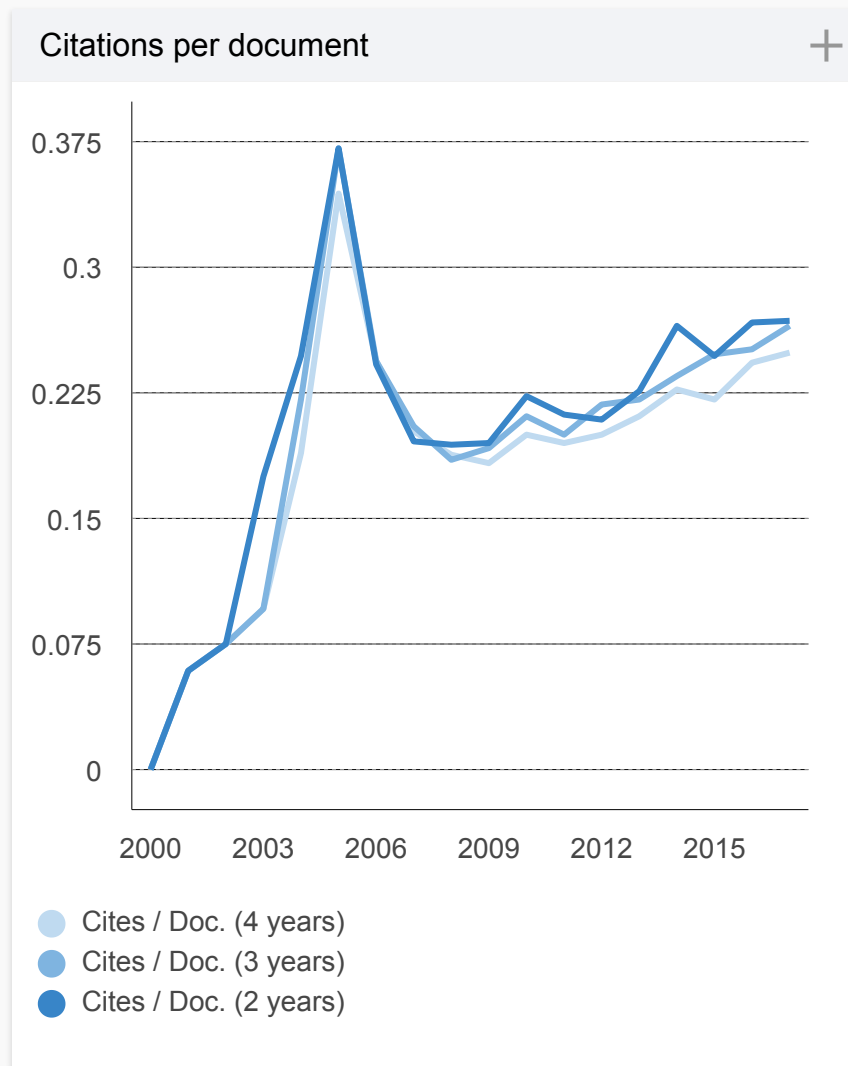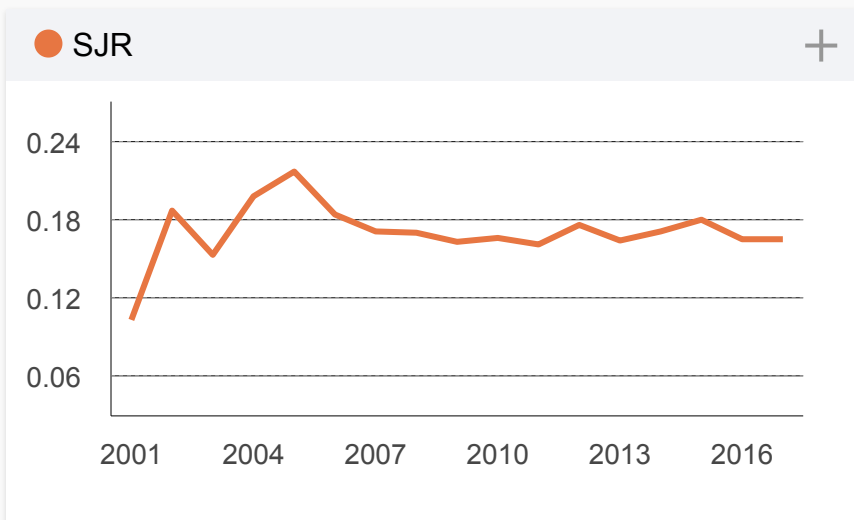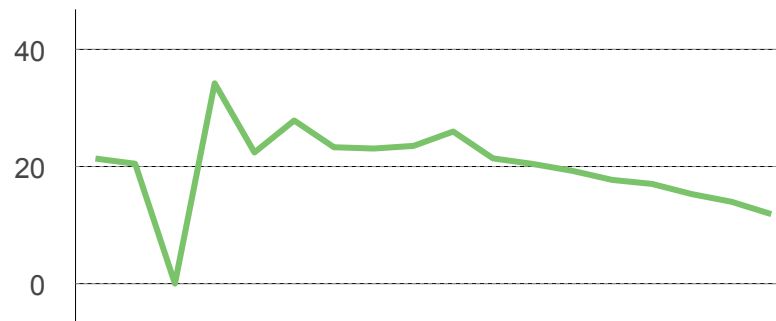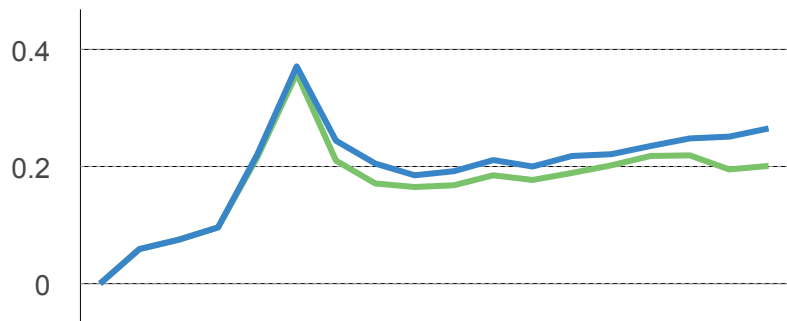|	
Country	United States - SIR Ranking of United States
Subject Area and Category	Physics and Astronomy Physics and Astronomy (miscellaneous)
Publisher	
Publication type	Conferences and Proceedings
ISSN	0094243X
Coverage	1983-1984, 2005-ongoing
Scope	AIP Conference Proceedings report findings presented at many of the most important scientific meetings around the world. Published proceedings are valuable as topical status reports providing quick access to information before it appears in the traditional journal literature.

