Journal of Theoretical and Applied Information Technology
10th June 2016. Vol.88. No.1

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

CUCKOO SEARCH ALGORITHM FOR CAPACITATED VEHICLE ROUTING PROBLEM

[1]MANSOUR ALSSAGER, [2]ZULAIHA ALI OTHMAN
[1,2]Data Mining and Optimization Group, Faculty of Information Sciences and Technology
University Kebangsaan Malaysia, 43600 Bangi, Selangor Darul Ehsan, Malaysia
E-mail: [1]mansour.alssager@gmail.com, [2]zao@ftsm.ukm.com

ABSTRACT

The cuckoo search algorithm is a novel metaheuristic based on the reproduction behavior of some cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies. It has been applied to a wide range of optimization problems with good performance such as a TSP; nonetheless, it has yet to be applied to the vehicle routing problem, specifically the capacitated vehicle routing problem. In this study cuckoo search is proposed to solve the capacitated vehicle routing problem. The proposed algorithm uses a set of twelve neighborhood structures process based on λ-interchanges scheme and cross-exchange operators with the aim of gaining significant improvements in solution quality. The result shows that the performance of the proposed algorithm is comparable to that of the other algorithms and is able to find near-optimal solution in reasonable time, which indicates that cuckoo search has the potential to solve the capacitated vehicle routing problem.

Keywords: *Optimization, Cuckoo Search, Capacitated Vehicle Routing Problem, Neighborhood Structures, Levy Flight.*

1. INTRODUCTION

Various metaheuristic have been successfully applied to solve the vehicle routing problem (VRP). However, as yet none of these algorithms have been able to reach the optimal solution across all the benchmark datasets and the community is still looking for a more stable and robust algorithm. The main challenge of the VRP is to design least-cost (distance, time) routes for a fleet of vehicles to serve geographically scattered clients. There are many types of VRP, which are classified according to different applications and restrictions. One of these is the capacitated VRP (CVRP) in which the total demand for any vehicle cannot exceed a preset capacity value.

As the CVRP is classified as a NP-hard problem based on the theory of computational complexity, various approaches have been presented to solve the CVRP, and these can be divided into two types: exact and metaheuristic algorithms. Exact algorithms such as branch and bound and dynamics programming which able to obtain the optimal solution [1, 2]. However, their performance is limited to small-sized problems with reasonable time. Metaheuristics is a category of algorithm that has been developed based on biology, physics and artificial intelligence and has been shown to have efficient optimization performance in various fields. They can be classified into two main types: single and population based methods.

Various metaheuristics have been presented to solve the CVRP such as bees mating optimization algorithm [3]. electromagnetism-like algorithm [4], Ant Colony Optimization [5, 6], artificial bee colony algorithm [7, 8], Tabu Search [9, 10], the Genetic Algorithm [11], Particle Swarm Optimization [12-14], Adaptive Memory Programming [15], water flow alike algorithm [16], membrane algorithm [17], cooperative parallel metaheuristic [18].

A successful metaheuristics approach will find a balance between exploring new parts of the search space and refining areas of the search space where current information suggests the minimum might be located. Recently, a new metaheuristic known as cuckoo search (CS) has received much attention from researchers in various optimization areas. It was introduced by Yang and Deb in 2009. Cuckoo search has shown better performance compared to GA and PSO. It is characterized by three key components: (i) a simple selection strategy; (ii) more efficient randomization as it uses so-called Lévy flight, which preserves the step length (whether large or small) and so there is a better balance between exploration and exploitation; and (iii) fewer parameters to be tuned, which means it is potentially more generic and can adapt to a wider

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

range of optimization problems [19]. In addition, in CS each nest can have many eggs and thus it can represent a set of solutions as it can be extended into a type of meta-population algorithm, or even a hyper-heuristic algorithm [20]. However, to the best of our knowledge no work has attempted to apply CS to solving the CVRP. Therefore, this research proposes a method of CS to solve the CVRP.

The remainder of this paper is organized as follows: In section 2, a brief literature review of CS and its application is provided. In section 3, the proposed CS algorithm and how it is applied to the CVRP is presented. In section 4, an experimental result for selecting the algorithm parameter is presented followed by the results of an experiment to compare the performance of the proposed CS algorithm with that of other metaheuristics. Finally, Section 5 concludes the study.

