JEMMME (Journal of Energy, Mechanical, Material, and Manufacturing Engineering)
Vol. 8, No. 2, 2023

ISSN 2541-6332 | e-ISSN 2548-4281
Journal homepage: http://ejournal.umm.ac.id/index.php/JEMMME

Parametric optimization technique for continuous and combinational problems based on simulated annealing algorithm

Yosua Heru Irawan[a,*], Po Ting Lin[b]
[a]Department of Mechanical Engineering, Institut Teknologi Nasional Yogyakarta
Babarsari Street, Depok, Sleman, Yogyakarta, Indonesia
+62-274-485390

[b]Department of Mechanical Engineering, National Taiwan University of Science and Technology
No. 43, Section 4, Keelung Rd, Da'an District, Taipei City, Taiwan
+886-2-2733-3141
*Corresponding author: yhirawan@itny.ac.id

Abstract

Simulated annealing is an optimization method adapted from the annealing process. The optimization process using simulated annealing method is done by mapping the elements of physical coolant process onto the elements of optimization problem. This method uses local neighborhood search to find solutions, meaning it searches around it for answers itself and takes another solution based on everything around it. The simulated annealing method has been used successfully for the optimization process in the continuous case (Himmelblau's function) and combinational case (Quadratic Assignment Problem or QAP). Based on the optimization results (global minima) for the Himmelblau's function, the points $x_f = 3.584$ and $y_f = -1.845$ are obtained with objective function $z = 0$. The optimal solution for the eight departmental arrangements is F, E, A, G for the bottom floor and H, D, C, B for the top floor, this arrangement produces an optimal total cost of 214. The simulated annealing method accepts an uphill move (worse move) by considering the probability, in this way we will not be trapped in the local minima position. These four search space variables M, N, T_0 and α determine the performance of the simulated annealing method, we can adjust them according to the optimized case.

Keywords: *Annealing, optimization, uphill, probabilistic, metaheuristic*

1. INTRODUCTION

In the realm of optimization algorithms, the pursuit of efficiency and effectiveness stands as an enduring objective. Simulated Annealing (SA), inspired by the annealing process in metallurgy, has traversed decades of research and adaptation to become a foundational technique in the field. Originally conceived by Kirkpatrick et al. in 1983 (1), SA has recently experienced a resurgence of interest, driven by the need to address contemporary optimization challenges across a spectrum of domains. This revival is underscored by novel variants, advanced strategies, and the integration of modern computing capabilities (2)(3).

In the ever-evolving landscape of optimization, it is imperative to appreciate the substantial body of work that has contributed to the refinement and applicability of SA. Recent years have seen significant progress in understanding and extending the algorithm's capabilities. These advancements span various disciplines, showcasing SA's versatility. For instance, Aydin and Fogarty's work in 2004 has explored the intricacies of SA in combinatorial optimization, enhancing its effectiveness in solving complex problems

(2). Ahmed et.al. in 2023 has leveraged SA for fine-tuning machine learning models, introducing an intersection between optimization and artificial intelligence (3).

Moreover, the algorithm's applicability has transcended traditional boundaries. In computational chemistry, Agostini et.al. in 2006 has illuminated SA's contributions to the study of complex molecular systems, underscoring its utility in understanding real-world phenomena (4). Xu et.al. in 2011 has introduced adaptive variations of SA to address multimodal optimization problems, further expanding the algorithm's reach across diverse problem domains (5).

As computational power continues to grow, the scalability and parallelizability of SA have gained prominence. Mu et.al. in 2019 has explored parallel SA algorithms, making it more amenable to solving large-scale optimization problems in distributed computing environments (6). In the context of wireless sensor networks, Osamy et.al. in 2020 has delved into SA-based routing algorithms, demonstrating its efficacy in resource-constrained, wireless communication scenarios (7).

Furthermore, SA has found its place in the ever-expanding world of cloud computing. We et.al. in 2012 has employed SA for energy-efficient scheduling in cloud data centers, aligning optimization with green computing principles (8). Zhang et.al. in 2021 has scrutinized SA's performance in solving the Traveling Salesman Problem, showcasing its adaptability and competitiveness (9). In the context of cloud computing environments, Tanha et.al. in 2021 has employed SA heuristics for job scheduling, addressing the challenges posed by the dynamic and resource-constrained nature of cloud systems (10).

