IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

INTELLIGENT WATER DROPS ENHANCED ALGORITHM FOR VEHICLE ROUTING PROBLEMS WITH MULTIPLE DEPOTS

[1]Rohit Sharma, [2]Sanjay Saini

[1]Research Scholar, [2]Assistant Professor
[1]Dept. of Physics and Computer Science,
[1]Dayalbagh Educational Institute
Agra, India

Abstract: The intelligent water drop algorithm is a swarm-based metaheuristic algorithm inspired by the properties of river water droplets and the environmental changes caused by the moving river's movement. The technique has found applications in a broad variety of combinatorial and functional optimization problems since its inception as an alternative stochastic optimization approach. This article discusses an enhanced intelligent water drop method for resolving multi-depot vehicle routing problems MATLAB R2019b software was used to test a set of two separate Cordeau's benchmark examples (p04, p05) from the University of Malaga's online repository. The results were evaluated in terms of the depot's route length, optimal route, and optimal distance.

***Index Terms* - Metaheuristic algorithm, Multiple Depot Vehicle Routing Problem, Intelligent Water Drop algorithm.**

I. INTRODUCTION

MDVRP has been subjected to several heuristic algorithms. Classical heuristics and current metaheuristics are two types of heuristic algorithms. Due to the superiority of heuristic algorithms for large-scale issues, metaheuristic algorithms are mostly employed in the literature for MDVRP types.

ACO is a relatively recent optimization technique suggested by Dorigo et al. [1]. Ant Colony Optimization methods, which are inspired by the phenomena of ants finding the shortest path to food and hence fall under the category of metaheuristics, are extensively utilized in the literature to study MDVRP and its variations. Demirel and Yilmaz [2] suggested a method for solving MDVRP based on ACO. Yu et al. [3] demonstrated an improved ACO for MDVRP using a coarse-grain parallel method, an ant-weight technique, and a mutation operation. Stodola [4] discussed the modified MDVRP (M-MDVRP), in which the optimization criteria were changed. The usual MDVRP optimization criteria is to minimize the total sum of all vehicles' routes, but the M-MDVRP criterion aims to minimize the longest route of all vehicles, i.e., the vehicle's journey time is as short as feasible. Additionally, they devised a metaheuristic method for solving standard MDVRP cases based on Ant Colony Optimization theory- which was updated and adapted for M-MDVRP. They evaluated outcomes using Cordeau's benchmark cases. ACO was used in many studies [5,6,7] to solve versions of MDVRP, including MDVRP with multiple goals and MDVRP with time frames.

Additionally, the genetic algorithm is a kind of metaheuristic. Ho and colleagues [8] created two hybrid genetic algorithms (HGAs). The initialization mechanism of HGA1 is random. HGA2 generated initial solutions using the Clarke and Wright saving approach and the closest heuristic neighbor. To resolve the VRP with homogenous vehicles, Dantzig and Ramser [9] used a phased aggregation technique. Clarke and Wright [10] proposed an additional approach using several truck capacities. While Ho et al. used CWS to solve a single VRP, we divided the VRP into many Traveling Salesman Problems (TSPs) [11] and solved them using CWS. For the TSP, the CWS lifted the vehicle capacity constraint, allowing for easy implementation and near-optimal outcomes in a short period of time. Karakati and Podgorelec [12] conducted a study of genetic algorithms for MDVRP solution. The exact techniques, operators, and settings are offered to assist researchers and practitioners in further optimizing their solutions.

The majority of studies used the local search method. Local search is a refinement of the hill-climbing algorithm. Juan et al. [13] solved MDVRP using two biased-randomized processes that used the geometric probability distribution at various phases of the iterated local search (ILS) framework. Li et al. [14] solved the MDVRP with simultaneous deliveries and pickups (MDVRPSDP) and implemented an adaptive neighbourhood selection method in the improvement and perturbation phases of iterated local search, respectively. Ahmadi [15] developed a variable neighbourhood search method for solving the deterministic MDVRP model in order to streamline humanitarian logistics operations. Alinaghian and Shokouhi [16] devised a hybrid method that combines adaptive big neighbourhood search and variable neighbourhood search to handle this issue, which occurs when each vehicle's cargo area has many compartments, each of which is devoted to a certain kind of goods. Ahmadi Vidal et al. [17] used a genetic algorithm in conjunction with neighbourhood selection procedures to control variety in a vast class of time-window vehicle routing issues.

