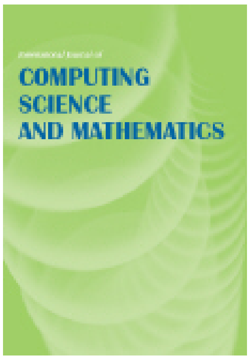


International Journal of Computing Science and Mathematics

ISSN online: 1752-5063 - ISSN print: 1752-5055
https://www.inderscience.com/ijcsm

Solving capacitated vehicle routing problem with route optimisation based on equilibrium optimiser algorithm

Ibrahim Fares, Aboul Ella Hassanien, Rizk M. Rizk-Allah, Roushdy Mohamed Farouk, Hassan Mostafa Abo-donia

DOI: 10.1504/IJCSM.2023.10055619

Article History:

Received:	03 August 2020
Last revised:	06 October 2020
Accepted:	06 October 2020
Published online:	20 April 2023

Solving capacitated vehicle routing problem with route optimisation based on equilibrium optimiser algorithm

Ibrahim Fares*

Faculty of Science,
Department of Mathematics,
Zagazig University,
Zagazig, 7120730, Egypt

and

Scientific Research Group in Egypt (SRGE),
Egypt
Email: ifares.cs@gmail.com
Website: http://www.egyptscience.net
*Corresponding author

Aboul Ella Hassanien

Faculty of Computer and information,
Cairo University,
Cairo, 3753450, Egypt

and

Scientific Research Group in Egypt (SRGE),
Egypt
Email: aboitcairo@gmail.com
Website: http://www.egyptscience.net

Rizk M. Rizk-Allah

Faculty of Engineering,
Menoufia University,
Shebeen El-Kom, 6121890, Egypt

and

Scientific Research Group in Egypt (SRGE),
Egypt
Email: rizk_masoud@yahoo.com
Website: http://www.egyptscience.net

Roushdy Mohamed Farouk
and Hassan Mostafa Abo-donia

Faculty of Science,
Department of Mathematics,
Zagazig University,
Zagazig, 7120730, Egypt
Email: rmfarouk1@yahoo.com
Email: donia_1000@yahoo.com

Abstract: In this paper, we have presented an optimising approach based on equilibrium optimiser (EO) algorithm for solving the capacitated vehicle routing problem (CVRP). The CVRP is considered one of the NP-hard combinatorial optimisation problems and most of algorithms failed to reach optimality in these problems. The EO algorithm is a powerful technique in solving several combinatorial optimisation problems. The performance of the EO algorithm compared with the artificial bee colony algorithm, the particle swarm optimisation algorithm, and the whale optimisation algorithm. The computational results obtained for the CVRP model illustrate the power of the EO algorithm over the competitor algorithms.

Keywords: metaheuristic; combinatorial optimisation; natured inspired algorithms; particle swarm optimisation; artificial bee colony.

Reference to this paper should be made as follows: Fares, I., Hassanien, A.E., Rizk-Allah, R.M., Farouk, R.M. and Abo-donia, H.M. (2023) 'Solving capacitated vehicle routing problem with route optimisation based on equilibrium optimiser algorithm', *Int. J. Computing Science and Mathematics*, Vol. 17, No. 1, pp.13–27.

Biographical notes: Ibrahim Fares (Ibrahim Ahmed) is a Teaching Assistant at Zagazig University. He received his MSc in Artificial intelligence from the Mathematics Department, Faculty of Science, Zagazig University. His research interests include, metaheuristics, IoT, deep learning, and cybersecurity.

Aboul Ella Hassanien (Abo) is a Professor at Cairo University. He received his PhD from the Department of Computer Science, Graduate School of Science & Engineering, Tokyo Institute of Technology, Japan. His research interests include computational intelligence, information security, and multimedia processing.

Rizk M. Rizk-Allah is a Lecturer at Faculty of Engineering, Menoufia University. His research interests include metaheuristics, swarm intelligence, Internet of Things, bio-inspired algorithms.

Roushdy Mohamed Farouk is a Professor at Faculty of Science, Zagazig University. His research interests include image processing and aritificial intelligence.

Hassan Mostafa Abo-donia is a Professor at Faculty of Science, Zagazig University. His research interests include pure mathematics.

1 Introduction

In the coming decade, especially in such conditions as the spread of the Covid-19 virus, the experts expect that the logistics business will encounter a pivotal evolution worldwide (Huang et al., 2019). In China, the logistics business has shown speedy growth in the last years and has listed on top of 45 world markets (Deloitte Research, 2017). In the US, the logistics business increased to 12.7% resulting in more than 400,000 available job positions (Damicis, 2018), also, in India and Europe (Mukherjee, 2017; Savills Investment Management, 2016). Satisfying growing customers' orders is the prominence in the civilised logistics dynamically and effectively in various areas like trading companies, retail store. Hence, the companies should find new technology methods to serve customers quickly and flexibly way and vie in the market to satisfy customer demands.

