Intelligent Water Drops a new optimization algorithm for solving the Vehicle Routing Problem

Iman Kamkar[1] Mohammad-R. Akbarzadeh-T[1] MahdiYaghoobi[1]

[1]Department of Artificial Intelligence, Islamic Azad University, Mashhad Branch, Iran
Emails: Kamkar.iman@gmail.com, Akbarzadeh@ieee.org, Yaghobi@mshdiau.ac.ir

***Abstract*- The Vehicle Routing Problem (VRP) is an NP-hard combinatorial optimization problem, seeking to serve a number of customers with a fleet of available vehicles. VRP is an important optimization problem in the field of transportation, distribution and logistics. To date, several exact and approximate approaches have been proposed to solve VRP. Here, we apply a population based algorithm to VRP by imitating the natural flow of water drops. The "Intelligent Water Drops" or IWD algorithm solves the VRP by modeling how water drops collectively modify their environment by picking up dirt from river bottoms during moving downhill and leaving sediments (such as on beaches) when slowing down. The computational results for fourteen benchmark VRP problems are reported and compared to several other meta-heuristic approaches.**

I. INTRODUCTION

Finding efficient vehicle routes is an important logistics problem which has been studied for several decades. When a firm is able to reduce the length of its delivery routes or is able to decrease its number of vehicles, it is able to provide better service to its customers, operate in a more efficient manner and possibly increase its market share. A typical vehicle routing problem includes simultaneously determining the routes for several vehicles from a central supply depot to a number of customers and returning to the depot without exceeding the capacity constraints of each vehicle. This problem is of economic importance to businesses because of the time and costs associated with providing a fleet of delivery vehicles to transport products to a set of geographically dispersed customers. Additionally, such problems are also significant in the public sector where vehicle routes must be determined for bus systems, postal carriers, and other public service vehicles.

In each of the above instances, the problem typically involves finding the minimum cost of the combined routes for a number of vehicles in order to facilitate delivery from a supply location to a number of customer locations. Since cost is closely associated with distance, a company might attempt to find the minimum distance traveled by a number of vehicles in order to satisfy its customer demand. In doing so, the firm attempts to minimize costs while increasing or at least maintaining an expected level of customer service. The process of selecting vehicle routes allows the selection of any combination of customers in determining the delivery route for each vehicle. Therefore, the vehicle routing problem is a combinatorial optimization problem where the number of feasible solutions for the problem increases exponentially with the number of customers to be served. In addition, the vehicle routing problem is closely related to the traveling salesman problem where an out and back tour from a central location is determined for each vehicle. Since there is no known polynomial algorithm that will find the optimal solution in every instance, the vehicle routing problem is considered NP-hard. For such problems, the use of heuristics is considered a reasonable approach in finding solutions.

Heuristic algorithms such as simulated annealing (SA) [1,2,3,4], genetic algorithms (GAs) [5,6], tabu search (TS) [7,8] and ant colony optimization [9,10,11,12,13] are widely used for solving the VRP.

Recently, the new meta-heuristic algorithm "Intelligent Water Drops," has been introduced in the literature and used for solving the traveling salesman problem (TSP) and multiple knapsack problem [14,15].This paper tries to solve the VRP using an IWD-based algorithm. The IWD algorithm is a population-based optimization algorithm that uses the constructive approach to find the optimal solution(s) of a given problem. Its ideas are based on the water drops that flow in nature such that each water drop constructs a solution by traversing in the search space of the problem and modifying its environment.

II. VEHICLE ROUTING PROBLEM

The vehicle routing problem has been an important problem in the field of distribution and logistics since at least the early 1960s [16]. It is described as finding the minimum distance or cost of the combined routes of a number of vehicles m that must service a number of customers n. Mathematically, this system is described as a weighted graph $G=(V, A, d)$ where the vertices are represented by $V=\{v_0,v_1,..,v_n\}$, and the arcs are represented by $A=\{(v_i, v_j): i\neq j\}$. A central depot where each vehicle starts its route is located at v_0 and each of the other vertices represents the n customers. The distances associated with each arc are represented by the variable d_{ij} which is measured using Euclidean computations. Each customer is assigned a non-negative demand q_i, and each vehicle is given a capacity constraint, Q. The problem is solved under the following constraints.

