Implementation and Comparison of Four Algorithms on Transportation Problem

Eghbal Hosseini[1], Line Reinhardt[1], Kayhan Zrar Ghafoor[2(✉)], and Danda B. Rawat[3]

[1] Department of People and Technology, Roskilde University, Roskilde, Denmark
hosseini@ruc.dk
[2] Department of Software Engineering, Salahaddin University-Erbil, Erbil 44002, Kurdistan Region, Iraq
mrkayhanz@gmail.com
[3] Department of Electrical Engineering and Computer Science, Howard University, Washington, DC, USA

Abstract. The transportation problem is a very applicable and relevant logistic problem. In this paper, to test meta-heuristics on the transportation problem and also improve initial feasible solutions in few number of iterations, four recent and effective meta-heuristic algorithms are used to solve transportation problems. Laying Chicken Algorithm (LCA), Volcano Eruption Algorithm (VEA), COVID-19 Optimizer Algorithm (CVA), and Multiverse Algorithm (MVA) are implemented to solve different sizes of the transportation problem. Computational results show that CVA is the most efficient optimizer for large size cases and LCA is the best algorithm for the others. Finally, convergence of algorithms will be discussed and rate of convergence will be compared. The advantage of these heuristics are that they can be easily adapted to more challenging versions of the transportation problem which are not solveable by the Simplex method.

Keywords: Transportation problems · Meta-heuristic algorithm

1 Introduction

One of the significant linear programming problems is the transportation problem which is used for inventory, assignment and traffic [1,8,10,11]. In the transportation problem a product is transported from a set of sources to a set of destinations minimizing the transportation cost while satisfying the demand, the mathematical formulation has been shown in (1)–(4).

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

$$Subject\ to \sum_{j=1}^{n} x_{ij} \leq S_i, i = 1, 2..., m \tag{2}$$

S. Paiva et al. (Eds.): SmartCity360° 2021, LNICST 442, pp. 422–433, 2022.
https://doi.org/10.1007/978-3-031-06371-8_28

$$\sum_{i=1}^{m} x_{ij} = D_j, j = 1, 2..., n \tag{3}$$

$$x_{ij} \geq 0 \tag{4}$$

Proposing optimal solution needs to start from a feasible solution as Initial Feasible Solution (IFS). Although several meta-heuristic approaches have been proposed to solve optimization problems, but there just few methods in references which can find IFS for the problem [8]. The existing methods to achieve the minimal total cost do not propose a suitable feasible solution to reduce the number of iterations [2,3]. Thus, it is still a challenge to present a better method of IFS. Our heuristic algorithms have been designed to fit the aforementioned challenges of the transportation problem.

In this paper, Laying Chicken Algorithm (LCA), Volcano Eruption Algorithm (VEA), COVID-19 Optimizer Algorithm (CVA), and Multiverse Algorithm (MVA) [4–7]are used to find IFS of different transportation problems. Feasibility, efficiency and convergence of these algorithms are discussed and finally algorithms with best results are compared with Vogel and ZSM [8,9].

2 LCA, VEA, MVA and CVA Algorithms

2.1 Laying Chicken Algorithm (LCA)

Laying Chicken Algorithm focuses on behavior of laying hens and finding an answer to how does the hen convert the egg to the chicken? LCA converts the feasible solutions to the optimal solution, same as eggs to the chicken. In fact, each feasible solution of a continuous programming problem displays an egg and the optimal solution of the problem is a chicken. Here hens try to warm their eggs; this concept is used to change and improve the solutions in LCA. As the temperature is increased the solutions to the problem are improved. Rotation of eggs is the next concept which will be simulated by a little mutation in the solutions. There are the following steps to formulate of the behavior the hen in the LCA optimizer [4]:

1. The first egg which displays initial solution.
2. More eggs displays initial population close to the initial solution.
3. Improve solutions of population inspiring from warming eggs.
4. Little mutation of solutions inspiring of rotation of eggs.