The optimization process using SA method is done by mapping the elements of physical coolant process onto the elements of optimization problem. This method uses local neighbourhood search to find solutions, meaning it searches around it for answers itself and takes another solution based on everything around it. This process is often referred to as a probabilistic metaheuristic that means it counts on probabilities and randomness.

The SA method allows some uphill moves or worse solutions to avoid getting stuck in the local minima position. The method used to accept an uphill move is to use probability, which is to compare equation (1) with a random number U (0.1). If the random number is smaller (the probability of the worse solution is high) then the uphill move is accepted, if the random number is higher, then the uphill move is not carried out. In equation (1) $f(x_t)$ is an objective function evaluated in new move and $f(x_i)$ is an objective function at current solution while T_t is temperature at the current location m. When m becomes larger (toward the end process or *T* become smaller) so the change in accepting uphill move is reduced.

$$\exp\left(-\left(\frac{f(x_t)-f(x_i)}{T_t}\right)\right) \quad (1)$$

The primary objective of this study is to demonstrate the versatility and efficacy of the Simulated Annealing (SA) algorithm in addressing a diverse range of optimization challenges. Specifically, we aim to showcase SA's ability to efficiently navigate complex, multi-modal solution spaces in continuous optimization problems like Himmelblau's Function, while also highlighting its adaptability in solving combinatorial optimization problems such as the Quadratic Assignment Problem (QAP). By achieving success in both domains, our research seeks to provide a unified framework that underscores SA's applicability across a spectrum of optimization problems, offering valuable insights for researchers and practitioners facing real-world decision-making complexities.

2. CASE STUDIES

2.1 Himmelblau's Function

Himmelblau's function is a multi-modal function, used to test the performance of optimization algorithms. This function can be written with the following equation, where x and y lie between -6 and 6:

$$f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \quad (2)$$

This function has four identical global minima at the point,

- $f(3.0, 2.0) = 0.0$,
- f(-2.8051, 3.1313) = 0.0
- f(−3.7793, -3.2891) = 0.0
- f(3.5844, -1.8451) = 0.0

We try to use the simulated annealing method to find the global minima of Himmelblau's function.

2.2 Quadratic Assignment Problem (QAP)

Quadratic Assignment Problem (QAP) is one type of problem determining the location of a number of facilities to a number of specific points. Among these facilities, there is a close relationship that can be assumed to be a flow of people or material. So that facilities that have high close relationship, these facilities need to be placed in locations adjacent anyway. Thus, the purpose of the problem to minimize the cost of displacement can be achieved.

For example, eight departments in one company are to be placed in eight locations with four in the top floor and four in the bottom floor. The objective is to minimize costs between the placed departments (the maximum cost cannot exceed 220). The cost is calculated by equation (3), and the total cost is the sum of all costs for every department. The schematic problem can be seen in figure 1. The detail of flow and distance value presented in table 1 and 2.

$$Cost = Flow * Distance \quad (3)$$

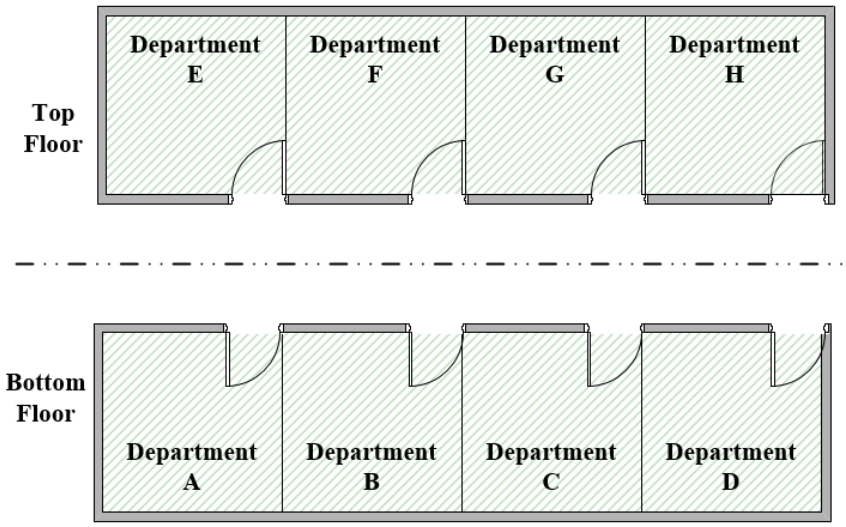

Fig. 1. Quadratic assignment problem schematic

Table 1. The distance between department

Dept.	A	B	C	D	E	F	G	H
A	0	1	2	3	1	2	3	4
B	1	0	1	2	2	1	2	3
C	2	1	0	1	3	2	1	2
D	3	2	1	0	4	3	2	1
E	1	2	3	4	0	1	2	3
F	2	1	2	3	1	0	1	2
G	3	2	1	2	2	1	0	1
H	4	3	2	1	3	2	1	0