II. MULTI-DEPOT VEHICLE ROUTING PROBLEM

MDVRP issues [18] are conventional VRP problems with a comparable expression to the regular VRP problem. MDVRP utilises multiple depots and is far more difficult than single depot VRP. The fundamental objective of the multi-depot VRP is to minimise distances and routes, as well as available vehicles, and to keep costs down. MDVRP's goal functions are listed below [18, 19, 21, 20].

1. To keep the solution's cost as low as possible
2. To decrease travel distance
3. To limit the number of routes
4. To keep the vehicle's total size as small as possible
5. Fulfillment of client demand/satisfaction of customers.

The limitations on solving MDVRP are as follows:

1. Each client should not be serviced more than once
2. Each tour should depart from many depots situated in various locations.
3. The maximum number of vehicles available should not be exceeded.
4. The capacity of the vehicle shall not exceed the capacity limit.

After visiting the assigned consumers, each vehicle shall return to the depot where it was begun.

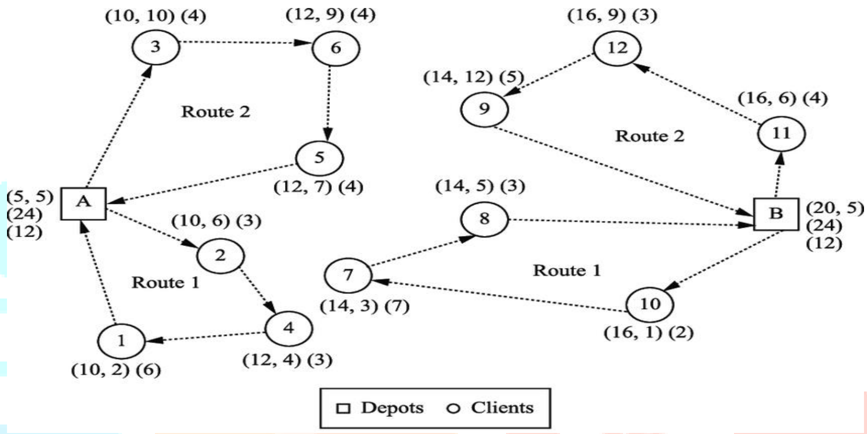

Fig. 2.1 Example of MDVRP using 2 depots

III. METHOD

We will now explain our suggested technique, but first, a concise description of the IWD algorithm will be given. Then, on the different benchmark Problems, use the suggested IWD method.

3.1 IWD Algorıthm

Shah Hosseini was the first to introduce the IWD algorithm (2009) [22]. The IWD algorithm is a genetic algorithm that is based on the driving of river schemes and activities that occur between water droplets in a waterway. The IWD method has been successfully used for a variety of difficult optimization issues [27]. [22] used the IWD algorithm to the issue of a traveling salesman and found near-optimal solutions. Hosseini further applied the IWD technique to a variety of difficult issues, including the multiple knapsack problems [22] and the clustering problem [23]. [24] Successfully substituted the IWD approach with the well-known Genetic Algorithm (GA) for different experimental functions in order to get multimodal peaks by demonstrating that IWD takes a shorter time to convergence. For optimizing COQUAMO, [25] employed the IWD method, which is a replication used to approximate the superiority of the software project. As a consequence of this change, a more accurate assessment of software development quality may be made.

IWDs have two critical qualities in the IWD method that regulate the process of determining the best possible route to their destination [26]. The water drop's velocity and the quantity of dirt it carries while traveling are these qualities. During the algorithm's execution, both characteristics change. Less dirt signifies higher velocity in the IWD algorithm, therefore IWDs pick paths with less soil on their connections. Sherylaidah Samsuddin et al. [28] proposes a method testing in which they have applied Ant colony optimization and IWD method both to solve MDVRP, Kumar, Saroj, et al.[29] introduced a genetic algorithm based IWD to solve optimal path optimization problem and [30] proposed a hybrid IWD method to solve timetable problem in examination.