2.2 Volcano Eruption Algorithm (VEA)

The Volcano Eruption Algorithm, inspired by the nature of a volcano eruption. VEA optimizer imitates the process of volcano eruption, which is a hole on the earth's surface. This phenomenon acts as a vent for release of pressurized gases, molten rock or magma deep beneath the surface of earth. Magma

Algorithm 1. LCA Procedure for Transportation Problem

```
n: Number of solutions
N: Number of Iterations
α: A given positive number (less than size of the problem)
Generate a random initial feasible solution X0
Generate initial population near initial solution
for i ← 1 to N do
    for k ← 1 to n do
        if Xk is not better than X0 then
Xk = X0 + α * (Xk/||Xk||)
        end if
Xk = Xk + Xk/||Xk||
    end for
end for
```

is passed through a channel from deep underground called the volcanic pipe. Magma erupts out of the earth's surface when it reaches the hole on the surface. There are the following steps leading to formation of a volcano to VEA optimizer [5]:

1. Rise of magma through the volcanic pipe.
2. Volcanic eruption by rising of magma to the surface of the earth.
3. Lava's cooling down and therefore formation of a crust.
4. Repetition of this process over time leading to several layers of rock that builds up over time resulting in a volcano.

Algorithms 1–4 show the pseudocodes of LCA, VEA, CVA, and MVA algorithms respectively.

2.3 COVID-19 Optimizer Algorithm (CVA)

The Covid-19 Optimizer Algorithm, inspired by the coronavirus, inspired by the coronavirus in nature which has started spreading rapidly due to its high transmission behavior. CVA has two significant parts: outbreaks and export. These outbreaks and export processes have given inspiration to the proposed CVA algorithm to generate its initial solutions and population mimicking COVID-19 spreading behavior. Some of the solutions are removed from the population because of the fact that those cases have recovered or already passed away. So the algorithm will carry on exploring the remaining best solutions. The following steps formulate COVID-19 to create the CVA optimizer [6]:

1. Initial place to start the virus which displays initial solution for CVA.
2. Outbreaks in the initial area displays initial population close to the initial solution.
3. Export processes displays solutions far from initial area.
4. Selection of the best solutions inspiring from the recovery process.

Algorithm 2. VEA Procedure for Transportation Problem

n: Number of solutions
s: Size of the problem
Rand: Random integer number between 1 and s
λ: A given integer positive number
Generate initial population
for $t \leftarrow 1$ to λ **do**
  for $k \leftarrow 1$ to n **do**
$Xk = Xk + Rand * \frac{Xk}{||Xk||}$ (Distribution of solutions in feasible region)
  end for
Find best solution
Let $Xt = xbest$
end for
for $t \leftarrow 1$ to λ **do**
  for $i \leftarrow 1$ to n **do**
$Xi = Xt + Rand * \frac{Xk}{||Xk||}$ (Distribution of best solutions)
  end for
Find best solution
Let $Yt = xbest$
end for

Algorithm 3. CVA Procedure for Transportation Problem

n: Number of solutions
N: Number of Iterations
s: Size of the problem
Rand: Random integer number between 1 and s
β: percentage of export
Generate a random initial solution X0
Generate initial population inccluding $(1 - \beta n)$ solutions near initial solution
Export $\beta n)$ solutions to other regions
for $k \leftarrow 1$ to N **do**
  for $k \leftarrow 1$ to $(1 - \beta)n$ **do**
$Xk = X0 + (\frac{Xk}{||Xk||})$
  end for
  for $k \leftarrow (1 - \beta)n + 1$ to n **do**
$Xk = X0 + Rand * (\frac{Xk}{||Xk||})$
  end for
end for

2.4 Multiverse Algorithm (MVA)

Multiverse Algorithm has been inspired from multiverse theory which states that there are several universes, not one, in the world. More particularly, multiverse theory states there are more than one big bangs besides to the big bang of our universe. So MVA algorithm comes from the existence of several worlds and big bangs. Therefore, MVA algorithm starts with a several solutions as initial population. Each universe is built from a very small and dense particle based on multiverse theory. This is the main idea to create the next population very close to the solutions of initial population in the simulated MVA. Finally, all solutions are distributed in the feasible region same as big bangs. Following steps have been used to formulate multiverse theory to MVA optimizer [7]:

1. Several universes which displays initial population including several solutions.
2. Very dense particles display many solutions very close to initial solutions.
3. Distribution of solutions in the feasible region inspiring from big bangs.
4. Rotation of solutions inspiring from the rotation of particles in multiverse theory.

