RESEARCH ARTICLE

Performance Analysis of Simulated Annealing and Genetic Algorithm on systems of linear equations [version 1; peer review: 1 approved, 1 not approved]

Md. Shabiul Islam¹, Most Tahamina Khatoon², Kazy Noor-e-Alam Siddiquee^{id}^{1,2}, Wong Hin Yong¹, Mohammad Nurul Huda³

¹Multimedia University, Persiaran Multimedia, Cyberjaya, Selangor, 63100, Malaysia

²Department of Computer Science and Engineering, University of Science and Technology Chittagong, Foy's Lake, Khulshi, Post Box-1079, Chattogram-4202, Bangladesh

³Department of Computer Science and Engineering, United International University, Dhaka 1212, Bangladesh

V1 First published: 20 Dec 2021, 10:1297
<https://doi.org/10.12688/f1000research.73581.1>
 Latest published: 20 Dec 2021, 10:1297
<https://doi.org/10.12688/f1000research.73581.1>

Abstract

Problem solving and modelling in traditional substitution methods at large scale for systems using sets of simultaneous equations is time consuming. For such large scale global-optimization problem, Simulated Annealing (SA) algorithm and Genetic Algorithm (GA) as meta-heuristics for random search technique perform faster. Therefore, this study applies the SA to solve the problem of linear equations and evaluates its performances against Genetic Algorithms (GAs), a population-based search meta-heuristic, which are widely used in Travelling Salesman problems (TSP), Noise reduction and many more. This paper presents comparison between performances of the SA and GA for solving real time scientific problems. The significance of this paper is to solve the certain real time systems with a set of simultaneous linear equations containing different unknown variable samples those were simulated in Matlab using two algorithms-SA and GA. In all of the experiments, the generated random initial solution sets and the random population of solution sets were used in the SA and GA respectively. The comparison and performances of the SA and GA were evaluated for the optimization to take place for providing sets of solutions on certain systems. The SA algorithm is superior to GA on the basis of experimentation done on the sets of simultaneous equations, with a lower fitness function evaluation count in MATLAB simulation. Since, complex non-linear systems of equations have not been the primary focus of this research, in future, performances of SA and GA using such equations will be addressed. Even though GA maintained a relatively lower number of average generations than SA, SA still managed to outperform GA with a reasonably lower fitness function evaluation count. Although SA sometimes converges slowly, still it is efficient for

Open Peer Review

Approval Status

	1	2
version 1 20 Dec 2021	 view	 view
1. Aloke Kumar Saha , University of Asia Pacific, Dhaka, Bangladesh		
2. Andranik S. Akopov , HSE University, Moscow, Russian Federation		
Any reports and responses or comments on the article can be found at the end of the article.		

solving problems of simultaneous equations in this case. In terms of computational complexity, SA was far more superior to GAs.

Keywords

Optimization, Simulated Annealing, Simultaneous Equations, Genetic Algorithms

This article is included in the **Research Synergy** Foundation gateway.

Corresponding authors: Md. Shabiul Islam (shabiul.islam@mmu.edu.my), Kazy Noor-e-Alam Siddiquee (knas11@gmail.com)

Author roles: **Islam MS:** Conceptualization, Supervision, Writing – Review & Editing; **Khatoon MT:** Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – Original Draft Preparation; **Siddiquee KNeA:** Conceptualization, Formal Analysis, Methodology, Project Administration, Writing – Original Draft Preparation, Writing – Review & Editing; **Yong WH:** Writing – Review & Editing; **Huda MN:** Supervision, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2021 Islam MS *et al.* This is an open access article distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Islam MS, Khatoon MT, Siddiquee KNeA *et al.* **Performance Analysis of Simulated Annealing and Genetic Algorithm on systems of linear equations [version 1; peer review: 1 approved, 1 not approved]** F1000Research 2021, 10:1297 <https://doi.org/10.12688/f1000research.73581.1>

First published: 20 Dec 2021, 10:1297 <https://doi.org/10.12688/f1000research.73581.1>

Introduction

The Simulated Annealing (SA) algorithm is a well-known meta-heuristic for solving problems that require best or optimal results. This algorithm can easily be implemented using some programming language and as such it has gained wide acceptance in the arena of computational and scientific research over the past few decades.