An IWD progresses from a starting point to a destination. The IWD's speed and quantity of dirt are both 0 at first. During its journey, the water drop passes through its surroundings, removing some dirt and sometimes gaining speed. From one place to the next, the speed of IWD increases in a nonlinear, inversely proportional manner to the difference in soil level. As a result, an IWD in a route with less dirt moves quicker than one in a pathway with more soil. In IWD, the mechanism is to pick the path to its next point. The IWD prefers paths with low soil levels over pathways with high soil levels in this process. This route selection behaviour is accomplished by creating a uniform random distribution on the soils of the available pathways. The likelihood of selecting the next step is hence inversely linked to the soil level of the available pathways. As a result, pathways with lower soil levels have a better chance of being picked by the IWD.

Next, we specify the major steps of the IWD algorithm as explained by [22].

Step-1: Initialize the static parameters:

I_{max} = max no of iterations
N_{IWD} = no of water drops
a_v,b_v,c_v = velocity update parameters
a_s,b_s,c_s = soil update parameters
ρ_{IWD} = global soil update
ρ_n = local soil update
init_soil = initial soil on all paths soil(i,j)
init_vel = initialize velocity for IWDs vel_{iwd}

Step-2: Initialize dynamic parameter:

vel_{IWD} = {} (initially each IWD has an empty customer list)

Step-3: For each IWD

- Vehicle(k) start from the depot
- From customer node i to select the next node j (to build customer visited list)

Using the probability given by:

$$P_{IWD}(i,j) = \frac{f(soil(i,j))}{\Sigma_{m \notin vel_{IWD}} f(soil(i,m))} \quad (3.1)$$

$$f(soil(i,j)) = \frac{1}{\epsilon_s + g(soil(i,j))} \quad (3.2)$$

$$g(soil(i,j)) = \left\{ \begin{array}{ll} soil(i,j) & if\ min_{l \in vel_{IWD}}(soil(i,l)) \geq 0 \\ soil(i,j) - min_{l \in vel_{IWD}}(soil(i,l)) & otherwise \end{array} \right\} \quad (3.3)$$

Here, small positive constant: $\in_s$ ($\in_s$= 0.0001) prevents the divide-by-zero condition in eq.2.
Function min in equation 5.3, find the minimum of all the values.

- Check capacity constraint. If the constraint is violated then go to step 3(I).

$$\sum_{i=1}^{T_k} d_k\,(i) \ \leq Q\ \forall\ k \quad (3.4)$$

Here, the sum of demands served by k[th] vehicle is shown as $d_k(i)$, which should be less than or equal to the velocity capacity Q. Also, r_k is total number of customers served by vehicle k.

- Add the new node in the list vel_{IWD} of visited customers and update the following:

a. Velocity of IWD

$$vel_{IWD}(t+1) = vel_{IWD}(t) + \frac{a_v}{b_v + c_v.soil(i,j)} \quad (3.5)$$

b. Soil loaded from the path(i,j)

$$load\_soil(i,j) = \frac{a_s}{b_s + c_s.time(i,j;vel_{IWD}soil(t+1))} \quad (3.6)$$

Such that,

$$time(i,j;vel_{IWD}soil(t+1) = \frac{HUD(i,j)}{max(\in_v, vel_{IWD}(t+1))} \quad (3.7)$$

Where, HUD(i,j) is the Euclidean distance between i & j. $\in_v$= 0.0001is the small constant used to threshold the negative velocities.