Algorithm 4. MVA Procedure for Transportation Problem

n: Number of solutions
N: Number of Iterations
m: A given integer positive number
s: Size of the problem
Rand: Random integer number between 1 and s
Generate initial population including n solutions
for $t \leftarrow 1$ to N **do**
  for $k \leftarrow 1$ to n **do**
    for $i \leftarrow 1$ to m **do**
$Xi = Xk + Rand * \frac{Xk}{||Xk||}$ (Explosion of initial population)
    end for
Find the best solution and let $Xk = xbest$
  end for
  for $j \leftarrow 1$ to m **do**
$Xj = Xk + Rand * \frac{Xk}{||Xk||}$ (Explosion of the best solutions)
Find the best solution and let $XtF = xbest$
  end for
end for

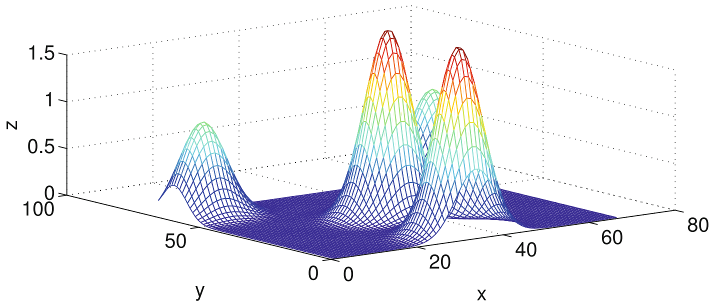
The behavior of the algorithms when finding of the optimal solution during the five iterations for an optimization problem with two global optima is shown in Fig. 2.

Figure 2-a shows initial population of LCA algorithm, and Fig. 2-b shows that optimal solutions, large red points, have been surrounded by best solution of LCA after five iterations. Figure 2-c shows initial population of VEA algorithm, and Fig. 2-d shows that one of optimal solutions has been completely surrounded by best solution of VEA. Figure 2-e and Fig. 2-f show the process of MVA to solve the problem. Finally, Fig. 2-g and Fig. 2-h show the behavior of CVA for solving the problem which the optimal solution is the blue point.

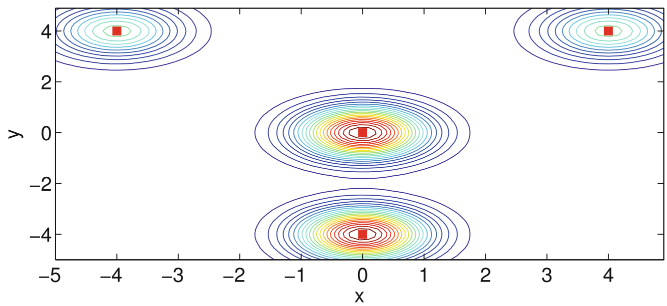
Problem 1:
Consider the following non-linear problem:

$$\max e^{-(x-4)^2-(y-4)^2} + e^{-(x+4)^2-(y-4)^2} + 2(e^{-x^2-y^2} + e^{-x^2-(y+4)^2}) \tag{5}$$

Figure 1 shows optimal solutions, contours and diagram of problem 1.



(a) Diagram



(b) Contours

Fig. 1. Contours and diagram of Problem 1

3 Computational Results

To show the numerical efficiency of the algorithms, several transportation problems are solved. CVA is the best for very large size problems and LCA is the best algorithm for others. Convergence rate of algorithms for a given problem has been shown in Fig. 3.