To fast customers serving in smart cities, researchers try to solve the vehicle routing problem (VRP) by many different methods to minimise the cost and time to deliver all customer demand. In VRP, selecting the order of customers to visit by using a set of vehicles such that each vehicle should start the journey from the depot, then visit a set of customers (less than the total customers), and it must return to the same depot. The VRP is *NP-hard* combinatorial optimisation problems and very difficult to solve a large instance. So, many heuristic methods applied to solve the VRP to get better solutions (Heuristics et al., 2002; Vidal et al., 2013). Recently, metaheuristic methods also used to address the VRP and try to reach optimality. A great deal of research has accomplished on the VRP. The state of the art metaheuristic algorithms used to solve the VRP includes, for example, the Tabu Search (Gendreau et al., 2008; Brandão, 2004; Montané et al., 2006), the simulated annealing (Afifi et al., 2013; Normasari et al., 2019), the artificial bee colony algorithm (Yao et al., 2017; Chen and Zhou, 2018), the particle swarm optimisation algorithm (Marinakis et al., 2018, 2010; Gong et al., 2012). An efficient review of using metaheuristic algorithms for solving the VRP and its variants presented in Elshaer and Awad (2020).

Metaheuristic algorithms have developed to solve hard or difficult *NP* real-world optimisation problems as stochastic techniques. These algorithms used to solve different problems in varied fields, for example, in engineering (Hadavandi et al., 2018), financial (Hafezi et al., 2015), bioinformatics (Das et al., 2008), and medicinal field (Lin et al., 2012). Also, these algorithms do not need to have full accommodate for problems. The most popular category of metaheuristic algorithms is natured-inspired algorithms. The nature-inspired algorithms designed to imitate natural phenomena such as materialistic and living organisms' phenomena or animal behaviour. In 1968, Dommel and Tinney introduced the first solution method for the optimal power flow problem (Dommel and Tinney, 1968). After this, several algorithms of other nature-inspired organised. For examples of these algorithms, the *PSO* algorithm designed to inspire the school behaviour of the fish or behaviour of the flowing birds (Kennedy and Eberhart, 1995). Artificial bee colony algorithm introduced by inspiring the rummaging and jumping behaviour of honey bees (Karaboga, 2005). The Whale optimisation algorithm which mimics the behaviour of humpback whales (Mirjalili and Lewis, 2016). And others (Rizk-Allah et al., 2018; Rizk-Allah and Hassanien, 2018; Rizk-Allah et al., 2018; Fares et al., 2020).

This paper proposed solving the CVRP by the new bio-inspired equilibrium optimiser (EO) algorithm which introduced recently in 2019 (Faramarzi et al., 2020). The computational results presented in the subsequent sections prove that the EO algorithm is more efficient than the ABC, PSO, and WOA algorithms.

The remainder of this paper is structured as follows. Section 2 presents an overview of the mathematical definition of CVRP and the EO algorithm. The computational results presented in Section 3. Finally, the conclusion and the future work in Section 4.

2 Preliminary

2.1 Capacitated vehicle routing problem (CVRP) formulation

The CVRP is a variant of *VRPs*. The main goal of CVRP is to obtain the shortest travel distance of vehicles (m) that visit or serve a group of customers (n). The CVRP controlled by a set of constraints, such that:

- each vehicle starts from the depot, serve a group of customers, and must return to that depot
- each customer must be served by one vehicle
- all demands of all customers assigned to one vehicle cannot exceed the capacity of that vehicle.

In mathematics as (Yao et al., 2017), the objective function of the CVRP described as follows:

$$Min \qquad \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{m} C_{ij} X_{ij}^{k}, \tag{1}$$

$$S.t. \ \sum_{k=1}^{m}\sum_{i=0}^{n} X_{ij}^{k} = 1, \qquad j = 1,2,\ldots,n, \tag{2}$$

$$\sum_{k=1}^{m}\sum_{j=0}^{n} X_{ij}^{k} = 1, \quad i = 1,2,\ldots,n, \tag{3}$$

$$\sum_{i=0}^{n} X_{ib}^{k} - \sum_{j=0}^{n} X_{bj}^{k} = 0 \ , \qquad k = 1,2,\ldots,m; \qquad b = 1,2,\ldots n \tag{4}$$

$$\sum_{i=0}^{n}\sum_{j=0}^{n} X_{ij}^{k} di \le Q, \qquad k = 1,2,\ldots,m, \tag{5}$$

$$\sum_{i=0}^{n}\sum_{j=0}^{n} X_{ij}^{k} (C_{ij} + S_i) \le T, \qquad k = 1,2,\ldots,m, \tag{6}$$

$$\sum_{j=1}^{n} X_{ij}^{k} = \sum_{j=1}^{n} X_{ji}^{k} 1, \qquad i = 0; \quad k = 1,2,\ldots,m, \tag{7}$$

$$\sum_{j,j\in R}^{n} X_{ij}^{k} \le |R|-1 \ , \ \ R \subseteq \{1,\ldots,n\}, 2 \le |R| \le n-1; k=1,2,..,m \tag{8}$$

$$X_{ij}^{k} \in \{0,1\}, \quad i,j=0,1,2,\ldots,n; \quad k=1,2,..,m. \tag{9}$$

where 0 is the depot position index, $X_{ij}^{k}=1$ if the edge from customer i to customer j visited by vehicle k; else, $X_{ij}^{k}=0$ such that $i \neq j$, and all other notations listed in Nomenclature and Abbreviations.