2. LITERATURE REVIEW

Cuckoo search has been applied to various optimization problems ranging from continuous to discrete optimization problems and from single-objective to multi-objective problems. In addition, several attempts have been made to improve the basic algorithm. Cuckoo search has been shown to be very effective in solving continuous optimization problems such as welded beam design problems [21] and spring design [22]. In addition, CS exhibits superior performance in a range of constrained optimization problems such as business optimization applications [23], single-objective optimum synthesis of a six-bar double dwell linkage [24] and for phase equilibrium and stability calculations, where it has been shown to be a reliable method for solving thermodynamic calculations [25].

Cuckoo search has also gained much attention in relation to solving discrete optimization problems in various domains. For instance, Pop, Chifu [26] proposed a CS hybrid algorithm for selecting optimal web service composition, Vazquez [27] used CS to train a spiking neural network and Layeb [28] proposed a new approach based on a quantum-inspired CS algorithm to deal with the basic problem in one-dimensional bin packing. The following year, Chifu, Pop [29] also presented a CS for selecting the optimal web service composition. More recently, Burnwal and Deb [30] proposed a CS-based approach for solving the flexible manufacturing system scheduling problem, while Ouaarab, Ahiod [31] proposed an extended and improved version of the standard CS to solve the traveling salesman problem (TSP).

Several studies have been conducted in attempts to improve CS. For instance, the study by Walton, Hassan [32] included a modification of the CS where a crossover between the solution is added to gain part of the other solution properties. Also Soneji and Sanghvi [33] compared the simplified version of the CS algorithm with a modification where Lévy flight is modified using Mantegna's algorithm and McCulloch's algorithm. This modification involves the addition of information exchange between the top eggs, or the best solutions.

As a further extension, Yang and Deb [34] produced a multi-objective CS for design engineering applications. For multi-objective scheduling problems, significant progress was made by Chandrasekaran and Simon [35] using a CS algorithm.

3. THE BASIC CUCKOO SEARCH FOR THE CAPACITATED VEHICLE ROUTING PROBLEM

Cuckoo search is one of the latest nature-inspired metaheuristic algorithms that belong to the swarm intelligence category. Recent studies show that CS is potentially far more efficient than PSO and GA [20, 21] for continuous search space problem. The algorithm is inspired by the reproduction behavior of cuckoo birds. Cuckoos are fascinating birds not only because of the beautiful sounds they can make, but also because of their aggressive reproduction strategy. Some species of cuckoos lay their eggs in communal nests, though they may remove others' eggs to increase the hatching probability of their own eggs. Quite a number of species engage in obligate brood parasitism by laying their eggs in the nests of other host birds (often other species). If a host bird discovers the eggs are not its own, it will either throw away these alien eggs or simply abandon its nest and build a new nest elsewhere. These characteristics are utilized in the CS approach because, after each step, the worst solutions are discarded and new solutions are generated, as if the worst nests are being identified by host birds so they have to be discarded and the new nests are searched for by host birds [20], and then in each iteration a cuckoo solution tries to replace a nest among the solution nests to get the best solution after each repetition. The strength of CS lies in how the cuckoo exploits and explores the solution space. The algorithm can be summarized based on the following ideal rules:

Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;

Journal of Theoretical and Applied Information Technology
10th June 2016. Vol.88. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The best nests with high-quality eggs will be carried over to the next generations;

The number of available host nests is fixed and there is always a probability that the cuckoo egg will be discovered by the host. If an egg is discovered, the host bird throws it away. This effect is approximated by discarding a fraction of the eggs and replacing them at each generation.

Essentially, these rules provide a selection process for the optimization algorithm, ensuring the best eggs survive from generation to generation. To complete the algorithm, a method of generating the eggs is required. This is where Lévy flight is applied. A cuckoo i generates a new solution $x_i(t+1)$ via Lévy flights, according to Eq. 6 [7]:

$$x_i(t+1)=x_i(t)+\alpha \oplus \text{Lévy}\,(s, \lambda) \qquad (6)$$

where α is the step size that follows the Lévy distribution that is shown in Eq. (7):

$$\text{Lévy}(s,\lambda) \sim s -\lambda ,(1 < \lambda \leq 3) \qquad (7)$$

This has an infinite variance with an infinite mean. Here, s is the step size drawn from a Lévy distribution. A detailed description of the CS can be found in [20, 21].