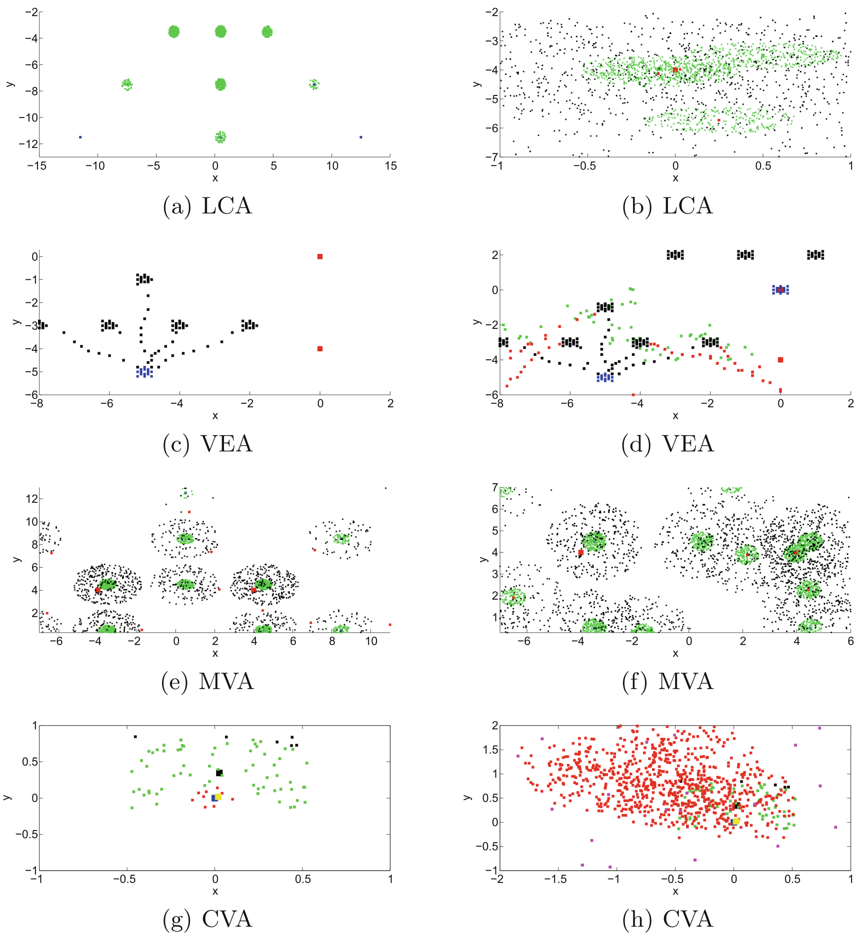


Fig. 2. Behavior of LCA, VEA, MVA and CVA Algorithms for solving Problem 1

Table 1. Results by the first iteration for different sizes of the problem up to four decimal places.

Size of problem	LCA	VEA	MVA	CVA
20 * 10	126.9417	**123.2145**	154.2567	138.6532
80 * 50	889.8801	1004.2255	**821.5467**	1010.3287
200 * 100	**1.3808e+03**	1.4511e+03	1.4321e+03	1.4124e+03
300 * 200	**2.8508e+03**	3.1256e+03	3.0345e+03	2.8621e+03
800 * 500	**7.7524e+03**	7.8865e+03	7.8724e+03	7.7843e+03
2000 * 1000	1.4991e+04	1.6324e+04	1.7843e+04	**1.4985e+04**
5000 * 2000	3.0816e+04	3.2567e+04	3.3587e+04	**3.0003e+04**
7000 * 3000	4.6838e+04	4.8456e+04	4.8632e+04	**4.4802e+04**
10000 * 5000	7.5023e+04	7.7834e+04	7.8246e+04	**7.3543e+04**

Table 2. Results with five iterations for different sizes of the transportation problem