The primary objective function presented in equation (1). While equations (2) and (3) guarantee that each customer can be visited by one vehicle. Equation (1) keeps a connection in each node for each vehicle. Equation (5) guarantees that the total demands of all customers assigned to one vehicle cannot overtake its full capacity. Likewise, equation (6) assurances that the total travelled distance of a vehicle's path cannot pass the length limit of the path. Equation (7) guarantees that each vehicle could be hired only once and each vehicle requires to begin its tour from the depot and return to that depot. Equation (8) apply constraints to delete any uncomplete path. Moreover, the border entireness constraint presented in equation (9).

2.2 The equilibrium optimiser (EO)

Recently in (2019), Seyedali Mirjalili et al. developed the EO algorithm (Faramarzi et al., 2020). The EO algorithm inspired by the control volume mass balance models used to estimate both dynamic and equilibrium states. For the inspiration details see (Faramarzi et al., 2020). The mathematical description of the EO algorithm as the following steps:

Step 1: The EO algorithm initialisation a collection of particles, where each particle expresses the concentration that includes the solution to the problem. The primary concentrations are randomly formed in the search space by the next equation:

$$\vec{V}_i = C_{min} + r*(C_{max} - C_{min}) \qquad i=0,1,2,\ldots,n \tag{10}$$

such that $\vec{V}_i$ is the concentration for the particle i, C_{min} is the upper bound, and C_{max} is the lower bound of the dimension of the problem. r is a random number in $[0,1]$. n is the max number of particles.

Step 2: The EO algorithm considers the equilibrium state as the global optima of the optimisation problem or near of it. Therefore, the main goal of the EO algorithm is to search for the equilibrium state in the search space. When the EO algorithm starts, the level of the concentration is anonymous for it such reach the equilibrium state. Therefore, it allocates five particles: first four of particles are the best-so-far found in the population at equilibrium candidates and their average. The first four particles assist in the diversification process of the EO algorithm and the average assists in the intensification process.

$$\vec{P}_{eq,\ pol} = \{\vec{P}_{eq(1)}, \vec{P}_{eq(2)}, \vec{P}_{eq(3)}, \vec{P}_{eq(4)}, \vec{P}_{eq(ave)}\} \tag{11}$$

For more details for the equilibrium candidates can found in (Faramarzi et al., 2020).

Step 3: the next term serves EO to a believable equilibrium between intensification and diversification. Because rotation average can change through time in a real control volume, $\vec{\lambda}$ is supposed to be a random vector between 0 and 1.

$$\vec{F} = e^{-\vec{\lambda}(t-t_0)} \tag{12}$$

where t is reduced with increasing the iteration (it) by the next equation:

$$t = \left(1 - \frac{it}{t_{max}}\right)^{\frac{it*a2}{t_{max}}} \tag{13}$$

such that it is the current iteration, t_{max} is the max number of iterations, and $a2$ is a constant used to manage intensification capability. Latter constant, $a1$, is used to develop the diversification and intensification of EO and is designed as the next formula:

$$t_0 = t + \frac{1}{\vec{\lambda}} \ln\left(-a1 * sign(\vec{r} - 0.5)[1 - e^{-\vec{\lambda}t}]\right) \tag{14}$$

Generation rate (R) is a latter factor used to increase the intensification capability and is formed in the next equation:

$$\vec{R} = \overrightarrow{R_0} * e^{-\vec{k}(t-t_0)} \tag{15}$$

where $\vec{k}$ is random vector in $[0,1]$, and $\vec{R}_0$ is the initial generation rate value and is formed as the following equation:

$$\vec{R}_0 = \overrightarrow{GCP} * (\vec{C}_{eq} - \vec{\lambda} * \vec{C}) \tag{16}$$

And

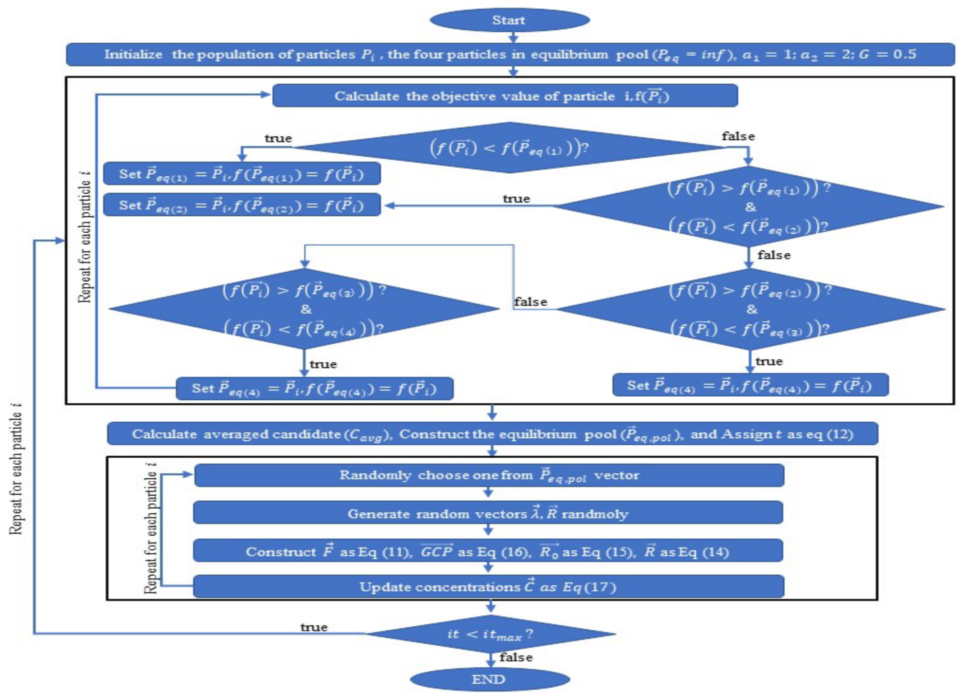
$$\overrightarrow{GCP} = \begin{cases} 0.5 r_1 & r_2 > RP, \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where r_1, r_2 are random numbers in range $[0,1]$. $\overrightarrow{GCP}$ is a control parameter of the generation rate that decides whether the R will be used to the updating process according to the value of the RP probability parameter. The EO updating equation as the next equation:

$$\vec{C} = \vec{C}_{eq} + \left(\vec{C} - \vec{C}_{eq}\right) * \vec{F} + \frac{\vec{R}}{\vec{\lambda} * V} * \left(1 - \vec{F}\right), \tag{18}$$

where V is set to be 1. For more details for the equilibrium candidates can found in (Faramarzi et al., 2020). Figure 1 presents the flowchart of the EO algorithm.

Figure 1 The EO mode (see online version for colours)



3 Experimental results

3.1 *Instance generation and categories*

According to Uchoa et al. (2017), the CVRP instances can be modified with the following features:

1 placement and the number of customers

2 placement of deposit

3 distribution of demand

4 average route size or several routes, defined by vehicle capacity.

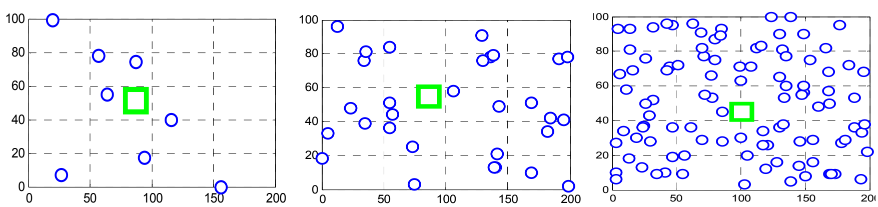
All instances generated by trying to sample a representative set of instances, randomising the choice of features for a particular instance within reasonable limits. By defining these limits, the definition of different instance classes could arrive automatically. All instances categorised into three categories. Small instance set $(Category\,A)$ with 8–25 customers, medium instance set $(Category\,B)$ with 30–70 customers, and a larger instance set $(Category\,C)$ with $100-250$ customers. Note that all customers randomly and evenly located on a square grid—also, Customer clustering not considered in these experiments.

All instances used in this paper generated and solved on *HP laptop ProBook 6360b* with $Intel\,Processor\,Core\,i5-2410M$ & $CPU\,2.30\,GHz$ and $6\,GB\,RAM$. The experiments and algorithms coded in *MATLAB 2013a*. Table 1 contains all parameter settings used for each algorithm, while Table 1 includes the details for the used instances. Where (*Depot Position*) is the position of the depot in which all vehicles start and back to it. For example, Figure 2 shows the instances $S1, S6,$ and $S11$, respectively, where the Blue circles represent the customers' positions, and the Green square is the depot position.

Table 1 Instances details

Instance	*n*	*M*	*Depot position* (X_Q, Y_Q)
S1	8	3	[93,47]
S2	10	3	[88,49]
S3	14	4	[93,47]
S4	20	4	[111,40]
S5	25	5	[117,42]
S6	30	5	[86,55]
S7	40	6	[114,42]
S8	50	7	[109,51]
S9	60	7	[82,45]
S10	70	8	[98,50]
S11	100	10	[101,45]
S12	250	15	[83,45]

Figure 2 Sample of instances: (a) S1 instance; (b): S6 instance and (c): S11 instance (see online version for colours)

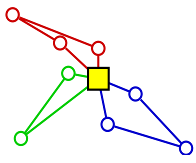
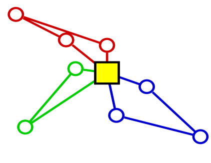
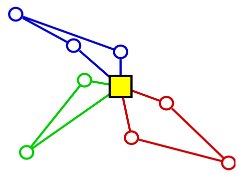
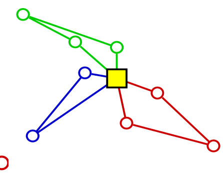
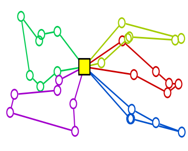
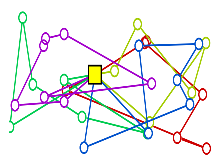
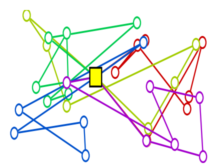
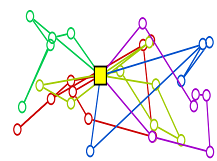
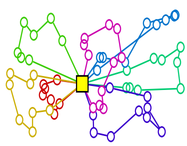
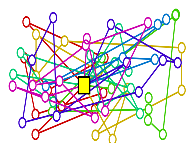
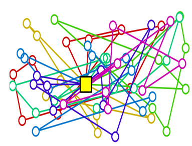
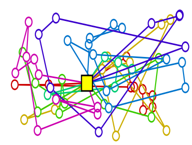
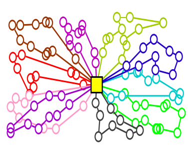
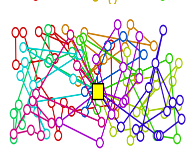
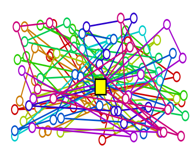
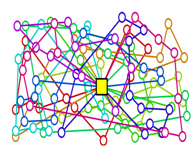