On the other hand, in Science and Engineering, systems of equations or simultaneous equations involving unknown variables are applied to solve real-life problems through mathematical modelling, in solving simultaneous equations, the two most common methods are the Substitution and the Elimination method.¹ In one way, Substitution works nicely, especially for a system of two equations with two variables. However, for larger cases, such as a system of three equations with three variables, the Substitution method can become a time-consuming process. On the contrary, the Elimination method can be an easier alternative to solving equations involving three equations with three variables. Still, with an increase in the number of equations and variables, it can become a difficult and lengthy process. Besides, a numerical method called the Gaussian Elimination method is not always capable of finding good quality solutions for complex systems of equations.² Hence, to be able to solve simultaneous equations using the SA approach could make the process of finding possible solution sets much easier, faster, and convenient.

The purpose of this research is to study the performances of the SA algorithm while comparing it with Genetic Algorithms (GAs), in solving simultaneous linear equations.

Literature review

Genetic Algorithms (GAs) have been used to solve simultaneous equations in the past. Abiodun *et al.*³ had implemented a GA to solve simultaneous linear equations. They experimented with seven different systems of equations and have performed a comparative analysis of their work with Gaussian Elimination method. They were successful in solving the equations using GA. In some cases, GA was able to give the best or exact solutions, which was not possible with the Gaussian Elimination method. Besides, their GA was able to find other combinations of solution sets which again was not possible with the existing conventional methods like Substitution, Elimination etc. Their work indicated the fact that GA was superior in its performances against the existing conventional methods of solving simultaneous equations.

A review on Simulated Annealing and its applications

Simulated Annealing (SA) is an approach to solving optimization problems introduced by Kirkpatrick *et al.*⁴ where it uses the property of randomization in its search operations. The idea of the SA algorithm comes from metallurgy where it mimics the metal's re-crystallization in the process of slow cooling of metal which is termed annealing.⁴

In the SA simulation,⁴ initially, the system undergoing optimization is melted at a high temperature. Next, the temperature is decreased slowly to freeze the system. This continues until no further change is possible. The SA algorithm has the capability to escape from local minima by accepting or rejecting new solution candidates based on a probability function.

The basic SA algorithm⁴ has the following procedure:

1. T_0 is the initial temperature that needs to be specified before the initialization of the solution candidate
2. E is the fitness of the candidate that needs to be evaluated
3. Next, the candidate has to be moved to a neighboring solution randomly
4. E' is the new solution whose fitness needs to be evaluated
5. In case of $E' \leq E$, the new solution is accepted
6. Whereas, in case of $E' > E$, the new solution is accepted based on acceptance probability P , as shown in (1) below
7. T stands for temperature that needs to be decreased.
8. The SA search will stop if T is close to zero. Otherwise, it will go back to step 2

$$P = e^{-(E' - E)/T} \quad (1)$$

The temperature is regulated by the following:

$$T(k+1) = \lambda T(k), 0 < \lambda < 1 \quad (2)$$

In equation (2), k is the number of iterations. The symbol λ represents a coefficient that is used to adjust the cooling schedule by modification. The advantage of the SA approach is that it can obtain the global optimum with a greater probability, however, it requires many iterations to converge to a solution.

Some recent applications of SA are discussed below.

Jayabalan *et al.*⁵ have proposed two modified versions of SAs to solve a scheduling problem known as permutation flowshop.

In another work, by Cunha and Marques,⁶ they have developed a trajectory-based SA with newer versions of reannealing and generation techniques that enable better convergence for producing solutions involving diversity and uniformity.

In a recent study,⁷ an approach called mixed-integer programming is discussed with SA proposed as a solution.

Odziemczyk⁸ has introduced a procedure to solve a three-dimensional coordinate transformation problem using a variant of the SA technique.

Grabusts *et al.*⁹ have experimented with the SA method and Travelling Salesman Problems (TSP) to find the shortest routes between destinations. Their study discussed applications of mathematical models that could be utilized to solve real-life problems.

Gallo and Capozzi¹⁰ have used SA to solve a scheduling problem that involves several key parameters like tempering, freezing etc. In one study,¹¹ the authors have used a modified SA procedure to solve a problem related to sample allocation.

In both studies by A. A. N. P. Redi,^{12,13} SA algorithms have been applied in routing problems. In the first study,¹² the objective was to find minimum routes that require less transportation time whereas, in the second study,¹³ the research aimed to reduce the transportation expenses.

In a most recent study in Engineering optimization, Meng¹⁴ tried to discover new parameters pertaining to the creative and cultural domain of product design to improve the performance of the SA algorithm.

Managing tasks in cloud computing environment is huge and daunting and it requires smart handling. Therefore, the study by Fakhrosadat Fanian *et al.*,¹⁵ has contributed towards a better way of managing the scheduling of tasks by developing a new algorithmic approach that takes its inspiration from Firefly and SA algorithms.