- Each customer is visited only once by a single vehicle.

978-1-4244-6588-0/10/$25.00 ©2010 IEEE

- Each vehicle must start and end its route at the depot, v_0.
- Total demand serviced by each vehicle can't exceed Q.

III. INTELLIGENT WATER DROPS

The IWD [14] have been designed to imitate the prominent properties of the natural water drops that flow in the beds of rivers. Each IWD is assumed to have an amount of the soil it carries, soil(IWD), and its current velocity, velocity(IWD).

The environment in which IWDs are moving is assumed to be discrete. This environment may be considered to be composed of N_c nodes and each IWD needs to move from one node to another. Every two nodes are linked by an arc which holds an amount of soil. Based on the activities of the IWDs flowing in the environment, the soil of each arc may be increased or decreased.

Consider an IWD is in the node *i* and wants to move to the next node *j*. The amount of the soil on the arc between these two nodes, represented by *soil(i, j),* is used for updating the velocity $vel^{IWD}(t)$ of the IWD by:

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

Where $vel^{IWD}(t+1)$ represents the updated velocity of the IWD at the next node *j*. Moreover, a_v, b_v, and c_v are some constant velocity parameters that are set for the given problem.

Consider that a local heuristic function HUD(.,.) has been defined for a given problem to measure the undesirability of an IWD to move from one node to another .The time taken for an IWD having the velocity $vel^{IWD}(t+1)$ to move from the current node *i* to its next node *j*, denoted by time$(i, j; vel^{IWD}(t+1))$, is calculated by:

$$time(i,j;vel^{IWD}) = \frac{HUD(i,j)}{\max(\varepsilon, vel^{IWD})} \quad (2)$$

The constant parameter ε is a small positive value. Here, ε = 0.001. The function HUD(i, j) denotes the heuristic undesirability of moving from node i to node j.

For the VRP, the form of the HUD(i, j) denoted by HUD_{VRP}(i, j) has been suggested as follows:

$$HUD(i,j) = HUD_{VRP}(i,j) = \|c(i) - c(j)\| \quad (3)$$

Where c(k) represents the two dimensional positional vector for the city k. The function || . || calculates the Euclidean norm. As a result, when two nodes (cities) *i* and *j* are near to each other, the heuristic undesirability measure HUD*(i, j)* becomes small which reduces the time taken for the IWD to pass from city *i* to city *j*.

As an IWD moves from the current node *i* to its next node *j,* it removes an amount of soil from the path (arc) joining the two nodes. The amount of the soil being removed depends on the velocity of the moving IWD. For the VRP, the amount of the soil taken from the path is related with the inverse of the time that the IWD needs to pass the arc or path between the two nodes. So, a fast IWD removes more soil from the path it flows on than a slower IWD. This mechanism is an imitation of what happens in the natural rivers. Fast rivers can make their beds deeper because they remove more soil from their beds in a shorter time while slow flowing rivers lack such strong soil movements. Moreover, even in a single river, parts of the river that water drops flow faster often has deeper beds than the slower parts.

For the VRP, the amount of the soil that the IWD removes from its current path from node *i* to node *j* is calculated by:

$$\Delta Soil(i,j) = \frac{a_s}{b_s + c_s.time(i,j;vel^{IWD})} \quad (4)$$

where Δsoil*(i, j)* is the soil which the IWD with velocity vel^{IWD} removes from the path between node *i* and *j*. The a_s, b_s, and c_s are constant velocity parameters that their values depend on the given problem. The value time(*i*, *j*; vel^{IWD}) was defined in equation (2) and represents the time taken for the IWD to flow from *i* to *j*.

After an IWD moves from node *i* to node *j*, the *soil(i, j)* on the path between the two nodes is reduced by:

$$soil(i,j) = \rho_0.soil(i,j) - \rho_n.\Delta soil(i,j) \quad (5)$$

Where ρ_0 and ρ_n are positive numbers that should be chosen between zero and one. In the original algorithm for the TSP [14], $\rho_0 = 1 - \rho_n$.