(a) (b) (c)

3.2 Graphical results

Table 2 contains graphical results for instances (S1, S6, S9, and S11) for EO, ABC, PSO, and WOA algorithms, respectively. Each figure demonstrates constructed paths for each vehicle in each instance. Each colour represents one path for one vehicle, the small circles represent the customers, and the Yellow square represents the depot. The EO algorithm in each iteration construct solution. This solution contains paths for each vehicle that starts from the depot, serve its assigned customers, and return to the depot at the end. At each iteration, each path updated based on minimising the cost and constructing the smallest path. The EO algorithm guaranteed that all customer's demands must be served; in other words, there are no customers who may not be served. As in Table 2, instance $S1$ *as an example*, all algorithms constructed excellent paths. For all other instances (from $S2$ to $S12$), the EO algorithm gives shorts, clear, simple, and smallest cost paths for each vehicle while competitors could not.

Table 2 Graphical results for the EO algorithm and its competitors for all instances ($S1, S6, S9$, and $S11$) (see online version for colours)

Instance	*EO*	*ABC*	*PSO*	*WOA*
$S1$				
$S6$				
$S9$				
$S11$				

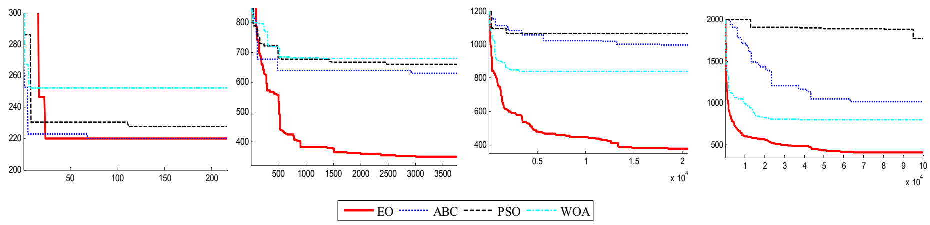
3.3 Convergence curves

Figure 3 contains some convergence curves for instances (S1, S6, S9, and S11). The y-axis is the *BestCost*, while the x-axis is the maximum number of function evaluations (*FunctionEvals*), which depending on the number of customers and the number of vehicles. The *FunctionEvals* calculated as follows:

$$FunctionEvals = m^3 * n,$$

the *FunctionEvals* differ from instance to another, and it increased while the number of customers increased, or the number of vehicles increased. All algorithms controlled to run for a fixed number of *FunctionEvals* to get a fair comparison between each of them. As clear in Figure 3, all curves prove the superiority of the EO algorithm over its competitors.

Figure 3 Convergence curves for instances ($S1, S6, S9$, and $S11$) (see online version for colours)



3.4 Analysis of statistical results

All algorithms run for ten times for all instances, and the best analytical results presented in Tables 3–6. Table 4 contains the Best Cost obtained ($Best\,Cost$) and the CPU time in Seconds took by each algorithm to form a feasible solution (CPU) for WOA, PSO, ABC, and EO algorithms for each instance. For instance, $S1$, ABC, and EO algorithms obtained the same costs ($Best\,Cost = 220.163$) (first row in Table 3). As indicated in Table 3, the EO algorithm obtained $Best\,Cost$ values over its competitors for all instances used in this comparison and also for the CPU time. Table 4 contains the mean values of best cost vectors ($Mean$) and the stander deviation (STD). For instances $S1$ and $S2$, the ABC algorithm got the smallest $Mean$ values. But for all other instances, the EO algorithm got the smallest Mean values.

Table 3 Best cost and CPU time results for the EO algorithm and its competitors

Instance	*WOA*		*PSO*		*ABC*		*EO*	
	Best Cost	*CPU(s)*	*Best Cost*	*CPU(s)*	*Best Cost*	*CPU(s)*	*Best Cost*	*CPU(s)*
$S1$	252.071	26.4688	227.652	5.7031	**220.163**	10.0469	**220.163**	**0.125**
$S2$	313.036	142.25	291.298	7.5313	291.109	12.8125	**284.979**	**0.10938**
$S3$	383.465	153.375	281.571	25.2656	310.709	44.0781	**275.402**	**0.375**
$S4$	443.920	397.8438	401.516	40.5313	401.761	72.6406	**340.125**	**0.59375**
$S5$	540.078	198.5781	468.027	97.375	498.399	174.4219	**327.709**	**1.3438**
$S6$	678.602	359.7656	659.668	133.1563	628.305	236.7969	**347.785**	**1.8281**
$S7$	674.214	829.2344	733.091	330.3594	746.895	599.3594	**364.251**	**4.6719**
$S8$	789.530	11967.25	903.404	727.5469	885.133	1345.437	**370.333**	**9.875**
$S9$	838.258	13580.91	1066.56	907.0781	996.625	1649.891	**379.702**	**11.6875**
$S10$	630.249	14284.094	1278.14	1461.5	885.525	2469.031	**376.823**	**22.75**
$S11$	799.702	96384.625	1771.58	4768.031	1019.60	7741.718	**413.065**	**78.25**
$S12$	1549.003	747392.35	4417.31	55292.45	3316.71	79884.82	**500.825**	**913.078**