- Update soil on the path(i,j) and carried by IWD using eq. 5.8 & 5.9 respectively.

$$soil(i,j) = (1-\rho_n).soil(i,j) - \rho_{n.}loadsoil(i,j) \quad (3.8)$$

$$soil_{IWD} = soil_{IWD} + loadsoil(i,j) \quad (3.9)$$

where, ρ_n is a small constant.
Step-4: If dimensions are not completed go to step 3(I).
Step-5: Apply local search: 2-opt + swap
Step-6: Cost

$$cost_{IWD} = \sum_{i=1}^{n=1} dist(x(i), x(i+1)) \quad (3.10)$$

Step-7: Soil update

$$soil(i,j) = (1+\rho_{IWD}).soil(i,j) - \rho_{IWD}.\left(\frac{soil(iwd)}{nodes}\right) \quad (3.11)$$

Step-8: Update memory with best so far solution value
Step-9: Reset soil after fixed iterations, which lowers the soil content on the paths.
Step-10: If maximum iterations are completed then stop else go to step 2.

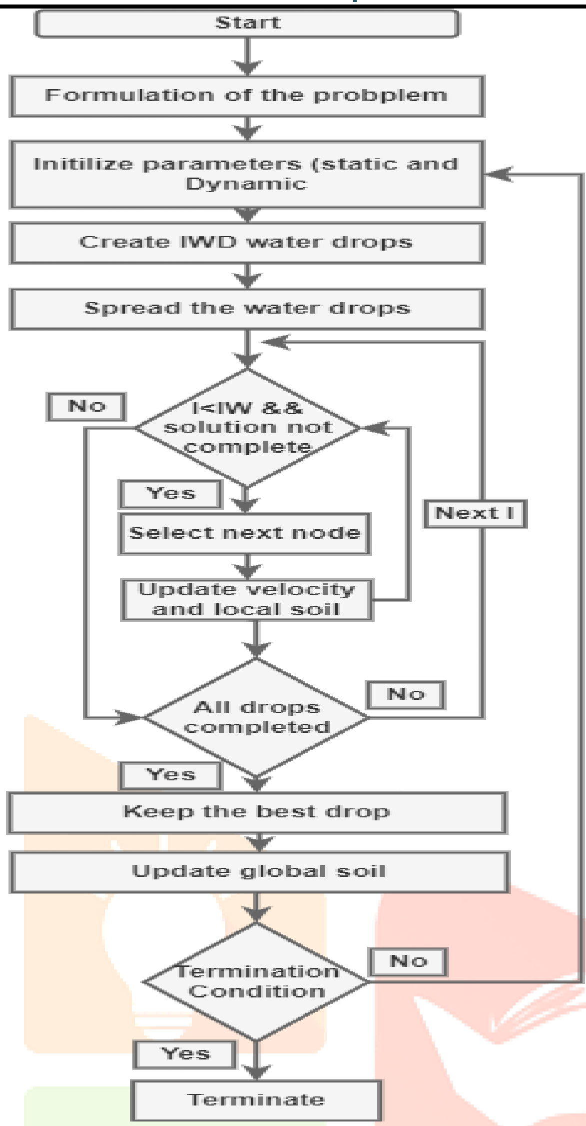

Fig. 3.1 Flow Chart of the proposed Algorithm

IV. RESULTS AND DISCUSSION

The experiment was implemented in the MATLAB Scripting environment, on Intel(R) Core(TM) i5 CPU M 460 @ 2.53GHz (manufacturer: Lenovo). The IWD proposed algorithms parameters are illustrated in table 1.

Table 4.1: Parameters for Proposed Algorithm

Parameter	Value
Iteration	60
a_v	1000
b_v	0.01
c_v	1
a_s	1000
b_s	0.01
c_s	1
Parallel solution to find	3
No. of Drops	4

Figure 4.1 and 4.3 displayed the routes selected while analyzing the algorithm and selecting the best possible optimal route for the problem P04 and P05 respectively. Figures 4 and 6 display the route cost divergence graph on both the problems P04 and P05 respectively.