The IWD that has moved from node i to j, increases the soil $soil^{IWD}$ it carries by:

$$soil^{IWD} = soil^{IWD} + \Delta soil(i,j) \quad (6)$$

Where *Δsoil(i,j)* is obtained from equation (4). Therefore, the movement of an IWD between two nodes reduces the soil on the path between the two nodes and increases the soil of the moving IWD.

One important mechanism that each IWD must contain is to how to select its next node. An IWD prefers a path that contains less amount of soil rather than the other paths. This preference is implemented by assigning a probability to each path from the current node to all valid nodes which do not violate constraints of the given problem. Let an IWD be at the node i, then the probability $p_i^{IWD}(j)$ of going from node *i* to node *j* is calculated by:

$$p_i^{IWD}(j) = \frac{f(soil(i,j))}{\sum_{k \notin vc(IWD)} f(soil(i,k))} \quad (7)$$

Such that $f(soil(i,j))$, computes the inverse of the soil between node *i* and *j*. specifically:

$$f(soil(i,j)) = \frac{1}{\varepsilon_s + g(soil(i,j))} \quad (8)$$

The constant parameter ε_s is a small positive number to prevent a possible division by zero in the function f(.). It is suggested to use ε_s= 0.01. *g(soil(i, j))* is used to shift the soil*(i, j)* on the path joining nodes *i* and *j* toward positive values and is computed by:

$$g(soil(i,j)) = \begin{cases} soil(i,j) & if \min_{l \notin vc(IWD)} (soil(i,l)) \geq 0 \\ soil(i,j) - \min_{l \notin vc(IWD)} (soil(i,l)) & else \end{cases} \quad (9)$$

The function min(.) returns the minimum value of its arguments. The set *vc*(*IWD*) denotes the nodes that the IWD should not visit to keep satisfied the constraints of the problem. Every IWD that has been created in the algorithm moves from its initial node to next nodes till it completes its solution. For the given problem, an objective or quality function is needed to measure the fitness of solutions. Consider the quality function of a problem to be denoted by *q(.)*. Then, the quality of a solution T^{IWD} found by the IWD is given by $q(T^{IWD})$. One iteration of the IWD algorithm is said to be complete when all IWDs have constructed their solutions. At the end of each iteration, the best solution T^{IB} of the iteration found by the IWDs is obtained by:

$$T^{IB} = \arg \max_{\forall T^{IWD}} q(T^{IWD}) \quad (10)$$

Therefore, the iteration-best solution T^{IB} is the solution that has the highest quality over all solutions T^{IWD}.
Based on the quality of the iteration-best solution, $q(T^{IB})$, only the paths of the solution T^{IB} are updated. This soil updating should include the amount of quality of the solution. Specifically:

$$soil(i,j) = \rho_s . soil(i,j) + \rho_{IWD} . k(N_s) . soil_{IB}^{IWD} \quad \forall (i,j) \in T^{IB} \quad (11)$$

Where $soil_{IB}^{IWD}$ represents the soil of the iteration-best IWD. The best-iteration IWD is the IWD that has constructed the best-iteration solution T^{IB}. $k(N_c)$ denotes a positive coefficient which is dependent on the number of nodes *Nc*. Here, $k(N_c)=1/(N_c-1)$ is used. ρ_s should be a constant positive value whereas the constant parameter ρ_{IWD} should be a negative value. The first term on the right-hand side of equation (11) represents the amount of the soil that remains from the previous iteration. In contrast, the second term on the right-hand side of equation (11) reflects the quality of the current solution, obtained by the IWD. Therefore, in equation (11), a proportion of the soil gathered by the IWD is reduced from the total soil soil*(i, j)* of the path between node *i* and *j*.
This way, the best-iteration solutions are gradually reinforced and they lead the IWDs to search near the good solutions in the hope of finding the globally optimal solution.
At the end of each iteration of the algorithm, the total best solution T^{TB} is updated by the current iteration-best solution T^{IB} as follows:

$$T^{TB} = \begin{cases} T^{IB} & if \quad q(T^{TB}) \geq q(T^{IB}) \\ T^{TB} & otherwise \end{cases} \quad (12)$$

By doing this, it is guaranteed that T^{TB} holds the best solution obtained so far by the IWD algorithm.