The main objective of optimising the CVRP problem is to find the shortest path (distance) to visit the assigned customers and go back to the depot for all vehicles. Table 4 contains all distances travelled by all vehicles for each algorithm for all instances. Where $maxD$ is the longest distance covered by one vehicle and remainder vehicles less than or equal to $maxD$. The $totalD$ is the total distance for all vehicles will be cover from the depot to all customers and return to the depot. All best results highlighted in Bold format in Table 5. As an example, for instance $S1$, all algorithms got equals $maxD$ $(maxD = 186.59)$ and $totalD$ $(totalD = 522.36)$. In instance $S3$, the PSO algorithm obtained the smallest $maxD$ $(maxD = 217.6)$, but the smallest $totalD$ $(totalD = 786.31)$ obtained by the EO algorithm. For all instances, The EO algorithm got the smallest $totalD$ except instance $S12$ obtained by the WOA algorithm $(totalD = 6512.6)$ and got the smallest $maxD$ except instance $S3$ obtained by the PSO algorithm $(maxD = 217.6)$. These results in Table 5 confirm that the EO algorithm can always get the shortest paths over its competitors in all conditions.

Table 4 Mean, STD, and function evaluations for the EO algorithm and its competitors

Instance	*WOA*		*PSO*		*ABC*		*EO*		*Function-Evals*
	Mean	*STD*	*Mean*	*STD*	*Mean*	*STD*	*Mean*	*STD*	
$S1$	253.018	9.026	231.152	10.893	**221.771**	**5.464**	256.196	159.740	216
$S2$	315.551	15.996	317.312	35.478	**305.593**	**20.466**	361.536	185.086	270
$S3$	383.465	**3.179**	307.496	43.157	328.433	25.782	**296.966**	87.837	896
$S4$	452.373	32.731	416.790	46.575	419.901	**29.613**	**367.087**	50.733	1280
$S5$	583.234	60.631	504.602	45.914	520.021	**33.143**	**383.295**	160.641	3125
$S6$	691.688	38.032	679.316	47.103	645.484	**37.827**	**420.196**	199.807	3750
$S7$	698.804	75.3307	789.978	59.858	784.938	**47.756**	**426.008**	144.139	8640
$S8$	802.228	**34.233**	909.880	41.726	945.731	63.643	**440.990**	157.062	17150
$S9$	850.132	47.729	1073.29	49.536	1032.887	**46.944**	**477.780**	198.198	20580
$S10$	675.173	86.010	1330.50	**62.004**	1071.156	166.244	**470.407**	189.573	35840
$S11$	844.856	116.109	1902.08	**54.388**	1211.289	285.284	**498.353**	167.252	100000
$S12$	1697.634	270.809	4482.83	**80.539**	3517.628	189.946	**631.447**	211.115	843750

Table 5 All distances travelled by all vehicles for the EO algorithm and its competitors

Instance	*WOA*		*PSO*		*ABC*		*EO*	
	maxD	*totalD*	*maxD*	*totalD*	*maxD*	*totalD*	*maxD*	*totalD*
$S1$	**186.59**	**522.36**	**186.59**	**522.36**	**186.59**	**522.36**	**186.59**	**522.36**
$S2$	244.14	715.70	244.14	715.70	244.24	712.95	**239.52**	**694.14**
$S3$	286.94	905.19	**217.60**	857.33	223.48	868.04	218.63	**786.31**
$S4$	305.46	1162.1	324.34	1096.1	320.74	1130.9	**275.41**	**922.57**
$S5$	386.35	1598.4	343.32	1590.4	334.07	1511.6	**246.65**	**1057.3**
$S6$	483.48	1970.7	462.03	2113.9	472.97	2026.3	**254.43**	**1188**
$S7$	481.53	2408.3	528.8	2571.8	511.74	2863.3	**254.53**	**1351.8**
$S8$	509.64	2987.9	612.55	3521.1	585.54	3581.5	**247.72**	**1473.8**
$S9$	563.57	3201.7	700.75	4078.9	653.63	4083.6	**259.18**	**1464.4**
$S10$	394.2	2692.8	733.16	5259.3	509.52	3797.8	**242.84**	**1582.7**
$S11$	447.72	3901.1	1006.6	7758.2	564.72	4888.2	**228.07**	**2078**
$S12$	518.4	**6512.6**	1821.6	20850	1535.9	17945	**274.41**	3138.6