4. PROPOSED CUCKOO SEARCH FOR THE CAPACITATED VEHICLE ROUTING PROBLEM

The adaptation of using the CS with the discrete solution space of CVRP is similar to the adaptation of Ouaarab, Ahiod [31] since its shows better performance for TSP. The CS works with a population of candidate solution eggs. At each generation, the n best individuals of the cuckoo eggs which is not discovered by the host bird nest are selected and ranked based on their fitness. There are considerably large neighborhoods structures have been used as a local search mechanism to improve the cuckoo eggs fitness to make the cuckoo eggs looks like host nest eggs and therefore have a better chance to survive and not discovered as an alien eggs. The details of CS procedure are outlined as follows:

4.1. Egg Representation

Assuming that a cuckoo lays a single egg in one nest, we can say that one egg in a nest is a solution represented by one individual in the population. An egg can also be one new candidate solution laid by a cuckoo for a place/location reserved by an individual in the population, while the nest is the container of that new cuckoo egg.

In CS the number of nests is fixed and this number represents the size of the population. A nest is a container of an individual of the population and its abandonment involves its egg being replaced in the population by a new one. Obviously, a nest can have multiple eggs for future extensions, but in this study each nest contains only one egg.

In the CVRP, we can say that an egg is the equivalent of a Hamiltonian cycle of served routes that start and end at the central depot. The egg representation adopted in this study can be described as follows: Assume a CVRP with n costumers and v available vehicles for delivery, then the number 0 denotes the depot and 1, 2,… n denotes the costumers. Based on the v vehicles at the depot, so each egg has at most the v distribution path (route), every path (route) starts at the depot and stops at the depot. Example of possible routes is shown in Figure 1.

	Depot	Customer					
Route 1	**0**	2	3	4	**0**		
Route 2	**0**	6	7	1	8	5	**0**
	.	.	.	.	.	.	
	.	.	.	.	.	.	
	.	.	.	.	.	.	
	.	.	.	.	.	.	
Route n	**0**	12	16	32	53	**0**	

Figure 1. Egg Representation.

4.2. Host Nest Initialization

The initial host bird eggs are generated using a cheapest insertion heuristic, whereby cheapest customer with minimum traveling cost sequentially inserted to its respective route until all vehicles are full. The main interest is to build initial quality solutions using relatively simple schemes.

4.3. The Neighborhood Structures and the Lévy Flight

A multiple neighborhood structures is used in this study to improve the cuckoo bird's eggs to imitate the host nest eggs in pattern and shape, and therefore have a good chance to survivor. These neighborhood structures consist of seven inter-route (two routes involved) and five intra-routes (only one route). Five of seven inter-route based on the λ-interchanges scheme [36], which consists of exchanging up to λ customers between two routes. In this study, λ = 2 considered due to high computational cost associated with large λ. The details of other can be found in [37-39]. The different neighborhood structures and their explanations can be outlined in Table 1. These

Journal of Theoretical and Applied Information Technology
10th June 2016. Vol.88. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

neighborhood structures need to be linked to the step length generated by the levy flight.

Table 1. Neighbourhood structures.

Name	*Category*	*Details*
SHIFT-1-0	inter-route	One customer is transferred from a route *one* to a route *two*.
SWAP-1-1	inter-route	Permutation between one customer from a route *one* and a one customer from a route *two* is swapped.
SHIFT-2-0	inter-route	Two adjacent customers are transferred from a route *one* to a route *two.*
SWAP-2-1	inter-route	Permutation of two adjacent customers from a route *one* swapped by a one customer from a route *two.*
SWAP-2-2	inter-route	Permutation between two adjacent customers from a route *one* swapped by another two adjacent customers from route *two*
CROSS	inter-route	the arc between two adjacent customers *i* and *j* belonging to a route *one*, and the one between *i'* and *j'* route *two* both are removed. Next an arc inserted connecting *i* and *j'* and another is inserted linking *i'* and *j*.
K-SHIFT	inter-route	A subset of consecutive customers is transferred from a route *one* to the end of a route *two.*
REINSERTION	intra-routes	One, customer is removed and inserted in another position of the route.
OR-OPT2	intra-routes	Two adjacent customers are removed and inserted in another position of the route.
OR-OPT3	intra-routes	Three adjacent customers are removed and inserted in another position of the route.
TWO-OPT	intra-routes	Two nonadjacent arcs are deleted and another two are added in such a way that a new route is generated.
EXCHANGE	intra-routes	Permutation between two customers is swapped.