Size of problem	LCA	VEA	MVA	CVA
20 * 10	108.5360	**101.6532**	142.1287	112.4765
80 * 50	826.2148	885.1265	**815.8721**	1000.1263
200 * 100	**1.3635e+03**	1.4167e+03	1.4214e+03	1.4014e+03
300 * 200	**2.8487e+03**	3.003e+03	3.002e+03	2.8523e+03
800 * 500	**7.7487e+03**	7.8860e+03	7.8521e+03	7.7753e+03
2000 * 1000	1.4976e+04	1.6301e+04	1.7821e+04	**1.4821e+04**
5000 * 2000	3.0814e+04	3.2521e+04	3.3501e+04	**2.9831e+04**
7000 * 3000	4.6814e+04	4.8412e+04	4.8600e+04	**4.4786e+04**
10000 * 5000	7.5017e+04	7.7821e+04	7.8213e+04	**7.3521e+04**

Tables 1, 2 and 3 show the results of all four algorithms to find IFS for nine transportation problems with different sizes. Table 1 shows the results at the first iteration, it is clear that CVA is very efficient for very large size problems and LCA is the best for other sizes. Tables 2 and 3 show the results after 5 and 20 iterations respectively. Table 4 shows comparison of LCA, CVA with Min-Cost and Vogel algorithms. Based on the results of this table, proposed algorithm is much better than Vogel algorithm which is the best method to find IFS. Finally, Table 5 shows the improvement of solution rather than Vogel by LCA and CVA.

Figure 2 shows the rate of convergence for LCA, CVA, VEA, and MVA algorithms to find IBFS of a given transportation problem which it's optimal solution is 87.

Table 3. Best results after 20 iterations up to four decimal places.

Size of problem	LCA	VEA	MVA	CVA
20 * 10	101.6923	**87.1245**	132.0327	110.173
80 * 50	820.2360	864.8234	**785.1241**	974.0217
200 * 100	**1.3601e+03**	1.4032e+03	1.4014e+03	1.3846e+03
300 * 200	**2.8300e+03**	2.991e+03	2.972e+03	2.8421e+03
800 * 500	**7.7457e+03**	7.8743e+03	7.8500e+03	7.7658e+03
2000 * 1000	1.4903e+04	1.6281e+04	1.7543e+04	**1.4748e+04**
5000 * 2000	3.0811e+04	3.2503e+04	3.2846e+04	**2.9785e+04**
7000 * 3000	4.6753e+04	4.8401e+04	4.85450e+04	**4.4700e+04**
10000 * 5000	7.5004e+04	7.7811e+04	7.8005e+04	**7.3456e+04**

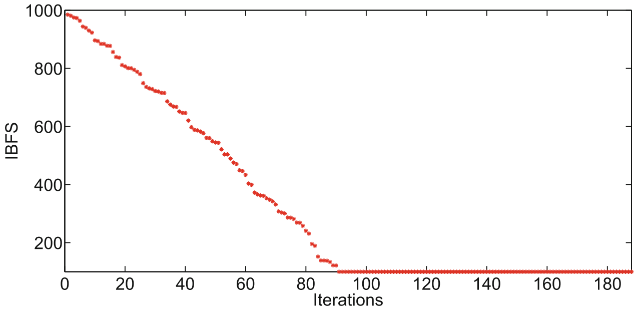
Table 4. Comparison of LCA and CVA with previous approaches.

Size of problem	Min-Cost	Vogel	LCA	CVA
200 * 50	843.1786	637.5617	**324.9941**	543.7385
400 * 80	1.5780e+03	902.3972	**502.6537**	743.1247
1000 * 800	1.4670e+05	1.4583e+04	**7.0897e+03**	7.1247e+03
3000 * 2000	3.8145e+05	4.6366e+04	2.2907e+04	**2.2901e+04**
8000 * 5000	4.1862e+06	1.2149e+05	7.5494e+04	**7.5356e+04**
12000 * 10000	1.8853e+06	1.8551e+05	1.2971e+05	**1.2886e+05**