Table 6 WSR test for the EO algorithm and its competitors

Instance	*p-value*		
	EO VS. WOA	*EO VS. PSO*	*EO VS. ABC*
*S*1	0.0234	1	1
*S*2	0.0020	0.1934	0.1602
*S*3	0.0039	0.0840	0.0098
*S*4	0.0020	0.0020	0.0020
*S*5	0.0020	0.0020	0.0020
*S*6	0.0020	0.0020	0.0020
*S*7	0.0020	0.0020	0.0020
*S*8	0.0020	0.0020	0.0020
*S*9	0.0840	0.0020	0.0020
*S*10	0.0297	0.014	0.0625
*S*11	0.0020	0.0020	0.0020
*S*12	0.0172	0.0194	0.0156

To test the performance of the EO algorithm, the Wilcoxon Signed-Rank (WSR) test used to estimate the statistically significant variation between any two algorithms. The statistical results for WSR presented in Table 6. The results show the superiority of the EO algorithm over its competitors at a 95% significance level $(a = 0.05)$.

4 Conclusion and future work

This paper proposed the EO algorithm for solving the CVRP model. The computational results proved that the EO algorithm has superior over the state-of-art algorithms. The EO algorithm perform better than the ABC, PSO, and WOA algorithms. The advantages of this study are it always finds the smallest best costs, shorts paths, best graphical results, best solutions, best convergence curves, and best mean values over its competitors. The future work for this paper can be using the EO algorithm to solve other versions of the VRP problem. For example, the dynamic vehicle routing problem, the multi depot vehicle routing problem, and the Feeder vehicle routing problem.

References

Afifi, S., Dang, D.C. and Moukrim, A. (2013) 'A simulated annealing algorithm for the vehicle routing problem with time windows and synchronization constraints', *Lect. Notes Comput. Sci* (*Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics*), Vol. 7997, LNCS, pp.259–265, doi: 10.1007/978-3-642-44973-4_27.

Brandão, J. (2004) 'A tabu search algorithm for the open vehicle routing problem', *Eur. J. Oper. Res.*, Vol. 157, No. 3, pp.552–564, doi: 10.1016/S0377-2217(03)00238-8.

Chen, C. and Zhou, K. (2018) 'Application of artificial bee colony algorithm in vehicle routing problem with time windows', *2018 Int. Conf. on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, IEEE, pp.781–785, doi: 10.1109/Sdpc.2018.8664999.

Damicis, J. (2018) *Recent and Emerging Trends in Transportation and Logistics*, [Online]. Available https://www.camoinassociates.com/recent-and-emerging-trends-transportation-and-logistics (Accessed 18 July, 2020)

Das, S., Abraham, A. and Konar, A (2008) 'Swarm intelligence algorithms in bioinformatics', *Stud. Comput. Intell.*, Vol. 94, No. 2008, pp.113–147, doi: 10.1007/978-3-540-76803-6_4.

Deloitte Research (2017) *China Logistics Industry Investment Promotion Report*, May, pp.1–34.

Dommel, H.W. and Tinney, W.F. (1968) 'Optimal power flow solutions', *IEEE Trans. Power Appar. Syst.*, Vol. PAS-87, No. 10, pp.1866–1876, doi: 10.1109/TPAS.1968.292150.

Elshaer, R. and Awad, H. (2020) 'A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants', *Comput. Ind. Eng.*, Vol. 140, p.106242, doi: 10.1016/j. cie.2019.106242.

Faramarzi, A., Heidarinejad, M., Stephens, B. and Mirjalili, S. (2020) 'Equilibrium optimizer: a novel optimization algorithm', *Knowledge-Based Syst.*, Vol. 191, doi: 10.1016/j. knosys. 2019.105190.

Fares, I., Rizk-allah, R.M., Hassanien, A.E. and Vaclav, S. (2020) *Multiple Cyclic Swarming Optimization for Uni-and Multi-Modal Functions*, pp.887–898, doi: 10.1007/978-981-15-1286-5.

Gendreau, M., Iori, M., Laporte, G. and Martello, S. (2008) 'A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints', *Networks*, Vol. 51, No. 1, pp.4–18, doi: 10.1002/net.20192.

Gong, Y.J., Zhang, J., Liu, O., Huang, R.Z., Chung, H.S.H. and Shi, Y.H. (2012) 'Optimizing the vehicle routing problem with time windows: a discrete particle swarm optimization approach', *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, Vol. 42, No. 2, pp.254–267, doi: 10.1109/TSMCC.2011.2148712.

Hadavandi, E., Mostafayi, S. and Soltani, P. (2018) 'A grey wolf optimizer-based neural network coupled with response surface method for modeling the strength of siro-spun yarn in spinning mills', *Appl. Soft Comput. J.*, Vol. 72, pp.1–13, doi: 10.1016/j. asoc.2018.07.055.

Hafezi, R., Shahrabi, J. and Hadavandi, E. (2015) 'A bat-neural network multi-agent system (BNNMAS) for stock price prediction: case study of DAX stock price', *Appl. Soft Comput. J.*, Vol. 29, pp.196–210, doi: 10.1016/j. asoc.2014.12.028.

Heuristics, R., Cordeau, J-F., Gendreau, M., Laporte, G., Potvin, J-Y. and Semet, F. (2002) 'A guide to vehicle', *Source J, Oper. Res. Soc.*, Vol. 53, No. 5, pp.512–522, doi: 10.1057/ palgrave/jors/2601319.