Lévy flight is generated by a probability density function that has a power law tail. The Cauchy distribution is often used for this purpose [40]. The method we use to generate random number from a Lévy distribution is shown in Figure 2. A search of this type is frequently found in nature and is generally considered to represent the optimum random search pattern [20].

```
Procedure LévyRandomNumber() begin
double u, v
u=π*(U(0,1]-0.5)
{When α = 1, the distribution simplifies to Cauchy}
if α == 1 then
    return (c tan u)
end if
v=0
While v == 0 do
    v=-log(U(0,1])
end while
{When α = 2, the distribution defaults to Gaussian}
if α == 2 then
    return (2c √v sin(u))
end if
{The following is the general Lévy case}
Return c sin(αu)/cos(u)^(1/α) (cos(u(1 − x))/v)^(1−α/α)
End
```

Figure 2. Lévy Flight Via Catchy Distribution

In this study, the twelve neighborhood structures will associated with the step length generated by levy flight which can be categorized into small to medium (most frequent) or large (less frequent). A small change in eggs shape and color made by a neighborhood that change only small part of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

solution such as (SHIFT-1-0, SWAP-1-1, SHIFT-2-0, REINSERTION, and OR-OPT2), gradually followed by another neighborhoods that modify larger part of the solution such as (OR-OPT3, TWO-OPT, EXCHANGE and SWAP-2-1), up to the larger jump which associated with neighborhood that change even much larger part of the solution such as (SWAP-2-2, CROSS, K-SHIFT). To facilitate the control of these numbers, interval between 0 and 1 are assumed. Therefore, according to the value given by the Lévy flight in this interval we can choose the appropriate step length as follows:

If the value of Lévy is in:

- $[0,i]$, one step of small neighborhood structure performed.
- $[(k-1)\times i,\ k\times i]$, one step of medium neighborhood structure performed.
- $[k\times i,1]$, we perform a big neighborhood step

The value of i in this process is i=(1/(n+1)),where n is the max number of steps; and k is$\{2,...,n\}$. So, if we assume that n=12, then i =0.07, so the interval is divided into twelve parts as listed in Table 2:

Table 2. Levy Flight Association With Neighbourhood Structures.

	Levy no. generated	Neighborhood structures used
1	{0,i}=(0,0.07)	SHIFT-1-0
2	{i,i×2}=(0.07,0.14)	SWAP-1-1
3	{i×2,i×3}=(0.14,0.21)	SHIFT-2-0
4	{i×3,i×4}=(0.21,0.28)	REINSERTION
5	{i×4,i×5}=(0.28,0.35)	OR-OPT2
6	{i×5,i×6}=(0.35,0.42)	OR-OPT3
7	{i×6,i×7}=(0.42,0.49)	TWO-OPT
8	{i×7,i×8}=(0.49,0.56)	EXCHANGE
9	{i×8,i×9}=(0.56,0.63)	SWAP-2-1
10	{i×9,i×10}=(0.63,0.7)	SWAP-2-2
11	{i×10,i×11}=(0.7,0.77)	CROSS
12	{i×11,1}=(0.77,1)	K-SHIFT

In this study, the association of these steps to the neighborhood structure is set without any prior knowledge. Obviously, it's better to sequence them based on experimental knowledge. The pseudo code of the basic CS is presented in Figure 3.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Initialization
Setting the ***MaxIter***
Setting the host nest size $\boldsymbol{n}$
Setting the fraction of worse nests $\boldsymbol{p_a}$
For i←0:$\boldsymbol{n}$
    Initialize host nest Egg_i, add it to │**nest**│ presented in section **4.2**
End for
Calculate the initial host nest fitness, f(Egg_i)
Sort *Eggs* by order of its fitness
Calculate the abandon worse nests, $aban = \boldsymbol{p_a}*\boldsymbol{n}$
Set best cuckoo, $Egg_{best} = Egg_1$
iteration=0;
Improvements
While (*iteration*< ***MaxIter***) **do**
    Iteration=Iteration+1
    Select randomly host nest *Egg* form │**nest**│ to lay a new cuckoo egg
    Generate a new cuckoo *Egg'* by taking a Lévy flight from the selected host nest *Egg* as presented in subsection **4.3**
    Calculate fitness function for the new egg, f(*Egg'*)
    If (f(*Egg'*) < f(*Egg*)) then
        Egg ← *Egg'*
        f(*Egg*) ← f(*Egg'*)
    end if
    for all eggs to be abandoned < *aban* **do**
        Generate a new cuckoo *Egg'* by taking a Lévy flight from the selected host nest *Egg* as presented in subsection **4.3**
    End for
    Evaluate the fitness of the new eggs and rank all solutions
end while