54

H Index

Homepage

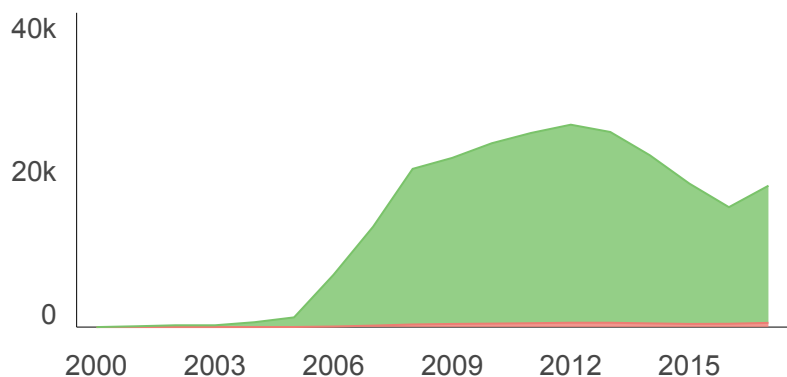How to publish in this journal

Contact

SJR

Total Cites   Self-Cites

Citations per document

Cites / Doc. (4 years)
Cites / Doc. (3 years)
Cites / Doc. (2 years)

External Cites per Doc   Cites per Doc
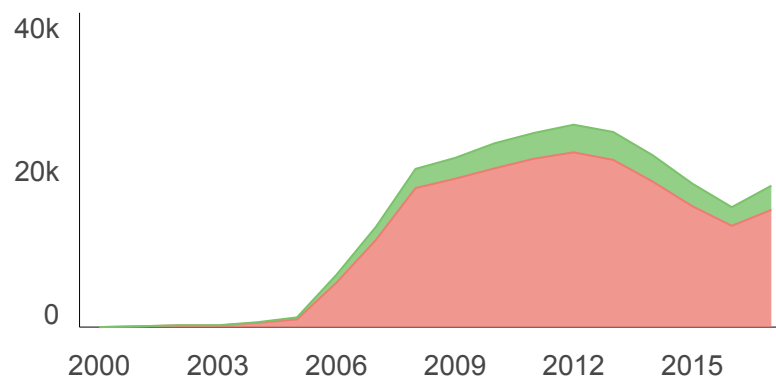
% International Collaboration
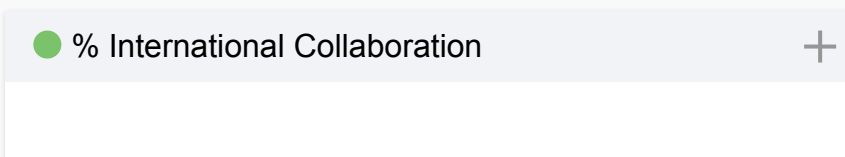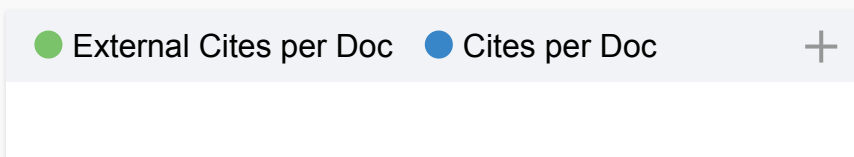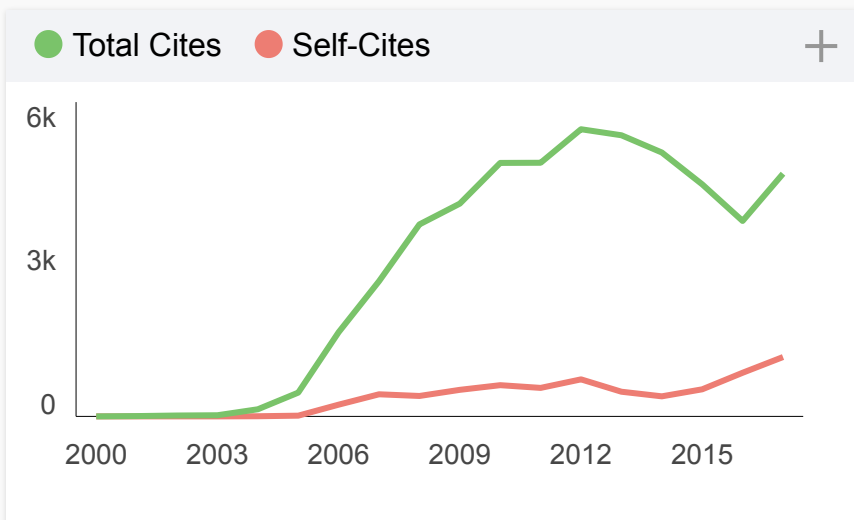
Citable documents · **Non-citable documents** +

Cited documents · **Uncited documents** +

AIP Conference Proceedings

Not yet assigned quartile

SJR 2017
0.16

powered by scimagojr.com

← Show this widget in your own website

Just copy the code below and paste within your html code:

`<a href="https://www.scimag`

H

Hassan Abdulhadi 6 months ago

I ASKE ABOUT AIP CONFERENCE PROCEEDINGS WITHIN SCOPUS OR THOMSON
REUTERS
WITH BEST WISHES

reply

H

Hassan Abdulhadi 4 months ago

I ASKE ABOUT AIP CONFERENCE PROCEEDINGS WITHIN SCOPUS OR THOMSON
REUTERS
WITH BEST WISHES

Elena Corera 6 months ago

Dear Hassan,

thank you for your request, all the journals included in SJR are indexed in Scopus. Elsevier /
Scopus is our data provider.

Best Regards,

SCImago Team

Tarik   6 months ago

Dear. Elena

Hi

Please can we concedar AIP conference proceeding as journal .What i mean ,the publication type
could be

journal of AIP conference proceedings .

Best regards

TArik AlOmran

reply

Elena Corera   6 months ago

Dear Tarik,