Table 2. The flow between department

Dept.	A	B	C	D	E	F	G	H
A	0	5	2	4	1	0	0	6
B	5	0	3	0	2	2	2	0
C	2	3	0	0	0	0	0	5
D	4	0	0	0	5	2	2	10
E	1	2	0	5	0	10	0	0
F	0	2	0	2	10	0	5	1
G	0	2	0	2	0	5	0	10
H	6	0	5	10	0	1	10	0

3. METHODS

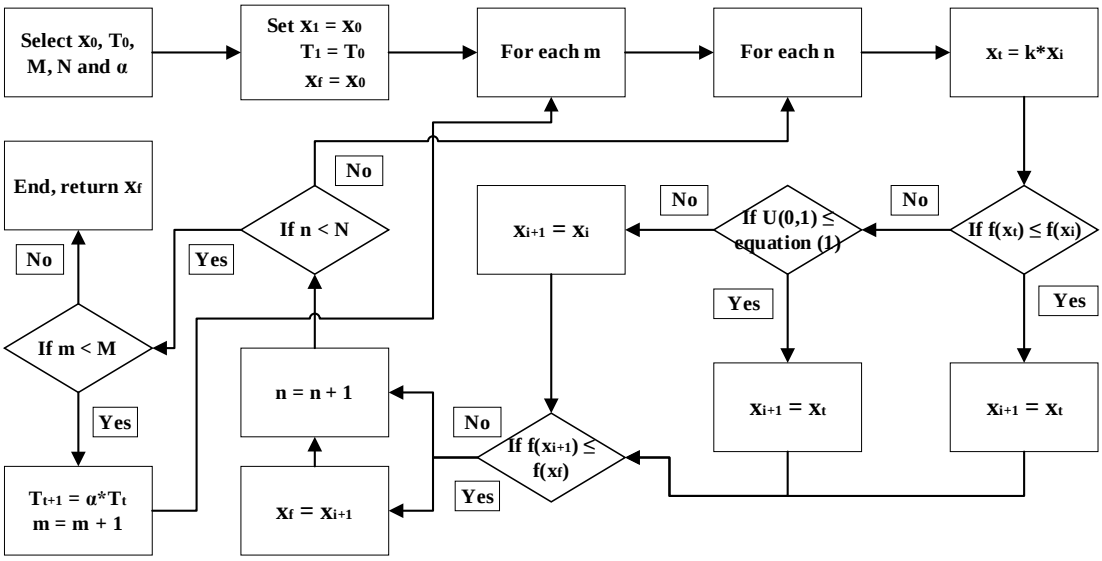

Fig. 2 Flow chart of simulated annealing

The search space notations in figure 2 are x_0 for initial solution, x_i for solution *i*, x_f for final solution and $f(x_i)$ for objective function evaluated in x_i. The notations for simulated annealing method are T_0 for initial temperature, T_t for temperature at stage *t*, T_f for final temperature, α for cooling parameter that cool down the temperature, *M* for the number of temperature, *N* for the number of move or the number of neighborhood at the temperature T_t, and *k* for the type of move operator.

The procedure for performing optimization with the simulated annealing method is as follows:

1. Set T_0, M, N, α and type of move operator *k*.
2. Start from *m = 1* (*m* means counter for *M*) and set initial value.
3. Start from random point in the search space, x_i and initial solution.
4. Move using move operator to another location, x_t.
5. Look around the neighborhood or point around your location and move to one of them, *n = 1*.
6. Check if the objective function at the current point is better,
   a. If yes, teke $x_{i+1} = x_t$.
   b. If no, take a random number and check if the random number is less than equation (1). If it less, take $x_{x+1} = x_t$ and if it greater don't take x_t and stay where we are, and look for another point (*n = n + 1*)
7. Do step 4 to 6 for *N* times (inner looping)
8. After we do it *N* times, we increase $m = m + 1$ and reduce the temperature by α, $T_{t+1} = \alpha * T_t$. We repeat this step until *M* times.
9. Get the solution, x_f.