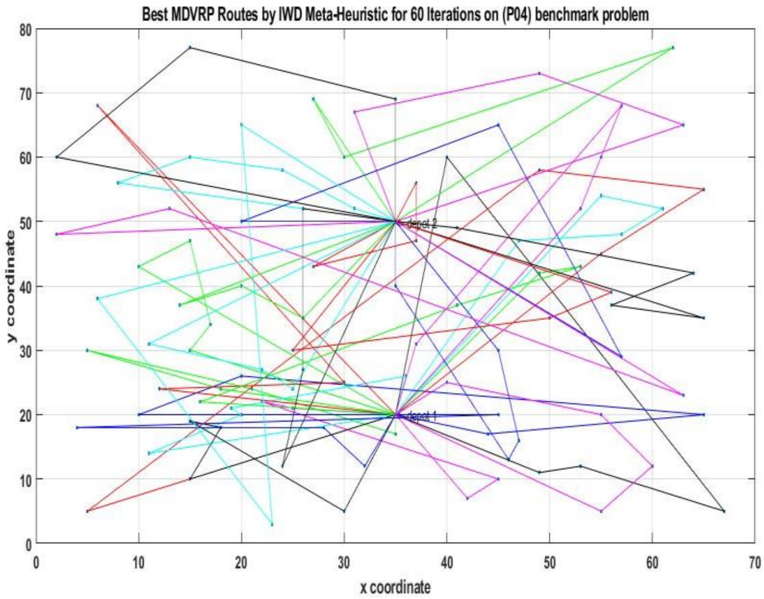

Fig. 4.1 Route map generated for benchmark problem (P04)

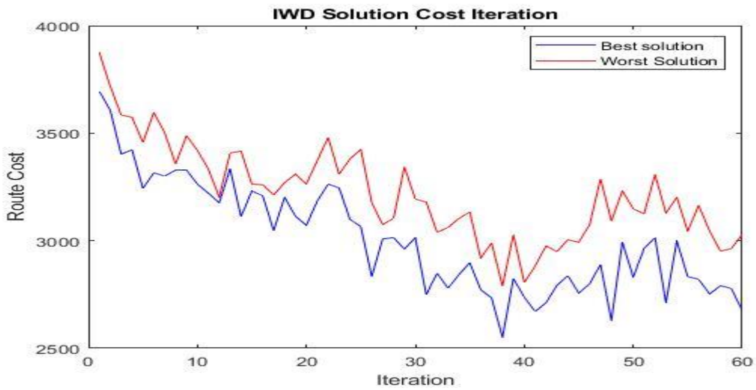

Fig. 4.2 Route cost convergence graph for benchmark P04 problem

Figure 4.2 and figure 4.4, these two figures demonstrate that, despite the higher computational cost incurred by the e-IWD, both the theoretical and e-IWD performed well in terms of solution quality and convergence speed when compared to the theoretical. This provides more evidence for the major contributions of the various methodologies used to implement the suggested algorithms. Several of these primary tactics include the use of an efficient cost function, approximation of SA's probability of acceptance criteria, and the use of SA as an alternative local search approach to enhance the convergence speed of the suggested enhanced algorithms.

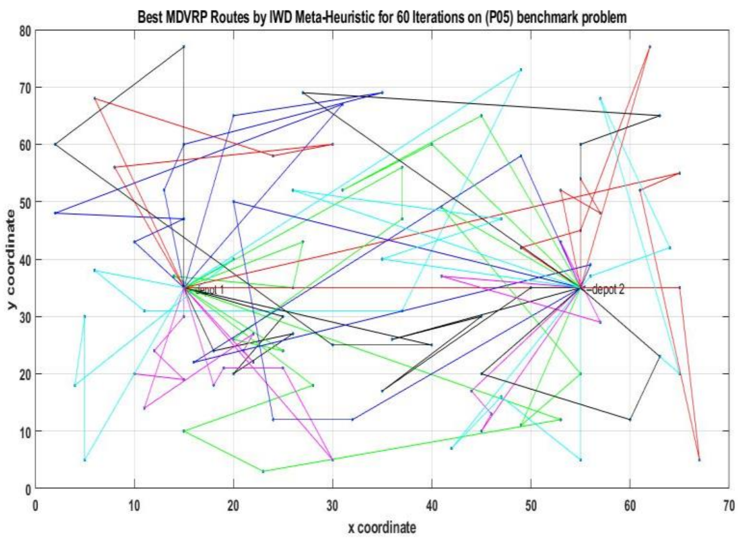

Fig. 4.3 Route map generated for benchmark problem (P05)