thank you very much for your comment. Unfortunately, we cannot help you with your request, we suggest you contact journal's editorial staff so they could inform you more deeply. You can find contact information in SJR website https://www.scimagojr.com

Best regards,
SCImago Team

Dunia 6 months ago

dear
did the AIP conference (TMREES 18)have Thomson roeters or scopus or SJR Rank or not?

reply

Elena Corera 6 months ago

Dear Dunia,

thank you very much for your comment. SCImago Journal & Country Ranks shows all the journal's vailable information in Open Access If you do not locate the journal in the search engine, Scopus / Elsevier has not provided us those data.

Best Regards,

SCImago Tea

Budi Adiperdana   8 months ago

Dear Admin,

Could you please add the Quartile Rank for AIP Conference Proceedings

Best regards,

Budi

reply

Elena Corera   8 months ago

Dear Budi, for Conferences and Proceedings the SJR is not calculated. Best Regards,

SCImago Team

Leave a comment

Name

Email

(will not be published)

[ ] I'm not a robot

reCAPTCHA
Privacy - Terms

Submit

The users of Scimago Journal & Country Rank have the possibility to dialogue through comments linked to a specific journal. The purpose is to have a forum in which general doubts about the processes of publication in the journal, experiences and other issues derived from the publication of papers are resolved. For topics on particular articles, maintain the dialogue through the usual channels with your editor.

Developed by: Powered by:

Scimago
Lab

Scopus

Follow us on @ScimagoJR

Scimago Lab, Copyright 2007-2018. Data Source: Scopus®

EST MODUS IN REBUS
Horatio (Satire 1,1,106)