IV. The Proposed IWD Algorithm for the VRP

We resume here the main characteristics of our IWD algorithm for VRP.

Initialization of parameters: in the beginning of IWD algorithm the following parameters must be initialized: number of water drops N_{IWD}, the number of cities N_c .The number of cities is depend to the problem at hand, and here we the number of water drops equal to the number of vehicles. For velocity updating we use parameters a_v=1000, b_v=0.01 and c_v=1. For soil updating we use parameters *as*=1000, *bs*=0.01 and *cs*=1.Moreover the initial soil of each link is denoted by the constant *Initsoil* such that the soil of the link between every two cities is set by *soil(i,j)= Initsoil.* The initial velocity of IWDs is denoted by the constant *Initvel.* Both parameters of *Initsoil* and *Initvel* are user selected. In this paper, we choose *Initsoil=1000* and *Initvel=100.*

Routes building: at each iteration of IWD algorithm each IWD builds a solution for the VRP, moving to next city according to selection rule based on a combination of the amount of soil at each arc and length of it (see (2) below). For every IWD, a visited node list V_c(IWD) is considered to include nodes just visited.
Routes can be determined in two versions:

sequential: each IWD start its solution determining the route for the first vehicle untill its capacity is complete. Then it continues with others vehicles till complete the solution .
Parallel : each IWD designs the route for all vehicles at the same time. At each iteration of the algorithm only one city is chosen, according to selection rule. Then best tour is extended.

in our experiments we have used parallel approach in order to build solutions.

Selection rule:

Next city is chosen according to probability $p_i^{IWD}(j)$ according to:

$$p_i^{IWD}(j)=\frac{f(soil(i,j))}{\sum_{k\notin vc(IWD)} f(soil(i,k))} \qquad (13)$$

Where *f(soil(i,j))* is computed from equation (8).

Soil and velocity updating:

local updating: for each IWD moving from city *i* to next city *j*, its velocity $vel^{IWD}(t)$ must be updated according to :

$$vel^{IWD}(t+1)=vel^{IWD}(t)+\frac{a_v}{b_v+c_v.soil(i,j)} \qquad (14)$$

And the soil of the path that traversed byl the IWD, *soil(i,j)* and the soil that IWD carries, $soil^{IWD}$ must be updated according to:

$$soil(i,j)=\rho_0.soil(i,j)-\rho_n.\Delta soil(i,j) \qquad (15)$$

$$soil^{IWD}=soil^{IWD}+\Delta soil(i,j) \qquad (16)$$

that *Δsoil(i,j)* is the amount of soil that current water drop loads from its current path between to nodes i and j and can be obtained from equation (4).

global updating: after each iteration is completed , the soils of the paths that exist in the current best –solution T^{TB} is updated using equation (12), by setting, $\rho_s=1-\rho_{IWD}$.

$$soil(i,j)-\rho_s.soil(i,j)+\rho_{IWD}.k(N_c).soil_{IB}^{IWD} \quad \forall(i,j)\in T^{IB} \qquad (17)$$

Stopping rules: IWD procedure stops when there is not improvement on the solution after several iterations or when n-max number of iterations is reached.

V. EXPERIMENTAL RESULTS

In this section, we present computational results of our proposed algorithm, which was coded in Matlab 7.1 and executed on a pc computer with a Pentium processor running at 1GHZ. To evaluate validity of our proposed algorithm for the vehicle routing problem, the performance of our algorithms was tested on a set of 14 benchmark instances designed by Christofides et al. and can be downloaded from the OR Library at the website with URL:http://mscmga.ms.ic.ac.uk/jeb/orlib/vrpinfo.html.