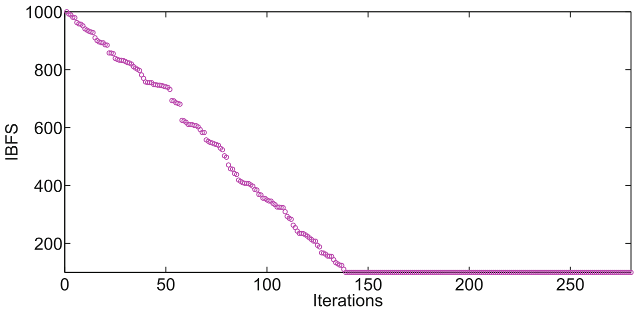
Table 5. Improvement rate of LCA and CVA Rather Vogel.

Size of problem	Laying Chicken Algorithm (LCA)	COVID-19 Optimizer Algorithm (CVA)
200 * 50	**49%**	15%
400 * 80	**44%**	18%
1000 * 800	**51%**	50%
3000 * 2000	**50%**	**50%**
8000 * 5000	37%	**38%**
12000 * 10000	30%	**31%**

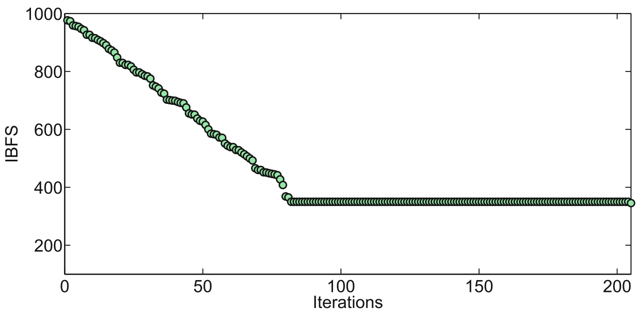
All four algorithms have been run for 300 times with 50 agents. Based on the results of this figure, LCA achieves 100 after 90 iterations and CVA obtains 100 after 140 iterations. Best solution by VEA is 365 after 80 iterations and MVA gets 400 after 110 iterations. Tables 6 shows the detail of test problems by [2] and Tables 7 shows the results and comparison of LCA and CVA with ZSM. results after 5 and 20 iterations respectively



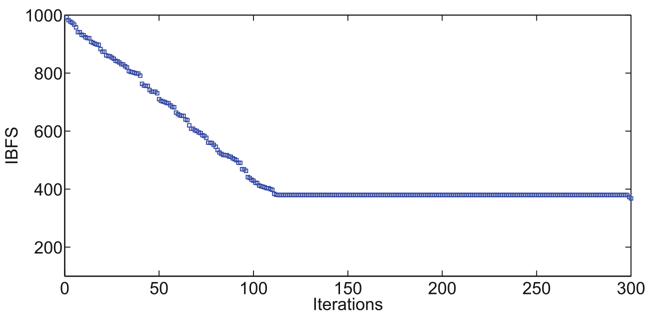
(a) LCA



(b) CVA



(c) VEA



(d) MVA

Fig. 3. Convergent Rate of LCA, CVA, VEA and MVA Algorithms with 300 Iterations for Finding IFS

Table 6. Data of proposed problems by [2]

Problems	Data
Problem 2	$C = [6, 8, 10; 7, 11, 11; 4, 5, 12], S = [150, 175, 275], D = [200, 100, 300]$
Problem 3	$C = [20, 22, 17, 4; 24, 37, 9, 7; 32, 37, 20, 15], S = [120, 70, 50], D = [60, 40, 30, 110]$
Problem 4	$C = [4, 6, 8, 8; 6, 8, 6, 7; 5, 7, 6, 8], S = [40, 60, 50], D = [20, 30, 50, 50]$
Problem 5	$C = [19, 30, 50, 12; 70, 30, 40, 60; 40, 10, 60, 20], S = [7, 10, 18], D = [5, 7, 8, 15]$
Problem 6	$C = [13, 18, 30, 8; 55, 20, 25, 40; 30, 6, 50, 10], S = [8, 10, 11], D = [4, 6, 7, 12]$
Problem 7	$C = [25, 14, 34, 46, 45; 10, 47, 14, 20, 41; 22, 42, 38, 21, 46; 36, 20, 41, 38, 44]$,
	$S = [27, 35, 37, 45], D = [22, 27, 28, 33, 34]$
Problem 8	$C = [9, 12, 9, 6, 9, 10; 7, 3, 7, 7, 5, 5; 6, 5, 9, 11, 3, 11; 6, 8, 11, 2, 2, 10], S = [2, 5, 6, 9]$
	$D = [2, 2, 4, 4, 4, 6]$

