Journal of Theoretical and Applied Information Technology
10th July 2015. Vol.77. No.1

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

A WATER FLOW-LIKE ALGORITHM FOR CAPACITATED VEHICLE ROUTING PROBLEM

[1] SUHAILA ZAINUDIN, [2] MOKHTAR MASSOUD KERWAD, [3] ZULAIHA ALI OTHMAN
[1,2,3] Centre for Artificial Intelligence Technology, Faculty of Information Science and Technology,
University Kebangsaan Malaysia, 43600 Bangi, Selangor Darul Ehsan, Malaysia.
Email: [1] suhaila.zainudin@ukm.edu.my, [2] kerwadd@yahoo.com, [3] zao@ukm.edu.my.

ABSTRACT

The goal of the capacitated vehicle routing problem (CVRP) is finding a useful vehicle route which is a fundamental issue of logistic management. CVRP consists of finding the optimal routes for distributing various items between customers and depot by a fleet of vehicles at a minimum traveling distance without exceeding the capacity of each vehicle. Among many proposed schemes, meta-heuristic algorithm is a well-known optimization method for solving non-deterministic polynomial-time problems. The population-based meta-heuristics has shown the ability to obtain an excellent solution in many domains but consumes time. This is due to the nature of the algorithms that have fixed a number of the solutions, so they suffer from high computation times to reach the solution. In this paper a new approximation algorithm named Water Flow-like Algorithm (WFA) is proposed to tackle CVRP. It is inspired by the natural behavior of water flowing from a higher to a lower level, which is self-adaptive and dynamic based on population sizes and parameter settings. The performance of the proposed algorithm is evaluated using 14 benchmark datasets. The result shown performance of the proposed algorithm is comparable with other recent enhanced algorithms. Therefore, it indicates that WFA is a potential alternative to solve the CVRP using other enhancement on the algorithm.

Keywords: *Combinatorial Optimization, Vehicle Routing Problem, Meta-heuristics, Dynamic solution, Water Flow-like Algorithm.*

1. INTRODUCTION

Capacitated Vehicle Routing Problem (CVRP) is a combinatorial optimization problem that has received considerable attention recently because finding an efficient vehicle is an important issue of logistic management [1], [2]. Hence, effective transportation management can increase competitiveness and reduce the environmental impact of organizations due to more optimal routes and shorter distances. CVRP was first designed by [3] and concerns the design of a set of least cost routes for a set of vehicles to service a number of customers with identified demands [4]. The problem combined two Non-deterministic Polynomial-time (NP) hard problems, the traveling salesman problem and the bin-packing problem (BPP), hence CVRP is also NP-hard [5].

As an NP-hard problem, a number of approaches were proposed to solve it, and they vary in terms of complexity, efficiency and their ability to solve the problem. In general, these techniques can be categorized into exact and approximate algorithms. Although exact algorithms such as dynamic programming [6], branch and bound [7], branch and cut [8], and branch and price [9] are suitable for small size instances, as problems become large and heavily constrained exact methods are no longer suitable to solve the problem and often fail to get an optimal solution owing to the computational time required [10]. On the other hand, approximate algorithms can obtain satisfactory solutions in competition time, but there is no guarantee to find global optimal solutions [11]. These algorithms can be classified as classical heuristics and meta-heuristics [12]. Some classical heuristics are saving heuristics of Clarke and Wright [13], and sweep algorithm has been proposed by [14]. Meta-heuristic combines basic heuristic strategies in higher level frameworks in order to explore search space more efficiently. Since these algorithms have good abilities to explore search space and are unquestionably efficient to get from a local optimum, they are

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

candidate algorithms for solving combinatorial optimization problems.