Gripon, V *et al.*¹⁶ conducted certain experiments on training weights, which were discrete in nature, using a combination of traditional SA and ideas taken from statistical physics.

P. Aravind *et al.*¹⁷ have used a heuristic method based on SA to overcome the problem of allocating switches within a short time for various networks. Their model was able to solve the problem within a reasonable amount of time for varied network sizes.

Avinash C. Pandey and Dharmveer S. Rajpoot¹⁸ have focused on getting accurate results in the classification of features by combining the techniques of SA and grey wolf optimization.

Jie Zhou *et al.*¹⁹ have introduced an efficient way to increase the life span of largescale wireless sensor networks by using a special version of SA called “Elite Adaptive Simulated Annealing Algorithm (EASA)”.

Czerniachowska, K *et al.*²⁰ have addressed an interesting problem of arranging products on shelves which can be a strenuous task involving retailing decisions. Therefore, they have applied the SA procedure to handle the effective decision making of shelf allocation of products on symmetrical planograms.

In the area of industrial cutting, one study has investigated a problem called multi-objective optimization for irregular objects,²¹ in the field of processing aquatic products, and have developed a SA technique for the problem of squid cutting.

Methods

In this study, both SA and GA have been implemented in MATLAB (Version: R2017a, RRID: SCR_001622) based on the setup of two algorithms stated in Table 1 and Table 2. MATLAB is software that provides functionality to implement algorithms, analyze data, or create models.

Simultaneous equations, in Mathematics, play a very important role in modeling real-life problems. These comprise a set of equations, which is finite in nature and should meet at a common point in space.²² These equations should have functions of at least two variables and can either have a linear or a non-linear characteristic.

According to Gilbert,²³ a system of simultaneous equations, linear in nature, can be written in the following manner:

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \dots + a_{1N}x_N &= y_1 \\a_{21}x_1 + a_{22}x_2 + \dots + a_{2N}x_N &= y_2 \\&\vdots \\a_{N1}x_1 + a_{N2}x_2 + \dots + a_{NN}x_N &= y_N\end{aligned}$$

This equation can be represented using a matrix-vector as follows:

$$Ax = y$$

In this expression, unknown variables are contained in a vector x . The number of unknown variables is represented by N , and constants are represented by a , in a coefficient matrix represented by A .

In this study, both SA and GA have been implemented to find the values of unknowns in the equations.

The data is randomly generated to represent each unknown variable in a system of simultaneous equations. The generated data comprises each candidate solution set in SA and each chromosome in GA. In SA, a single vector of values or a solution set is initially generated, while in a GA, a population of 100 chromosomes is initially generated. In evaluating the fitness of each solution set in SA or a chromosome in GA, the randomly generated vector of values is used to evaluate the equation on the left-hand side (LHS). Next, the values of LHS are checked with the values on the right-hand side (RHS) to see if they are equal. If the evaluated value/values of equation/equations on the LHS are equal to the given value/values on

Table 1. Simulated Annealing parameter setup.

SA parameters
Cooling schedule : Linear $\left(T = 1 - \frac{\text{iteration}}{C + \text{Total No. of iterations}}\right)$
Probability calculation : $P = \begin{cases} 1 & \text{if } E_{\text{new}} < E_{\text{curr}} \\ e^{-(E_{\text{new}} - E_{\text{curr}})/T}, & \text{otherwise} \end{cases}$
Total number of iterations: 1000

Table 2. Genetic algorithm parameter setup.

GA parameters
Population size: 100
Reproduction: 60%
Crossover: 30%
Elitism: 10%
Mutation: 0.1
Total number of generations: 1000

the RHS of the equation/equations, then the solution set/chromosome of the unknowns is considered to be valid or fit to a system of equations.

In order to evaluate the fitness of each solution set in SA or a chromosome in GA, an arbitrary value of 3000 is taken as the start fitness value. For each equation or a system of equations, the absolute value of the difference of the LHS and the RHS is taken. Next, it is subtracted from the start fitness value of 3000, i.e.,

$$\text{DIFFERENCE} = \text{absolute value of (LHS - RHS)} \quad (4)$$

$$\text{FITNESS} = \text{FITNESS} - \text{DIFFERENCE} \quad (5)$$

If the difference is equal to 0, the fitness value remains unchanged (i.e., 3000), otherwise, the fitness decreases. To calculate the energy of each solution set in SA, the fitness value of each solution set is subtracted from this arbitrary value (i.e., 3000), since fitness and energy are opposite to each other in the case of SA. This method of converting fitness to energy using the subtraction method resembles a flip flop operation.