4. RESULT AND DISCUSSION

4.1 Himmelblau's Function

To find the global minima in Himmelblau's function using the simulated annealing method, we must follow the nine steps outlined in section 3. For this case, we can solve it in the following way:

1. Set $T_0 = 1000$, $M = 300$, $N = 15$, $\alpha = 0.85$ and type of move operator *k = 0.1*. Basically, the use of move operators is optional, in this case move operators are used to produce smoother steps.
2. Start from *m = 1* (*m* means counter for *M*) and set the initial value $x_0 = 0$, $y_0 = 0$.
3. Start from initial solution $x_i = x_0$, $y_i = y_0$.
4. Move using move operator and random number to another location, x_t and y_t. As we know that the random number function gives values between 0 and 1, the move operator here is used to refine the steps generated from the random number.
5. Look around the neighborhood or point around current location and move to one of them, *n = 1*.
6. Check if the objective function at the current point is better,
   a. If yes, take $x_{i+1} = x_t$ and $y_{i+1} = y_t$.
   b. If no, take a random number and check if the random number is less than equation (1). If it less, take $x_{x+1} = x_t$, $y_{i+1} = y_t$ and if it greater don't take x_t and y_t, and stay where we are, and look for another point (*n = n + 1*)
7. Do step 4 to 6 for *15* times (inner looping)
8. After we do it *15* times, we increase $m = m + 1$ and reduce the temperature by $\alpha = 0.85$, $T_{t+1} = \alpha * T_t$. We repeat this step until *300* times (we have 300 temperature positions).
9. Get the solution, x_f and y_f.

All of the procedures that have been explained are coded using python with the numpy library for calculations and the matplotlib library for the calculation result plot. Here are the results of the calculation.

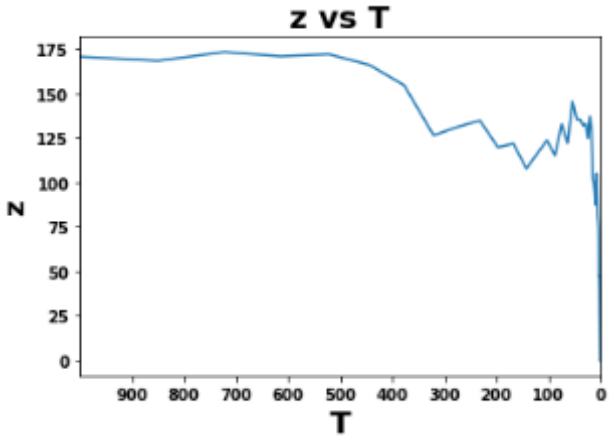

Fig. 3. Results for Hinnelblau's function case

Figure 3 shows the results of the global minima search of the Himmelblau's function using the simulated annealing method. For the initial solution with $x_0 = 0$ and $y_0 = 0$ produces the objective function z = 170, and for the objective function z = 0 is obtained from the optimal points $x_f = 3.584$ and $y_f = -1.845$. The results of this calculation can be validated with an analytical solution from Himmelblau's function that has 4 solutions or global minima positions. The above results also show that there are several local minima in the $z\ vs\ T$ graph, with the simulated annealing method that allows uphill move, we will not be trapped in the local minima position. To understand more about the algorithm of this method, we can change the optimization parameters, for example we change M = 30 and produce the following output.

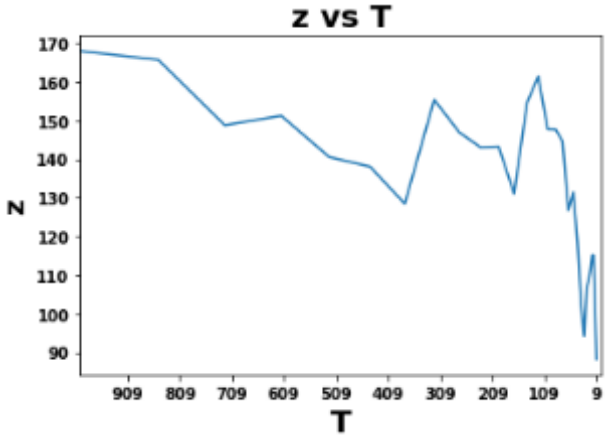

Fig. 4. Result for modified Himmelblau's function case

Figure 4 shows global Minima search results for modified Himmelblau's case with $M = 30$. The final result of this calculation is $z = 88.24$ with position $\mathrm{x_f} = 1.2$ and $y_f = 1.762$. When the number of temperatures M is changed, the final result will also change because M determines how many temperature positions will be analyzed, the less M then the calculation becomes inaccurate, like the sample results from Figure 6 above. Basically, there are still some more variables that can be changed to see the effect on the calculation results, such as N, T_0 and α. These four search space variables determine the performance of the simulated annealing method.