The information of the 14 problems is shown in columns 2–4 in Table 1, which consists of the problem size *n*, the vehicle capacity *Q* and the well-known published results [17] and [18].We compare IWD with a number of the better methods available for the VRP, and the results of some problems are described in columns 5-8 of Table 1, where SA refers to Simulated Annealing by Osman[3], TS to Tabu Search by Osman[3], IACO to Improved Ant Colony by Yu.Bin [12] and IWD is the algorithm we proposed. The IWD algorithm has shown to be competitive with the best existing methods in terms of solution quality.

TABLE 1
Comparison of heuristics for the vehicle routing problem

Prob.	n	Q	Best	SA	TS	IACO	IWD
C1	50	160	524.61	528	524	524.61	524.61
C2	75	140	835.26	838	844	835.26	836.76
C3	0	200	826.14	829	835	830.00	829.34
C4	150	200	1028.42	1058	1052	1028.42	1054.26
C5	199	200	1291.45	1376	1354	1305.5	1326.12
C6	50	160	555.43	555	555	555.43	555.43
C7	75	140	909.68	909	913	909.68	914.53
C8	100	200	865.94	866	866	865.94	866.14
C9	150	200	1162.55	1164	1188	1162.55	1163.76
C10	199	200	1395.85	1418	1422	1395.85	1408.47
C11	120	200	1042.11	1176	1042	1042.11	1043.35
C12	100	200	819.56	826	819	819.56	819.75
C13	120	200	1541.14	1545	1547	1545.93	1544.65
C14	100	200	866.37	890	866	866.37	868.92

Figure 1 shows the relation of the length of minimum tour versus its iteration by the IWD algorithm for 51 city problem with 5 vehicles (C1 problem). Almost all part of the curve is descending, except for the iteration 30 and 35 which shows a slight increase in the length of minimum tour in contrast to their previous iteration.

However, after these short ascending, the curve follows its general downward movement. This property shows that the IWD is able to go upward to get rid of some local optimum in order to get to better optimums.

In order to have a correct evaluation and comparison of the quality of algorithms the computing times must be taken into account. As different researchers have used different kind of computers, correct evaluating and comparison of computing times is difficult. A very rough measure of computers' performance can be obtained using Dongarra's tables [19] where the number (in millions) of floating point operations per second (Mflop/seconds) executed by each computer was used, when solving standard linear equations, with LINPACK program. Regarding computational times, Osman used a VAX 8600(about 2.48 MFlop/s), Yu Bin used Pentium 1 GHz (about75 MFlop/s). In this research Pentium 1GHz running IWD has an estimated power of 75 MFlop/s. Table 2 shows the

TABLE 2. Computation times of several meta-heuristic approaches

Prob.	TS run times	TS scaled	SA run times	SA scaled	IACO	IWD
C1	60	2	6	0.2	2	1.5
C2	48	1.5	3564	117.8	11	2
C3	894	29.5	6174	204.1	30	12
C4	1764	58.3	4296	142.1	211	25
C5	1704	56.3	1374	45.4	677	36
C6	60	2	696	23	24	2
C7	744	24.6	312	10.4	20	8
C8	1962	64.8	366	12.1	57	15
C9	2472	81.8	59016	1951.5	307	29
C10	4026	133	2418	79.9	840	58
C11	780	25.8	264	8.7	61	16
C12	342	11.2	48	1.5	31	12
C13	1578	52.1	4572	151.1	127	28
C14	582	19.3	300	9.9	43	8

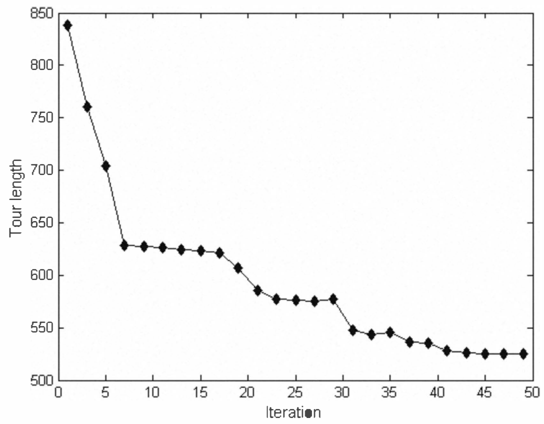

Figure 1. The length of minimum tour versus the iteration found by the algorithm for the 51-city problem

origin computation times and the scaled Computation times, which use Pentium 1 GHz as the baseline, of different approaches. The performance of IWD is competitive when compared with other meta-heuristic approaches, such as SA,TS and ACO.As it can be seen from Table 2 our proposed algorithm can find near optimal solution in a better computation time than the other algorithms.