Results

Table 3 above presents the results of SA, solving simultaneous equations. For this study, 10 different systems of equations, involving two, three and four unknown variables, were selected. Each experiment was performed 15 times, to record the average performance of SA. As can be seen in **Table 3**, for the first three systems of equations, involving two variables (Experiment number (No.) 1, 2 and 3), SA was able to give its best performance by solving them within a minimum of 3, 2 and 3 iterations each, while keeping the maximum average at 73 (Experiment No. 3). **Table 1** and **Table 2** below show the parameter setup for SA and GA, respectively.

Again, with equations involving three variables, the algorithm performed relatively well, by solving them within as few as 5 minimum iterations and 57, 80 and 83 iterations on average, (Experiment No. 4, 5 and 6). However, soon after that, the average number of iterations kept increasing, reaching a maximum of 176 iterations (Experiment No. 9). This increasing rate is perhaps an indication of the pattern complexity of solution sets (usually having a combination of both positive and negative numbers).

Also, it is interesting to note that, in experiment No. 8, SA was able to find three different sets of solutions, therefore, making it an efficient method over the other traditional techniques like substitution and elimination, as discussed earlier. While, in experiment No. 10, involving 4 variables, there was a sudden but usual increase in the minimum number of iterations, reaching a maximum of 61 iterations. However, in the long run, SA managed to keep its pace by maintaining a standard average of 150 iterations, relatively lower than in experiment No. 9.

Overall, the SA algorithm was successful in solving the selected systems of equations within a reasonable number of iterations, maintaining a reasonable average of 176 iterations and 176 times fitness evaluation count. **Figures 1–10** gives a glimpse of the performances of the SA algorithm implemented in MATLAB software, showing the various energy levels at each iteration count, until reaching a level of 0 at the target iteration, to find the expected solutions.

Table 4 presents the results of GA solving simultaneous equations. In order to evaluate the performances of SA against GA, the same set of equations from SA were tested with GA. Each experiment was performed 15 times just like in SA.

In the first three experiments (Experiment No. 1, 2 and 3), GA was successful in solving the equations within an average of 2 generations. However, there was a sudden increase in the average no. of generations to 35 in experiment 4 with equations involving 3 variables. Next, in the following experiments from 5 through 8, there was a gradual decline in the number of average generations (i.e., 17, 12, 9 and 3, respectively), indicating the improved performance of the algorithm. In experiment 9, there was a dramatic increase in the average no. of generations reaching a maximum of 196. It is important to note that this sudden increase in the number of generations is consistent with the same experiment performed with SA earlier, where the average no. of iterations reached a maximum of 176 from 120. Perhaps, this increase is an indication of the pattern complexity in solution sets (usually having a combination of both positive and negative numbers), just like in the case of SA. In all the experiments from 1 to 9, the algorithm was able to find solutions within a minimum of 1 or 2 generations, until in experiment 10 (involving equations with 4 unknowns) there was a significant rise in the number of minimum generations to 8 and number of average generations to 50 due to the increase in problem size. However, the algorithm managed to maintain a relatively much lower number of average generations as compared to 196 in experiment 9 which is again consistent with the SA results shown earlier. GA was also able to find three different

Table 3. Results of SA solving simultaneous equations.