3.2 Quadratic Assignment Problem (QAP)

To find a solution from QAP using the simulated annealing method, we must follow the nine steps outlined in section 3. For this case, we can solve it in the following way:

1. Set $T_0 = 1500,\ M = 250,\ N = 20,\ \alpha = 0.9$. For this case we don't need an move operator, because the type of the case is a combinational case so the random number used to make the step must be an integer.
2. Start from *m = 1* (*m* means counter for *M*) and set the initial solution for department arrangement, $initial\ solution = B, D, A, E, C, F, G, H$.
3. Calculate the objective function for initial solution (use equation 3)
4. Move using integer random number. Two sets of random integer number (1 to 8) is used to construct a new arrangement. New arrangement arrangements are obtained by swapping positions obtained from integer random numbers.
5. Check if the objective function at the current arrangement is better,
   a. If yes, take the new arrangement become the solution.
   b. If no, take a random number and check if the random number is less than equation (1). If it less, the new arrangement become the solution and if it greater don't take it, and stay where we are, and look for another arrangement (*n = n + 1*)
6. Do step 4 to 6 for *20* times (inner looping)
7. After we do it *20* times, we increase $m = m + 1$ and reduce the temperature by $\alpha = 0.9$, $T_{t+1} = \alpha * T_t$. We repeat this step until *250* times (we have 250 temperature positions).
8. Get the solution for the new department arrangement.

All the procedures that have been explained are coded using python with the numpy library for calculations, the matplotlib library for plot calculation results and the pandas library for data structure. Here are the results of the calculation.

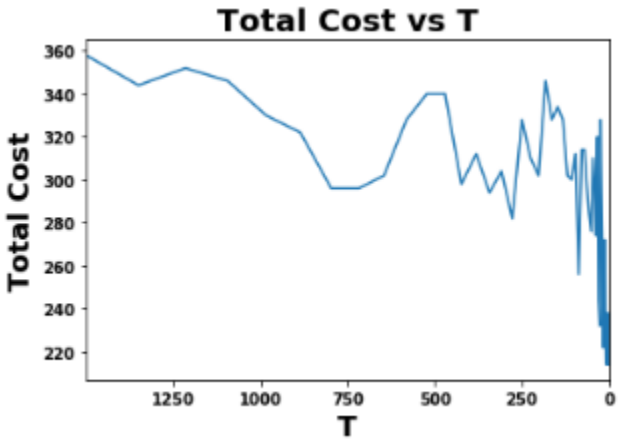

Fig. 5. Result for QAP

Figure 5 represents the results of calculations in the form of objective functions (total cost) and temperature (T). Initial arrangements *B, D, A, E, C, F, G, H* produce total cost 336 while optimal arrangements *F, E, A, G, H, D, C, B* produce total cost 214. In Figure 7 we can also see that the calculation results of the objective function (total cost) fluctuate at high temperatures (initial search). This behavior occurs because the greater T in equation (1), the probability of receiving an uphill move is also large, so that at the beginning of the search process fluctuations will occur. Because QAP is a combinational type of case, there are a number of arrangement solutions that meet the requirements, table 3 below summarizes some optimal arrangement options that meet the requirements after several trials.

Table 3. The optional solution for QAP

Bottom floor				Top floor				Total Cost
F	E	A	G	H	D	C	B	214
B	A	E	C	D	H	G	F	214
G	H	D	F	E	A	B	C	214
G	H	D	B	F	E	A	C	218
C	D	H	F	B	A	E	G	218
F	E	A	C	G	H	D	B	218
A	C	E	B	D	H	G	F	220

To better understand the procedure of how simulated annealing solves a combinational case like QAP, we can change the variables M = 10 and N = 1, this means we only do one search in one step temperature. Table 4 below shows the calculation results with the M and N variables that have been changed.