Based on the literature, several meta-heuristic algorithms have been utilized to solve CVRP. Meta-heuristics has been categorized into two classes, single solution and population solution. The single solutions are simulated annealing (SA) [15], tabu search (TS) [16], and greedy randomized adaptive search procedure (GRASP) [17]. These algorithms were designed as a single solution search. The population based solution has been designed as a multiple solutions search and their objective is to guide that search in state space to obtain a satisfactory solution; an example of population meta-heuristics genetic algorithm (GA) [18], [19], ant colony optimization (ACO) [20], [21], and ant colony system (ACS) [22], [10]. Single solution meta-heuristics uses a single solution to search the solution space step by step. Meanwhile, their searching may be inefficient due to a weakness in the solution exploration. Hence, these algorithms are more suitable for problems that have a smooth solution space rather than the problems with many local optima [23]. Population meta-heuristics deals with a set of solutions that can search the solutions space more efficiently and make the problem space exploration more powerful [24]. These algorithms are not rapid enough to conduct an efficient search solution due to the fixed number of solutions in its nature as a low number may increase the convergence of algorithm and reduce the solution exploitation, while a large number may cause unnecessary computation and useless searching because of a redundant search [25], [26]. Furthermore, it is not easy to determine a suitable population size during the optimization process as different problems require different parameter settings based on the problem size [27]. Nonetheless, finding good parameter values needs human expertise and time which are both expensive and rare [28].

Based on the aforementioned, it is evident that single or population approaches are not good enough to search the problem space. Hence, the new meta-heuristic designed by [23], called Water Flow-like Algorithm (WFA) is proposed. WFA is categorized as a population based meta-heuristic with a new concept of self-adaptive and dynamic solutions size during the problem process. Unlike other population sizes of meta-heuristics that have fixed numbers of solution sizes, the algorithm's assumption depends on two main points that affect the efficiency of algorithm optimization. Firstly, redundant search causes an increase in the computational cost during the optimization process to the algorithm. WFA avoids this problem by combining the solutions that target the same objective value. Secondly, it has adaptive ability and parameter tuning during the problem processing, unlike other meta-heuristic algorithms such as GA or ACO which have a fixed population size. The above two issues motivate us to use WFA for solving the CVRP problem.

In this paper, we adopt WFA and develop a heuristic algorithm (WFACVRP) for the CVRP problem. Based on our knowledge, this algorithm is applied for the first time to solve CVRP. This algorithm combines WFA using a random method for generating quick initial solutions for improvement later, and three different solution improving strategies (swap, move and 2-opt operator) for searching on the neighborhood solution.

This work aims to develop WFA to tackle CVRP. The main research question of this work is "Does the WFA is able to solve CVRP?". Moreover, the performance of WFA is studied with different stopping criteria (number of iteration). A second research question appear in this work is "Dose the number of iteration effect the performance of the algorithm?". The answer of these research questions will be found during the experiments.

This paper is organized as follows; Section 2 describes the problem, Section 3 discusses neighborhood operators, Section 4 details the proposed WFA for CVRP and Section 5 illustrates the computational results on test problems and the conclusion.

2. THE CAPACITATED VEHICLE ROUTING PROBLEM

CVRP is an important distribution management problem that can be used to model many real-world systems; such as school bus routing, refuse-collection truck routing, and postal service routing. This problem is important due to the social, economic and environmental issues in providing transportation and convenience to the population. Given a set of constraints, CVRP is used to design an optimal route for a fleet of vehicles to service a set of customers.

Journal of Theoretical and Applied Information Technology
10th July 2015. Vol.77. No.1

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

2.1 Problem Description

The CVRP is described as the graph theoretic problem: let $G = (V, E)$ be a complete and undirected graph where $V = \{0, \cdots, n\}$ is the vertex set and E is the edge set. Vertex set $V = \{0, 1, 2, \cdots, n\}$ corresponds to n customers, whereas vertex 0 corresponds to the depot. A fleet of k identical vehicles of capacity Q is based at the depot, each customer i has a non-negative demand qi. Its objective is to find the optimal routes for distributing various items between customers and depot by a fleet of vehicles at minimal traveling distances, without violating the capacity of each vehicle and some constraints associated with this problem. The mathematical model for CVRP can be written as follows [10]:

Variables:
Dis = total distance travelled by all vehicles.

$$x_{ijs} = \begin{cases} 1, & vehicle\ departs\ from\ i\ to\ j \\ 0, & otherwise \end{cases}$$

$$y_{is} = \begin{cases} 1, & customer\ \ is\ served\ by\ vehicle\ s \\ 0, & otherwise \end{cases}$$