Experiment No.	Equations	SA Results: Solution of unknowns	No. of test runs	Minimum No. of iterations to find solution (Best result)	Average No. of iterations to find solution	Average No. of fitness function evaluation
1.	$x_1 + 4x_2 = 6$	$x_1 = 2$	15	3	41	41
	$x_1 + 2x_2 = 4$	$x_2 = 1$				
2.	$2x_1 + 5x_2 = 33$	$x_1 = 4$	15	2	60	60
	$x_1 + 3x_2 = 19$	$x_2 = 5$				
3.	$3x_1 + 2x_2 = 36$	$x_1 = 8$	15	3	73	73
	$5x_1 + 4x_2 = 64$	$x_2 = 6$				
4.	$10x_1 + x_2 + x_3 = 12$	$x_1 = 1$	15	5	57	57
	$2x_1 + 10x_2 + x_3 = 13$	$x_2 = 1$				
	$2x_1 + 2x_2 + 10x_3 = 14$	$x_3 = 1$				
5.	$2x_1 + x_2 + 3x_3 = 13$	$x_1 = 1$	15	5	80	80
	$x_1 + 5x_2 + x_3 = 14$	$x_2 = 2$				
	$3x_1 + x_2 + 4x_3 = 17$	$x_3 = 3$				
6.	$x_1 + 2x_2 + 3x_3 = 6$	$x_1 = 1$	15	28	83	83
	$2x_1 + 4x_2 + x_3 = 7$	$x_2 = 1$				
	$3x_1 + 2x_2 + 9x_3 = 14$	$x_3 = 1$				
7.	$x_1 + x_2 + x_3 = 6$	$x_1 = 1$	15	15	101	101
	$x_1 - x_2 + x_3 = 2$	$x_2 = 2$				
	$x_1 + 2x_2 - x_3 = 2$	$x_3 = 3$				
8.	$x_1 + 2x_2 + 3x_3 = 14$	$x_1 = 1$	15	12	120	120
	$x_1 + x_2 + x_3 = 6$	$x_2 = 2$				
	$3x_1 + 2x_2 + x_3 = 10$	$x_3 = 3$				
		$x_1 = 2$				
		$x_2 = 0$				
		$x_3 = 4$				
		$x_1 = 0$				
		$x_2 = 4$				
		$x_3 = 2$				
9.	$3x_1 + 2x_2 - x_3 = 1$	$x_1 = 1$	15	14	176	176
	$2x_1 - 2x_2 + 4x_3 = -2$	$x_2 = -2$				
	$-x_1 + 0.5x_2 - x_3 = 0$	$x_3 = -2$				
10.	$4x_1 + 3x_2 + 2x_3 + x_4 = 10$	$x_1 = 1$	15	61	150	150
	$3x_1 + 2x_2 + x_3 + 4x_4 = 6$	$x_2 = 0$				
	$2x_1 + x_2 + 4x_3 + 3x_4 = 14$	$x_3 = 3$				
	$x_1 + 4x_2 + 3x_3 + 2x_4 = 10$	$x_4 = 0$

Figure 1. Experiment No. 1, minimum no. of iterations = 3.

Figure 2. Experiment No. 2, minimum no. of iterations = 2.

sets of solutions in experiment 8 just like SA, thus making both the optimization techniques in general, more efficient over the traditional methods like substitution and elimination, as mentioned earlier.

Even though GA was able to solve all the equations just like SA, within a maximum average of 196 generations, its comparison with SA does make a huge difference since the maximum average number of iterations for SA stand at 176, which is lesser than in GA. Moreover, in SA, a single iteration means just 1 solution set, whereas, in GA, a single generation means a whole population of 100 chromosomes or solution sets (taken for this study). Therefore, the average fitness function evaluation for each solution set in the case of SA stands at 176 times, while in GA it stands at $196 \times 100 = 19600$ times, where the fitness evaluation takes place for the entire population of chromosomes in each generation, thus

Figure 3. Experiment No. 3, minimum no. of iterations = 3.

Figure 4. Experiment No. 4, minimum no. of iterations = 5.

making it computationally expensive unlike in SA where the fitness evaluation takes place for just one solution set at each iteration.

Figures 11-14 give a glimpse of the performances of the GA algorithm, implemented in MATLAB software, showing the maximum and the average population fitness of each generation until it reaches a maximum fitness of 3000 (a defined arbitrary fitness value) at the target generation and finds the expected solution sets.

Figure 5. Experiment No. 5, minimum no. of iterations = 5.

Figure 6. Experiment No. 6, minimum no. of iterations = 28.

Discussion

This research successfully applied Simulated Annealing (SA) algorithm and evaluated its performances against Genetic Algorithms (GAs). During this experimental study, the 10 different systems of simultaneous linear equations involving 2, 3 and 4 variables were solved.

In all the experiments, an initial solution set or a population of solution sets were randomly generated in both the SA and the GA, for the optimization to take place.

In the case of SA, the linear cooling schedule was used to control the temperature. The linear cooling schedule was effective in successfully finding all the solutions. Further, slow cooling by adding a constant ($C = 50$ and $C = 100$) was also tried in some cases but that did not make a big difference in the number of iterations to find solutions.

Figure 7. Experiment No. 7, minimum no. of iterations = 15.

Figure 8. Experiment No. 8, minimum no. of iterations = 12.

In the case of GA, a population size of 100, a reproduction rate of 60%, a crossover of 30% and elitism of 10% were used. The mutation rate was kept at 0.1.

During the experimentation phase, the following observations were made:

- While solving simultaneous equations, both SA and GA were equally able to solve them and resulted in more than one set of linear solutions for respective systems of those equations. For instance, in experiment 8 (Table 3 and Table 4), for both SA and GA, there have been observed three different sets of perfect solutions, which were a perfect fit on those equations. Thus, indicating the optimization technique to be efficient over the other conventional methods like the Gaussian Elimination Method etc.