Figure 3. The Proposed CS

5. COMPUTATIONAL EXPERIMENT

Experiments are conducted to test the performance of the CS on [14] instances, and have been used by [13] and [12]. There are 16 CVRP instances, the total number of clients varies from 30 to 135 clients, and the total number of vehicles varies from 3 to 10 vehicles. The locations of customers appear in some instances in clusters, while in other problems the customers are randomly scattered or semi clustered.
The experiments were performed on a 3.2 gigahertz Intel core i3 CPU, and the heuristics were coded using C++ in a Microsoft Visual Studio 2013 environment. The best solution (Min.), average solution (Avg.), standard deviation (Std.) are computed over 31 independent runs on each problem, along with the average computational time in seconds required to reach the final best solutions. The best solutions that are equal to the best-known solutions (BKS) for the benchmark problems are asterisked and shown in bold. The parameters setting in this study was set based on the suggestion from [20, 21] which is shown in the Table 3.

Table 3. Parameter Setting.

Parameter	*value*
the abandon of the worst nest(p_a)	0.25
The eggs size (n)	50
The iteration number (S)	20000

Experimental result of the proposed basic CS and other basic methods found in literature which are: GA of [41], SA of [14] and WFA of [16] is summarized in Table 4. The performance of the proposed basic CS is capable to get promising result and able to get near to optimal solution for some instances such as the A-n33-k5, B-n35-k5 and F-n72-k4, while the GA is able to get three optimal

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

solutions for A-n33-k5, B-n35-k5 and E-n30-k3 instances, and the SA is able to get two optimal solutions for A-n33-k5 and E-n30-k3 instances, there are only three instances available for the basic WFA, the other are not available (shown as dashed line). Moreover, the other three methods did not report the (avg. and std.) to carry out fair comparison and statistical analysis. From other prospective, the computational time of the basic CS is outperform the basic GA and SA for all instances with improvements in time reach up to 95% for some instances, (WFA is reported without computational time). Therefore, the basic CS is has potential to solve the CVRP.

Table 4. Computational Result For 16 Benchmark Problems.

Instance	BKS	Objective function						Computational time (s)		
		GA	SA	WFA	CS					
					Min.	Avg.	Std.	GA	SA	CS
A-n33-k5	661	**661**	**661**	-	688	692.0	16.67	39.6	38.2	3.391
A-n46-k7	914	928	931	-	973	994.76	8.05	136.4	143.8	8.782
A-n60-k9	1354	1360	1363	-	1414	1413.49	11.16	295.5	286.3	12.72
B-n35-k5	955	**955**	960	-	962	966.37	24.87	46.9	58.4	5.31
B-n45-k5	751	762	760	-	770	794.88	10.45	129.3	123.5	5.62
B-n68-k9	1272	1296	1298	-	1320	1326.73	15.39	396.2	409.2	24.44
B-n78-k10	1221	1248	1256	-	1284	1307.94	26.79	568.4	483.3	27.18
E-n30-k3	534	**534**	**534**	-	565	559.07	24.41	30.5	69.3	6.67
E-n51-k5	521	531	541	545	590	618.28	7.98	289.6	362.4	14.38
E-n76-k7	682	697	704	-	746	763.69	22.25	498.7	619.3	56.62
F-n72-k4	237	246	253	-	255	265.27	19.35	468.5	604.6	80.41
F-n135-k7	1162	1246	1243	-	1301	1319.58	11.32	1894.2	2533.9	1165.8
M-n101-k10	820	836	848	864	844	843.69	12.84	992.1	986.6	24.51
M-n121-k7	1034	1068	1081	1149	1088	1110.22	15.11	1643.1	2729.5	126.45
P-n76-k4	593	605	612	-	679	688.27	8.72	528.4	489.6	66.65
P-n101-k4	681	706	715	-	758	760.75	14.16	1213.2	1964.9	167.47