VI. CONCLUSION

IWD, "Intelligent Water Drops" is a population based algorithm that imitates the flow of water in river banks and beaches. In this paper, we propose the use of this algorithm for solving vehicle routing problem, likening loads/tasks to sediments that are left behind or picked up by the water drops in their natural motion in a river bed. The IWD algorithm is experimented on 14 bench mark VRP problems. The computational results of these problems reveal that the proposed IWD converges fast to the optimum solutions and finds good and promising results. Further researches on additional modifications of the IWD to extensions of the vehicle routing problem with time windows or with more depots are of interest.

REFERENCES

[1] Chiang, W.C., Russell, R., 1996. Simulated annealing meta-heuristics for the vehicle routing problem with time windows. Annals of Operations Research 93, 3–27.

[2] Koulamas, C., Antony, S., Jaen, R., 1994. A survey of simulated annealing applications to operations research problems. Omega 22 (1), 41–56.

[3] Osman, I.H., 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research 41, 421–451.

[4] Tavakkoli-Moghaddam, R., Safaei, N., Gholipour, Y., 2006. A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length. Applied Mathematics and Computation 176, 445–454.

[5] Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research 31, 1985–2002.

[6] Geonwook Jeon, Herman R. Leep and Jae Young Shim, 2008," A vehicle routing problem solved by using a hybrid genetic algorithm", Journal of Computers and Industrial Engineering, Elsevier.

[7] Renaud, J., Laporte, G., Boctor, F.F., 1996. A tabu search heuristic for the multi-depot vehicle routing problem. Computers & Operations Research 23 (3), 229–235.

[8] Brandao, J., Mercer, A., 1997. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. European Journal of Operational Research 100,180–191.

[9] Doerner, K.F., Hartl, R.F., Kiechle, G., Lucka, M., Reimann, M., 2004. Parallel ant systems for the capacitated vehicle routing problem. In: Evolutionary Computation in Combinatorial Optimization: 4th European Conference,EvoCOP 2004, LNCS 3004, pp. 72–83.

[10] Reimann, M., Stummer, M., Doerner, K., 2002. A savings based ant system for the vehicle routing problem. In: Langdon, W.B. et al. (Eds.), GECCO 2002:Proceedings of the Genetic and Evolutionary Computation Conference. Morgan Kaufmann, San Francisco.

[11] Peng, W., Tong, R.F., Tang, M., Dong, J.X., 2005. Ant colony search algorithms for optimal packing problem. ICNC 2005, LNCS 3611, pp. 1229–1238.

[12] Yu Bin ,Yang Zhong-Zhen , Yao Baozhen ,2008 .An improved ant colony optimization routing problem, European Journal of Operational Research, Elsevier.

[13] John E. Bell,Patrick R. McMullen,2004. Ant colony optimization techniques for the vehicle routing problem, Journal of Advanced Engineering Informatics, Elsevier.

[14] Hamed Shahhosseini, 2007. Problem solving by Intelligent Water Drops,IEEE 2007.

[15] Hamed Shahhosseini, 2008, "Intelligent Water Drops: A new optimization method for solving the multiple knapsack problem" ,International journal of Intelligent Computing and Cybernetics, Emerald.

[16] Clark G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. Oper Res 1964;12:568–81.

[17] Taillard, R.E., 1993. Parallel iterative search methods for vehicle routing problems.Networks 23, 661–673.

[18] Rochat, Y., Taillard, R.E., 1995. Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics 1, 147–167.

[19] Dongarra, J., 2001. Performance of various computer using standard linear equations software. Report CS-89-85, University of Tennessee.