Huang, Y.H., Blazquez, C.A., Huang, S.H., Paredes-Belmar, G. and Latorre-Nuñez, G. (2019) 'Solving the feeder vehicle routing problem using ant colony optimization', *Comput. Ind. Eng.*, Vol. 127, November, pp.520–535, doi: 10.1016/j. cie.2018.10.037.

Karaboga, D. (2005) *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Tech. Rep. TR06, Erciyes Univ., no. TR06, p. 10, 2005, doi: citeulike-article-id:6592152.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', *Neural Networks 1995 Proceedings., IEEE Int. Conf.*, Vol. 4, pp.1942–1948, doi: 10.1109/ICNN.1995.488968.

Lin, C.L., Mimori, A. and Chen, Y.W. (2012) 'Hybrid particle swarm optimization and its application to multimodal 3D medical image registration', *Comput. Intell. Neurosci.*, Vol. 2012, doi: 10.1155/2012/561406.

Marinakis, Y., Marinaki, M. and Dounias, G. (2010) 'A hybrid particle swarm optimization algorithm for the vehicle routing problem', *Eng. Appl. Artif. Intell.*, Vol. 23, No. 4, pp.463–472, doi: 10.1016/j. engappai.2010.02.002.

Marinakis, Y., Marinaki, M. and Migdalas, A. (2018) *Particle Swarm Optimization for the Vehicle Routing Problem: A Survey and a Comparative Analysis*, Vol. 2-2.

Mirjalili, S. and Lewis, A. (2016) 'The whale optimization algorithm', *Adv. Eng. Softw.*, Vol. 95, pp.51–67, doi: 10.1016/j. advengsoft.2016.01.008.

Montané, F.A.T. and Galvão, R.D. (2006) 'A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service', *Comput. Oper. Res.*, Vol. 33, No. 3, pp.595–619, doi: 10.1016/j. cor.2004.07.009.

Mukherjee, B. (2017) #5 *Key Areas Logistics Businesses Should Focus for Growth* (2017) [Online]. Available: https://www.entrepreneur.com/article/297992 (Accessed 18 July, 2020).

Normasari, N.M.E., Yu, V.F., Bachtiyar, C. and Sukoyo (2019) 'A simulated annealing heuristic for the capacitated green vehicle routing problem', *Math. Probl. Eng.*, Vol. 2019, doi: 10.1155/2019/2358258.

Rizk-Allah, R.M. and Hassanien, A.E. (2018) 'New binary bat algorithm for solving 0-1 knapsack problem', *Complex Intell. Syst.*, Vol. 4, No. 1, pp.31–53, doi: 10.1007/s40747-017-0050-z.

Rizk-Allah, R.M., Hassanien, A.E. and Bhattacharyya, S. (2018) 'Chaotic crow search algorithm for fractional optimization problems', *Appl. Soft Comput. J.*, Vol. 71, pp.1161–1175, doi: 10.1016/j. asoc.2018.03.019.

Rizk-Allah, R.M., Hassanien, A.E. and Elhoseny, M. (2018) 'A multi-objective transportation model under neutrosophic environment', *Comput. Electr. Eng.*, Vol. 69, pp.705–719, doi: 10.1016/j. compeleceng.2018.02.024.

Savills Investment Management (2016) *European Logistics: Warehousing the Future*, [Online]. Available http://www.savillsim.com/documents/2017-sim-european-logistics-warehousing-the-future-final.pdf (Accessed 18 July, 2020).

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T. and Subramanian, A. (2017) 'New benchmark instances for the capacitated vehicle routing problem', *Eur. J. Oper. Res.*, Vol. 257, No. 3, pp.845–858, doi: 10.1016/j. ejor.2016.08.012.

Vidal, T., Crainic, T.G., Gendreau, M. and Prins, C. (2013) 'Heuristics for multi-attribute vehicle routing problems: a survey and synthesis bureaux de montréal□: bureaux de québec', *Eur. J. Oper. Res.*, Vol. 231, No. 1, pp.1–21.

Yao, B., Yan, Q., Zhang, M. and Yang, Y. (2017) 'Improved artificial bee colony algorithm for vehicle routing problem with time windows', *PLoS One*, Vol. 12, No. 9, pp.1–18, doi: 10. 1371/journal. pone.0181275.

Nomenclature and Abbreviations

Nomenclature			
n	Total number of customers	d_i	Customer i demand, $d_0 = 0$
m	Total number of vehicles	X_{ij}^k	A variable $\in [0,1]$
C_{ij}	Travelling cost from the customer i to the customer j	R	Set of customers visited by a vehicle
S_i	The time required to serve the customer i, $S_0 = 0$	$\lvert R \rvert$	Count of elements in R
Q	Total capacity for each vehicle	T_i	Current temperature
T	The greatest travelling distance of a vehicle	*rand*	A random number in $[01]$

Abbreviations			
EO	Equilibrium optimiser	VRP	Vehicle routing problem
ABC	Artificial bee colony	CVRP	Capacitated vehicle routing problem
PSO	Particle swarm optimisation	WSR	Wilcoxon Signed-Rank
WOA	Whale optimisation algorithm		