6. CONCLUSIONS

This study proposed a simple cuckoo search for solving the capacitated vehicle routing problem. Computational experiments on benchmark datasets demonstrated the effectiveness of the proposed approach compared to state of the art metaheuristics. Although our proposed approach produced acceptable and promising results in terms of solution quality when compared to existing techniques, it requires less computation time for most problem instances, so reducing the solution quality without compromising computation time calls for further research. However, while the CS could obtain near to optimal solutions for some CVRP, we could not find the best-known solutions for all problem instances. A deeper understanding of the CS mechanism is still needed in order to design a more effective way to improve solution quality.

REFERENCES:

[1] Laporte, G., *The vehicle routing problem: An overview of exact and approximate algorithms.* European Journal of Operational Research, 1992. 59(3): p. 345-358.

[2] Liong, C.Y., I. Wan, and O. Khairuddin, *Vehicle routing problem: Models and solutions.* Journal of Quality Measurement and Analysis, 2008. 4(1): p. 205-218.

[3] Marinaki, M., Y. Marinakis, and C. Zopounidis, *Honey bees mating optimization algorithm for financial classification problems.* Applied Soft Computing, 2010. 10(3): p. 806-812.

[4] Yurtkuran, A. and E. Emel, *A new Hybrid Electromagnetism-like Algorithm for capacitated vehicle routing problems.* Expert Systems with Applications, 2010. 37(4): p. 3427-3433.

[5] Meihua, W., et al., *Hybrid Ant Colony Optimization Algorithm for Two Echelon Vehicle Routing Problem.* Procedia Engineering, 2011. 15(0): p. 3361-3365.

[6] Lee, C.-Y., et al., *An enhanced ant colony optimization (EACO) applied to capacitated vehicle routing problem.* Applied Intelligence, 2010. 32(1): p. 88-95.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[7] Szeto, W.Y., Y. Wu, and S.C. Ho, *An artificial bee colony algorithm for the capacitated vehicle routing problem.* European Journal of Operational Research, 2011. 215(1): p. 126-135.

[8] Gomez, A. and S. Salhi. *Solving capacitated vehicle routing problem by artificial bee colony algorithm*. in *Computational Intelligence in Production and Logistics Systems (CIPLS), 2014 IEEE Symposium on*. 2014.

[9] Du, L. and R. He, *Combining Nearest Neighbor Search with Tabu Search for Large-Scale Vehicle Routing Problem.* Physics Procedia, 2012. 25(0): p. 1536-1546.

[10] Jin, J., T.G. Crainic, and A. Løkketangen, *A parallel multi-neighborhood cooperative tabu search for capacitated vehicle routing problems.* European Journal of Operational Research, 2012. 222(3): p. 441-451.

[11] Nazif, H. and L.S. Lee, *Optimised crossover genetic algorithm for capacitated vehicle routing problem.* Applied Mathematical Modelling, 2012. 36(5): p. 2110-2117.

[12] Kim, B.-I. and S.-J. Son, *A probability matrix based particle swarm optimization for the capacitated vehicle routing problem.* Journal of Intelligent Manufacturing, 2012. 23(4): p. 1119-1126.

[13] Ai, T.J. and V. Kachitvichyanukul, *A Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem.* Computers & Industrial Engineering, 2009. 56(1): p. 380-387.

[14] Chen, A., G. Yang, and Z. Wu, *Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem.* Journal of Zhejiang University SCIENCE A, 2006. 7(4): p. 607-614.

[15] Gounaris, C.E., et al., *An Adaptive Memory Programming Framework for the Robust Capacitated Vehicle Routing Problem.* Transportation Science. 0(0): p. null.

[16] Zainudin, S., M. Kerwad, and Z.A. Othman, *A WATER FLOW-LIKE ALGORITHM FOR CAPACITATED VEHICLE ROUTING PROBLEM.* Journal of Theoretical & Applied Information Technology, 2015. 77(1).

[17] Niu, Y., et al., *A novel membrane algorithm for capacitated vehicle routing problem.* Soft Computing, 2015. 19(2): p. 471-482.

[18] Jin, J., T.G. Crainic, and A. Løkketangen, *A cooperative parallel metaheuristic for the capacitated vehicle routing problem.* Computers & Operations Research, 2014. 44: p. 33-41.