Figure 9. Experiment No. 9, minimum no. of iterations = 14.

Figure 10. Experiment No. 10, minimum no. of iterations = 61.

- Even though GA maintained a relatively lower number of average generations than SA, SA still managed to outperform GA with a reasonably lower fitness function evaluation count and thereby proved to be far more computationally efficient than GA.
- Even though SA converges slowly sometimes, it can still be regarded as an efficient technique for solving problems that require optimization.
- In terms of computational complexity, the SA algorithm proved to be far more superior to GAs.

Table 4. Results of GA solving simultaneous equations.

Experiment No.	Equations	GA Results: Solution of unknowns	No. of test runs	Minimum No. of generations to find solution (Best result)	Average No. of generations to find solution	Average No. of fitness function evaluation
1.	$x_1 + 4x_2 = 6$ $x_1 + 2x_2 = 4$	$x_1 = 2$ $x_2 = 1$	15	1	2	200
2.	$2x_1 + 5x_2 = 33$ $x_1 + 3x_2 = 19$	$x_1 = 4$ $x_2 = 5$	15	1	2	200
3.	$3x_1 + 2x_2 = 36$ $5x_1 + 4x_2 = 64$	$x_1 = 8$ $x_2 = 6$	15	1	2	200
4.	$10x_1 + x_2 + x_3 = 12$ $2x_1 + 10x_2 + x_3 = 13$ $2x_1 + 2x_2 + 10x_3 = 14$	$x_1 = 1$ $x_2 = 1$ $x_3 = 1$	15	2	35	3500
5.	$2x_1 + x_2 + 3x_3 = 13$ $x_1 + 5x_2 + x_3 = 14$ $3x_1 + x_2 + 4x_3 = 17$	$x_1 = 1$ $x_2 = 2$ $x_3 = 3$	15	2	17	1700
6.	$x_1 + 2x_2 + 3x_3 = 6$ $2x_1 + 4x_2 + x_3 = 7$ $3x_1 + 2x_2 + 9x_3 = 14$	$x_1 = 1$ $x_2 = 1$ $x_3 = 1$	15	1	12	1200
7.	$x_1 + x_2 + x_3 = 6$ $x_1 - x_2 + x_3 = 2$ $x_1 + 2x_2 - x_3 = 2$	$x_1 = 1$ $x_2 = 2$ $x_3 = 3$	15	1	9	900
8.	$x_1 + 2x_2 + 3x_3 = 14$ $x_1 + x_2 + x_3 = 6$ $3x_1 + 2x_2 + x_3 = 10$	$x_1 = 1$ $x_2 = 2$ $x_3 = 3$ $x_1 = 2$ $x_2 = 0$ $x_3 = 4$ $x_1 = 0$ $x_2 = 4$ $x_3 = 2$	15	1	3	300
9.	$3x_1 + 2x_2 - x_3 = 1$ $2x_1 - 2x_2 + 4x_3 = -2$ $-x_1 + 0.5x_2 - x_3 = 0$	$x_1 = 1$ $x_2 = -2$ $x_3 = -2$	15	2	196	19600
10.	$4x_1 + 3x_2 + 2x_3 + x_4 = 10$ $3x_1 + 2x_2 + x_3 + 4x_4 = 6$ $2x_1 + x_2 + 4x_3 + 3x_4 = 14$ $x_1 + 4x_2 + 3x_3 + 2x_4 = 10$	$x_1 = 1$ $x_2 = 0$ $x_3 = 3$ $x_4 = 0$	15	8	50	5000

Figure 11. Experiment No. 4, minimum no. of generations = 2.

Figure 12. Experiment No. 5, minimum no. of generations = 2.

Figure 13. Experiment No. 9, minimum no. of generations = 2.

Figure 14. Experiment No. 10, minimum no. of generations = 8.

Conclusions

This study evaluated the performances of the Simulated Annealing (SA) algorithm over Genetic algorithms (GAs) in solving simultaneous linear equations.

The SA algorithm was applied to solve simultaneous linear equations and evaluated its performances against GAs.

The limitation of this paper is that the problem of the real-time system has been solved with a set of only linear equations instead of both linear and non-linear equations.

However, in future, the focus of this research experiment with SA will be on providing solutions with non-linear systems of equations. Furthermore, as part of future work, Harmony Search²⁴ as a recent optimization technique can be used to solve systems of equations based on music improvisation.