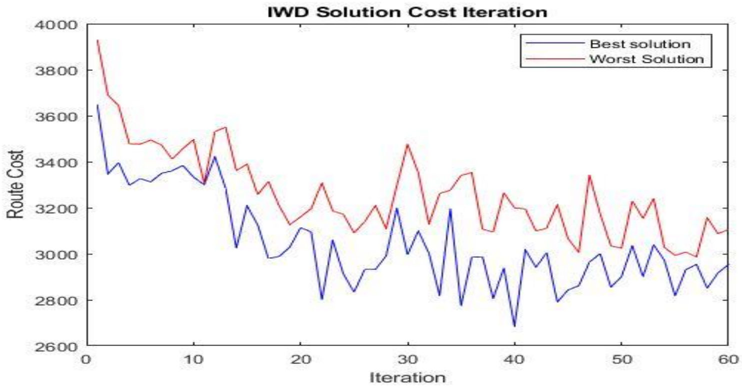

Fig. 4.4 Route cost convergence graph for benchmark P05 problem

IV. CONCLUSIONS

A set of two separate Cordeau benchmark examples, notably p04 and p05 in the MATLAB R2019b environment, are used to assess the efficacy of the suggested strategies. Customers were originally assigned to their appropriate depots based on Euclidean distance after the issue occurrences were categorized. The Clarke and Wright saving approach assigns clients of the same depot to various routes in the routing phase, and each route is sequenced in the scheduling phase. Enhanced IWD is used to optimize the planned routes. The suggested heuristic algorithms' simulation results were compared in terms of depot route length and best route. The performance of IWD was remarkable in the testing, and the computational time demonstrated that improved e-IWD is substantially quicker for addressing the multi-depot vehicle routing issue. In the future, attempts will be made to place more realistic limitations on the issue structure, and the suggested technique will be applied to large-scale real-time situations

REFERENCES

[1] Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B* **1996**, *26*, 29–41.

[2] Demirel, T.; Yilmaz, S. A New Solution Approach To Multi-Depot Vehicle Routing Problem With Ant Colony Optimization. *J. Mult. Valued Log. Soft Comput.* **2012**, *18*, 421–439.

[3] Yu, B.; Yang, Z.Z.; Xie, Y.X. A parallel improved ant colony optimization for multi-depot vehicle routing problem. *J. Oper. Res. Soc.* **2011**, *62*, 183–188.

[4] Stodola, P. Using Metaheuristics on the Multi-Depot Vehicle Routing Problem with Modified Optimization Criterion. *Algorithms* **2018**, *11*, 74.

[5] Li, Y.; Soleimani, H.; Zohal, M. An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives. *J. Clean. Prod.* **2019**, *227*, 1161–1172.

[6] Yu, B.; Jin, P.H.; Yang, Z.Z. Two-stage heuristic algorithm for multi-depot vehicle routing problem with time windows. *Syst. Eng. Theory Pract.* **2012**, *32*, 1793–1800.

[7] Jabir, E.; Panicker, V.V.; Sridharan, R. Design and development of a hybrid ant colony-variable neighborhood search algorithm for a multi-depot green vehicle routing problem. *Transp. Res. Part D* **2017**, *57*, 422–457.

[8] Ho, W.; Ho, G.T.S.; Ji, P.; Lau, H.C. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Eng. Appl. Artif. Intell.* **2008**, *21*, 548–557.

[9] Dantzig, G.B.; Ramser, J.H. The Truck Dispatching Problem. *Manag. Sci.* **1959**, *6*, 80–91.

[10] Clarke, G.; Wright, J.W. Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* **1964**, *12*, 568–581.

[11] Näher, S. The travelling salesman problem. *Oper. Res.* **1985**, *22*, 1044–1056.

[12] Karakatič, S.; Podgorelec, V. A survey of genetic algorithms for solving multi depot vehicle routing problem. *Appl. Soft Comput. J.* **2015**, *27*, 519–532.

[13] Juan, A.A.; Pascual, I.; Guimarans, D.; Barrios, B. Combining Biased Randomization with Iterated Local Search for Solving the Multi-Depot Vehicle Routing Problem. *Int. Trans. Oper. Res.* **2014**, *22*, 647–667.