In Table 4, we can see the process of finding a solution of this case using simulated annealing. For example, from the initial arrangement to arrangement 1 there is an exchange of positions between A and E, with this exchange the cost will also change. This process continues until the calculation has reached the specified limit. On the step of arrangement 3 to arrangement 4, there is an uphill move where total cost arrangement 4 is greater than total cost arrangement 3. This behavior can occur because the probability of arrangement 4 is considered high so the calculation accepts the worse move.

Table 4. The results for modified QAP

	Bottom floor				Top floor				Total Cost
Initial arrangement	B	D	A	F	C	E	G	H	336
Arrangement 1	B	D	E	F	C	A	G	H	330
Arrangement 2	G	D	E	F	C	A	B	H	322
Arrangement 3	G	D	E	B	C	A	F	H	318
Arrangement 4	G	F	E	B	C	A	D	H	326
Arrangement 5	G	C	E	B	F	A	D	H	292

Arrangement 6	G	C	E	F	B	A	D	H	296
Arrangement 7	H	C	E	F	B	A	D	G	254
Arrangement 8	G	C	E	F	B	A	D	H	296
Arrangement 9	G	C	E	F	B	H	D	A	254

4. CONCLUSION

The simulated annealing method has been used successfully for the optimization process in the continuous case (Himmelblau's function) and combinational case (Quadratic Assignment Problem or QAP). Based on the optimization results (global minima) for the Himmelblau's function, the points $x_f = 3.584$ and $y_f = -1.845$ are obtained with objective function $z = 0$. This result is valid and can be validated with analytical solutions from Himmelblau's function which has 4 solutions or global position minima. The optimal solution for the eight departmental arrangements is *F, E, A, G* for the bottom floor and *H, D, C, B* for the top floor, this arrangement produces an optimal total cost of 214. For this QAP case there are several solutions for the arrangement as in table 3. The simulated annealing method accepts an uphill move (worse move) by considering the probability, in this way we will not be trapped in the local minima position. These four search space variables M, N, T_0 and α determine the performance of the simulated annealing method, we can adjust them according to the optimized case.

REFERENCES

1. Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220(4598), 671-680.
2. Aydin, M. E., & Fogarty, T. C. (2004). A distributed evolutionary simulated annealing algorithm for combinatorial optimisation problems. Journal of heuristics, 10, 269-292.
3. Ahmed, A., Al-Maamari, M., Firouz, M., & Delen, D. (2023). An Adaptive Simulated Annealing-Based Machine Learning Approach for Developing an E-Triage Tool for Hospital Emergency Operations. Information Systems Frontiers, 1-21.
4. Agostini, F. P., Soares-Pinto, D. D. O., Moret, M. A., Osthoff, C., & Pascutti, P. G. (2006). Generalized simulated annealing applied to protein folding studies. Journal of computational chemistry, 27(11), 1142-1155.
5. Xu, Q., Zhang, G., Zhao, C., & An, A. (2011, May). A robust adaptive hybrid genetic simulated annealing algorithm for the global optimization of multimodal functions. In 2011 Chinese Control and Decision Conference (CCDC) (pp. 7-12). IEEE.
6. Mu, D., Wang, C., Zhao, F., & Sutherland, J. W. (2016). Solving vehicle routing problem with simultaneous pickup and delivery using parallel simulated annealing algorithm. International Journal of Shipping and Transport Logistics, 8(1), 81-106.
7. Osamy, W., El-sawy, A. A., & Khedr, A. M. (2019). SATC: A simulated annealing based tree construction and scheduling algorithm for minimizing aggregation time in wireless sensor networks. Wireless Personal Communications, 108(2), 921-938.
8. Wu, Y., Tang, M., & Fraser, W. (2012, October). A simulated annealing algorithm for energy efficient virtual machine placement. In 2012 IEEE international conference on systems, man, and cybernetics (SMC) (pp. 1245-1250). IEEE.
9. Zhang, Y., Han, X., Dong, Y., Xie, J., Xie, G., & Xu, X. (2021). A novel state transition simulated annealing algorithm for the multiple traveling salesmen problem. The Journal of Supercomputing, 77, 11827-11852.
10. Tanha, M., Hosseini Shirvani, M., & Rahmani, A. M. (2021). A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynamic simulated annealing algorithms in cloud computing environments. Neural Computing and Applications, 33, 16951-16984.