Data availability

No data are associated with this article

Ethical approval and consent

No humans or animals were included in this study.

Author contribution

Md. Shabiul Islam: Conceptualization, Supervision, Writing-Review & Editing

Most Tahamina Khatoon: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing-Original Draft

Kazy Noor-e-Alam Siddiquee: Conceptualization, Formal Analysis, Methodology, Project Administration, Writing-Original Draft Preparation, Writing-Review & Editing

Wong Hin Yong: Writing-Review & Editing

Mohammad Nurul Huda: Supervision, Writing-Review & Editing

References

1. Ugbooduma, Samuel S: **Students' Preference of Method of Solving Simultaneous Equations**. *Glob. J. Educ. Res.* 2012; **11**(2) [Publisher Full Text](#)
2. Grcar JF: **Mathematicians of Gaussian Elimination**. *Notice of the AMS*. 2011; **58**(6) [Reference Source](#)
3. Abiodun IM, Olawale LN, Adebawale AP: **The Effectiveness of Genetic Algorithm in Solving Simultaneous Equations**. *Int. J. Comput. Appl.* 2011; **14**(8): 38–41. [Publisher Full Text](#)
4. Kirkpatrick S, Gelatt JD, Vecchi MP: **Optimization by simulated annealing**. *Science*. 1983; **220**(4598): 671–680. [Publisher Full Text](#)
5. Jayabalan K, Pugazhendhi S, Rajendran C, et al.: **Simulated annealing algorithms to minimise the completion time variance of jobs in permutation flowshops**. *Int. J. Ind. Syst. Eng.* 2019; **31**: 425. [Publisher Full Text](#)
6. Cunha M, Marques J: **A New Multiobjective Simulated Annealing Algorithm—MOSA-GR: Application to the optimal design of water distribution networks**. *Water Resour. Res.* 2020; **56**: e2019WR025852. Wiley Online Library ADS Web of Science®Google Scholar. [Publisher Full Text](#)
7. Yang W, Cheng W: **A Mathematical Model and a Simulated Annealing Algorithm for Balancing Multi-manned Assembly Line Problem with Sequence-Dependent Setup Time**. *Math. Probl. Eng.* 2020; **2020**: 1–16. Article ID 8510253. [Publisher Full Text](#)
8. Odziemczyk W: **Application of Simulated Annealing Algorithm for 3D Coordinate Transformation Problem Solution**. *Open Geosci.* 2020; **12**: 491–502. [Publisher Full Text](#)
9. Grabusts P, Musatovs J, Golenkov V: **The application of simulated annealing method for optimal route detection between objects**. *Procedia Comput. Sci.* 2019; **149**: 95–101. 18770509. [Publisher Full Text](#) | [Reference Source](#)
10. Gallo C, Capozzi V: **A Simulated Annealing Algorithm for Scheduling Problems**. *J. Appl. Math. Phys.* 2019; **07**: 2579–2594. [Publisher Full Text](#)
11. O'Luing M, Prestwich S, ArmaganTarim S: **A Simulated Annealing Algorithm for Joint Stratification and Sample Allocation Designs**. *arXiv e-prints*. 2020.
12. Redi AANP: **Simulated annealing algorithm for solving the capacitated vehicle routing problem: a case study of pharmaceutical distribution**. *J. Sist. Manaj. Ind.* Jul. 2020; **4**(1): 41–49. [Publisher Full Text](#)
13. Redi AANP, Jewpanya P, Kurniawan AC, et al.: **A Simulated Annealing Algorithm for Solving Two-Echelon Vehicle Routing Problem with Locker Facilities**. *Algorithms*. Sep. 2020; **13**(9): 218. [Publisher Full Text](#)
14. Meng X: **Optimization of Cultural and Creative Product Design Based on Simulated Annealing Algorithm**. *Complexity*. 2021; **2021**: 1–10. Article ID 5538251. [Publisher Full Text](#)
15. Fanian F, Bardsiri VK, Shokouhifar M: **A New Task Scheduling Algorithm using Firefly and Simulated Annealing Algorithms in Cloud Computing**. *Int. J. Adv. Comput. Sci. Appl.* 2018; **9**(2). [Publisher Full Text](#)
16. Gripon V, Löwe M, Vermet F: **Some Remarks on Replicated Simulated Annealing**. *J. Stat. Phys.* 2021; **182**: 51. [Publisher Full Text](#)
17. Aravind P, Saradhi Varma GP, Prasad Reddy PVGD: **Simulated annealing based optimal controller placement in software defined networks with capacity constraint and failure awareness**. *J. King Saud Univ. - Comput. Inf. Sci.* 2021; 1319-1578. [Publisher Full Text](#)
18. Pandey AC, Rajpoot DS: **Feature Selection Method Based on Grey Wolf Optimization and Simulated Annealing**. *Recent Adv. Comput. Sci. Commun.* 2021; **14**(2): 635–646. [Publisher Full Text](#)
19. Zhou J, Jia W, Liu M, Xu M: **Elite Adaptive Simulated Annealing Algorithm for Maximizing the Lifespan in LWSNs**. *J. Sensors*. 2021; **2021**: 11. Article ID 9915133. [Publisher Full Text](#)
20. Czerniachowska K, Hernes M: **Simulated Annealing Hyper-Heuristic for a Shelf Space Allocation on Symmetrical Planograms Problem**. *Symmetry*. 2021; **13**: 1182. [Publisher Full Text](#)
21. Liu S, Wang H, Cai Y: **Research on Fish Slicing Method Based on Simulated Annealing Algorithm**. *Appl. Sci.* 2021; **11**: 6503. [Publisher Full Text](#)
22. Nordin N, Tengah K, Shahrill M, Tan A, Leong E: **Using Visual Representations as an Alternative in Teaching Simultaneous Equations**. 2017. [Publisher Full Text](#)
23. Gilbert S: *Linear Algebra and Its Applications*. Pacific Grove: Brooks Cole; 2007.
24. Geem Z, Kim J: **A new heuristic optimization algorithm: Harmony search**. *Simulation*. 2001; **76**(2): 60–68. [Publisher Full Text](#)