[19] Yang, X.-S., *Bat Algorithm and Cuckoo Search: A Tutorial*, in *Artificial Intelligence, Evolutionary Computing and Metaheuristics*, X.-S. Yang, Editor. 2013, Springer Berlin Heidelberg. p. 421-434.

[20] Yang, X.-S. and S. Deb. *Cuckoo Search via Levy flights*. in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. 2009.

[21] Yang, X.-S. and S. Deb, *Engineering optimisation by cuckoo search.* International Journal of Mathematical Modelling and Numerical Optimisation, 2010. 1(4): p. 330-343.

[22] Gandomi, A., X.-S. Yang, and A. Alavi, *Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems.* Engineering with Computers, 2013. 29(1): p. 17-35.

[23] Yang, X.-S., et al. *Cuckoo search for business optimization applications*. in *Computing and Communication Systems (NCCCS), 2012 National Conference on*. 2012.

[24] Bulatović, R.R., S.R. Đorđević, and V.S. Đorđević, *Cuckoo Search algorithm: A metaheuristic approach to solving the problem of optimum synthesis of a six-bar double dwell linkage.* Mechanism and Machine Theory, 2013. 61(0): p. 1-13.

[25] Bhargava, V., S.E.K. Fateen, and A. Bonilla-Petriciolet, *Cuckoo Search: A new nature-inspired optimization method for phase equilibrium calculations.* Fluid Phase Equilibria, 2013. 337(0): p. 191-200.

[26] Pop, C.B., et al. *Cuckoo-inspired hybrid algorithm for selecting the optimal Web service composition*. in *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*. 2011.

[27] Vazquez, R.A. *Training spiking neural models using cuckoo search algorithm*. in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. 2011.

[28] Layeb, A., *A novel quantum inspired cuckoo search for knapsack problems.* Int. J. Bio-Inspired Comput., 2011. 3(5): p. 297-305.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[29] Chifu, V., et al., *Optimizing the Semantic Web Service Composition Process Using Cuckoo Search*, in *Intelligent Distributed Computing V*, F.M.T. Brazier, et al., Editors. 2012, Springer Berlin Heidelberg. p. 93-102.
[30] Burnwal, S. and S. Deb, *Scheduling optimization of flexible manufacturing system using cuckoo search-based approach.* The International Journal of Advanced Manufacturing Technology, 2013. 64(5-8): p. 951-959.
[31] Ouaarab, A., B. Ahiod, and X.-S. Yang, *Discrete cuckoo search algorithm for the travelling salesman problem.* Neural Computing and Applications, 2013: p. 1-11.
[32] Walton, S., et al., *Modified cuckoo search: A new gradient free optimisation algorithm.* Chaos, Solitons & Fractals, 2011. 44(9): p. 710-718.
[33] Soneji, H. and R.C. Sanghvi. *Towards the improvement of Cuckoo search algorithm*. in *Information and Communication Technologies (WICT), 2012 World Congress on*. 2012.
[34] Yang, X.-S. and S. Deb, *Multiobjective cuckoo search for design optimization.* Computers & Operations Research, 2013. 40(6): p. 1616-1624.
[35] Chandrasekaran, K. and S.P. Simon, *Multi-objective scheduling problem: Hybrid approach using fuzzy assisted cuckoo search algorithm.* Swarm and Evolutionary Computation, 2012. 5(0): p. 1-16.
[36] Osman, I., *Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem.* Annals of Operations Research, 1993. 41(4): p. 421-451.
[37] Taillard, É., et al., *A tabu search heuristic for the vehicle routing problem with soft time windows.* Transportation science, 1997. 31(2): p. 170-186.
[38] Nagy, G. and S.d. Salhi, *Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries.* European journal of operational research, 2005. 162(1): p. 126-141.
[39] Croes, G.A., *A method for solving traveling-salesman problems.* Operations Research, 1958. 6(6): p. 791-812.
[40] Husselmann, A. and K. Hawick, *Levy flights for particle swarm optimisation algorithms on graphical processing units.* Parallel and Cloud Computing, 2013. 2(2): p. 32-40.
[41] Wang, Z.Z., et al., *A hybrid optimization algorithm solving vehicle routing problems.* Operations Research and Management Science, 2004(52 (in Chinese)): p. 13:48.