Table 7. Comparison of LCA and CVA with ZSM [2]

Prob. no	ZSM	LCA	CVA	Improvement by LCA	Improvement by CVA
2	4525	4525	**4525**	0%	**0%**
3	3460	**2960**	3220	**14%**	7%
4	920	1030	1010	−11%	−9%
5	864	**664**	764	**23%**	11%
6	476	**380**	402	**20%**	15%
7	3598	4221	3820	−17%	−6%
8	136	210	**130**	−54%	**4%**

4 Conclusion

This paper used four recent efficient meta-heuristic algorithms for finding Initial Feasible Solution (IFS) of transportation problems. The computational results show that not only the algorithms are feasible but also they are very efficient. So, using these algorithms for other practical problems of optimization such as: routing problems, emergency logistics, green supply chain, and stowage planning could be a suitable idea for future works.

References

1. Aizemberg, L., Kramer, H.H., Pessoa, A.A., Uchoa, E.: Formulations for a problem of petroleum transportation. Eur. J. Oper. Res. **237**(1), 82–90 (2014). https://doi.org/10.1016/j.ejor.2014.01.036
2. Amaliah, B., Fatichah, C., Suryani, E.: A new heuristic method of finding the initial basic feasible solution to solve the transportation problem. J. King Saud Univ. Comput. Inf. Sci. **22** (2020)
3. Cosma, O., Pop, P.C., Dănciulescu, D.: A novel matheuristic approach for a two-stage transportation problem with fixed costs associated to the routes. Comput. Oper. Res. **118**, 104906 (2020). https://doi.org/10.1016/j.cor.2020.104906

4. Hosseini, E.: Laying chicken algorithm: a new meta-heuristic approach to solve continuous programming problems. J. Appl. Comput. Math. **6**(1), 1–8 (2017)
5. Hosseini, E., Sadiq, A.S., Ghafoor, K.Z., Rawat, D.B., Saif, M., Yang, X.: Volcano eruption algorithm for solving optimization problems. Neural Comput. Appl. **33**(7), 2321–37 (2021)
6. Hosseini, E., Ghafoor, K.Z., Sadiq, A.S., Guizani, M., Emrouznejad, A.: Covid-19 optimizer algorithm, modeling and controlling of coronavirus distribution process. IEEE J. Biomed. Health Inf. **24**(10), 2765–2775 (2020)
7. Hosseini, E., Ghafoor, K.Z., Emrouznejad, A., Sadiq, A.S., Rawat, D.B.: Novel metaheuristic based on multiverse theory for optimization problems in emerging systems. Appl. Intell. **11**, 1–8 (2020)
8. Juman, Z.A., Hoque, M.A.: An efficient heuristic to obtain a better initial feasible solution to the transportation problem. Appl. Soft Comput. **1**(34), 813–26 (2015)
9. Juman, Z.A.M.S., Hoque, M.A., Buhari, M.I.: A sensitivity analysis and an implementation of the Wellknown Vogel's approximation method for solving unbalanced transportation problem. Malays. J. Sci. **32**(1), 66–72 (2013)
10. Ramadan, S.Z., Ramadan, I.Z.: Hybrid two-stage algorithm for solving transportation problem. Mod. Appl. Sci. **6**(4), 12 (2012)
11. Soriano, A., Vidal, T., Gansterer, M., Doerner, K.: The vehicle routing problem with arrival time diversification on a multigraph. Eur. J. Oper. Res. **286**(2), 564–575 (2020)