[14] Li, J.; Pardalos, P.M.; Sun, H.; Pei, J.; Zhang, Y. Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups. *Expert Syst. Appl.* **2015**, *42*, 3551–3561.

[15] Ahmadi, M.; Seifi, A.; Tootooni, B. A humanitarian logistics model for disaster relief operation considering network failure and standard relief time: A case study on San Francisco district. *Transp. Res. Part E Logist. Transp. Rev.* **2015**, *75*, 145–163.

[16] Alinaghian, M.; Shokouhi, N. Multi-Depot Multi-Compartment Vehicle Routing Problem, Solved By A Hybrid Adaptive Large Neighborhood Search. *Omega* **2018**, *76*, 85–99.

[17] Vidal, T.; Crainic, T.G.; Gendreau, M.; Prins, C. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Comput. Oper. Res.* **2013**, *40*, 475–489.

[18] Benoit Crevier, Jean-François Cordeau, Gilbert Laporte. The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*. **2007**, Vol.176.

[19] Jairo R. Montoya-Torres, Julián López Franco, Santiago Nieto Isaza, Heriberto Felizzola Jiménez, Nilson Herazo-Padilla.. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering.* **2015**, vol.79, pp.115-129.

[20] M.Filipec, D.Skrlec, S.Krajcar. October. Darwin meets computers: new approach to multiple depot capacitated vehicle routing problem. IEEE International Conference on Systems, Man, and Cybernetics. *Computational Cybernetics and Simulation.* **1997** Vol.1, pp. 421 - 426.

[21] Nikhil, Amandeep Sharma, Navdeep Minhas, Arshdeep Singh. A Modified Winding Function Theory and Firefly Algorithm based Optimization for Diagnosis of Eccentricity in Induction Motors. *International Journal of Innovative Technology and Exploring Engineering.* Vol.8(9), **2019**, pp:2869-2876.

[22] Hosseini, H., Optimization with the NatureInspired Intelligent Water Drops Algorithm. In: Evolutionary Computation, Wellington P.D.S., (Ed.), Austria: I-Tech, Vienna, **2009**, pp: 297-320

[23] Hosseini, H.,Improving K-means clustering algorithm with the intelligent water drops algorithm. Int. J. Data Min. Model. Manage. **2013**, pp: 608.

[24] Harish, Y., B. Venkateshwarlu and C. Jada, Intelligent water drops algorithm for multimodal spaces. Proceedings of the International Conference on Swarm, Evolutionary and Memetic Computing, (EMC' 14), **2014**, Springer, Cham

[25] Abdulelah, G., S.A. Saif and Z. Fayed, Intelligent Water Drops (IWD) algorithm for COQUAMO optimization. Proceedings of the World Congress on Engineering and Computer Science, (ECS' 15), **2015**, San Francisco, USA.

[26] Hosseini, H.,. Problem solving by intelligent water Drops. Proceedings of the IEEE Congress on Evolutionary Computation, **2007**, Sept. 25-28, IEEE Xplore Press, Singapore.

[27] Sharma R., and S. Saini, Heuristics and Meta-Heuristics Based Multiple Depot Vehicle Routing Problem: A Review. International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE, **2020**, pp: 683-689.

[28] Samsuddin S., M. S. Othman, and L. M. Yusuf, Utilizing Ant Colony Optimization and Intelligent Water Drop for Solving Multi Depot Vehicle Routing Problem. IOP Conference Series: Materials Science and Engineering. Vol. 864(1). IOP Publishing, **2020**.

[29] Kumar S., D. R. Parhi, K. K. Pandey, and M. K. Muni, Hybrid IWD-GA: an approach for path optimization and control of multiple mobile robots in obscure static and dynamic environments. Robotica, **2021**, pp: 1-28.

[30] Aldeeb B. A., M. A. Al-Betar, N. M. Norwawi, K. A. Alissa, M. K. Alsmadi, A. A. Hazaymeh, and M. Alzaqebah, Hybrid intelligent water Drops algorithm for examination timetabling problem. Journal of King Saud University-Computer and Information Sciences, **2021**.