Open Peer Review

Current Peer Review Status:

Version 1

Reviewer Report 14 January 2022

<https://doi.org/10.5256/f1000research.77242.r116261>

© 2022 Akopov A. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Andranik S. Akopov

Department of Business Informatics, HSE University, Moscow, Russian Federation

The problem statement described in the submitted paper does not make sense. Genetic Algorithms are applied to solve more complex optimization problems (e.g., non-convex, multiobjective optimization, etc.). Thus, the proposed method looks like an application of complex heuristic algorithms to solve very simple optimization problems related to the systems of linear equations. The second important concern is the examined algorithms (i.e., GA and Simulated Annealing) are standard and do not have any significant improvements. I did not find any comparisons of studied algorithms with other GAs (e.g., RCGAs, parallel GAs, Hybrid GAs) in performance. Moreover, the detailed description of used methods (i.e., SA and GA) with the use of pseudocode is absent. I found the reference to the MATLAB framework only without any technical details of the used methods and procedures. That also confirms the examined algorithms are standard and don't have any improvements. Based on these facts, I cannot recommend this article for indexing.

Is the work clearly and accurately presented and does it cite the current literature?

No

Is the study design appropriate and is the work technically sound?

No

Are sufficient details of methods and analysis provided to allow replication by others?

Partly

If applicable, is the statistical analysis and its interpretation appropriate?

No

Are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions drawn adequately supported by the results?

Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Genetic Algorithms, Neural Networks, and Deep Learning, Fuzzy Logic, Artificial Intelligence, Multiagent systems

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Author Response 14 Jan 2022

KAZY NOOR E ALAM SIDDIQUEE, Multimedia University, Cyberjaya, Malaysia

Dear Reviewer

Thank you very much for your formatted review.

The work is primarily focusing on linear systems equation prioritizing 10 equations as a sample. Other variants of SA are our future milestones to go for experimentation.

With regards,
Kazy

Competing Interests: No competing interests were disclosed.

Reviewer Report 14 January 2022

<https://doi.org/10.5256/f1000research.77242.r116271>

© 2022 Saha A. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aloke Kumar Saha

University of Asia Pacific, Dhaka, Bangladesh

This is a good work. However there are some observations from my end:

1. Authors mentioned that no data are associated with this article, but they provided some examples. I think the number of examples should be increased.
2. Authors did not mentioned the number of variables for solving the problems. They should mention if there are any restrictions about the number of variables.
3. This is highly mathematical, but the problem of the real-time system should be developed with

a set of linear and non-linear equations.

Is the work clearly and accurately presented and does it cite the current literature?

Yes

Is the study design appropriate and is the work technically sound?

Yes

Are sufficient details of methods and analysis provided to allow replication by others?

Yes

If applicable, is the statistical analysis and its interpretation appropriate?

Yes

Are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions drawn adequately supported by the results?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Algorithms, Machine Learning, Natural Language Processing, Deep Learning.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research