Coefficients:
c_{ij} = cost from customer i to customer j
q_i = the demand of customers i (i = 1, 2, 3..., n)
n = total number of customers
k = total number of vehicles k
Q = capacity of vehicle s
s = the vehicle number (1, 2, 3 ..., k)

$$Minimize \quad dis = \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{s=1}^{k} c_{ij} x_{ijs} \qquad (1)$$

Subject to

$$\sum_{i=1}^{n} q_i\, y_{is} \le Q, \quad s = 1, \dots, k \qquad (2)$$

$$\sum_{i=0}^{n} x_{ijs} = y_{js}, j = 1,2, \dots, n; \;\; s = 1,2, \dots, k \qquad (3)$$

$$\sum_{j=0}^{n} x_{ijs} = y_{is}, i = 1,2, \dots, n; \;\; s = 1,2, \dots, k \qquad (4)$$

$$\sum_{s=1}^{k} y_{is} = \begin{cases} 1, & i = 1,2, \dots n \\ k, & i = 0 \end{cases} \qquad (5)$$

Eq. (1) is the objective function of the problem to minimize the traveling distance. Eq. (2) avoids exceeding capacity of each vehicle. Eq. (3) and Eq. (4) ensure that does not exceed the maximum number of vehicles. Eq. (5) guarantees that each customer is served exactly by one vehicle. Figure 1, show the CVRP solution as graph.

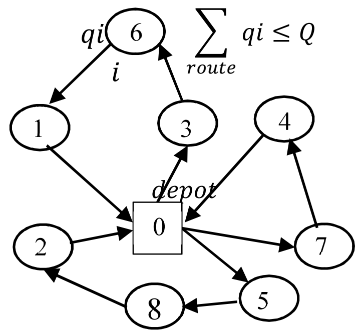


Figure 1: A solution to a CVRP as a graph

The solution is presented as a one-dimensional vector where the numbers between “1” to “8” represent customers and number “0” indicates the depot. The length of the solution represents the total number of customers. Each solution has a number of routes which can be counted based on the total number of “0” minus one. The example of solution representation shown in figure 2, has three routes. The first route serves two customers 7 and 4, the second serves three customers 5, 8 and 2 and the third route serves three customers 3, 6 and 1.

0	7	4	0	5	8	2	0	3	6	1	0

Figure 2. Solution representation

3. WATER FLOW-LIKE ALGORITHM

WFA was first proposed by [23] for solving BPP. WFA has been designed to mimic the natural behavior of water flowing from higher to lower levels that help in the process of searching for the optimal solution. The search solution of WFA is modeled as water flows; then, each flow is considered as a solution. Water moving to a lower position is regarded as a solution searching for the optimal, and the objective functions are modeled as the terrain traversed by the flows (flow altitude), and geographic train modeled as solution space. One water flow is deployed when the solution search starts. Thereafter, the flow splits into multiple sub-flows and occurs when rugged terrains are traversed. Conversely, a number of flows will merge into a single flow when they meet at the

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

same location. Therefore, the number of flows changes for searching optimal solutions. The WFA has been proven in its performance in different optimization problems, including the BPP [23], manufacturing cell fraction problem [25], TSK-type interval-valued fuzzy system (TIVFS) optimization [26], and traveling salesman problem (TSP) [29], [30].

The WFA put forward a new approach for computational optimization. It has features that are self-adaptive and has dynamic parameter settings and population sizes. It has an unfixed solution size, not like other meta-heuristic algorithms such as GA or ACO meaning that while solving the problem, the number of flows (solutions) are subject to increases or decreases according to diminution of the problem and the quality of the solution that will be found. [23], described the dynamic of the solutions size to mimic the natural behavior of water flows as split, move and merge.

The first version of WFA was applied to solve the BPP. The algorithm mainly depended on neighbor solution searches. The result was favorable in quality of the solution, executed time and based on the result; the authors summarized that WFA has the ability to solve complicated optimization problems. [25], applied a new improved model of WFA known as WFACF, to tackle the manufacturing cell fraction problem, using coefficient and machine assignment methods aiming to improve the algorithm in its initial solution; and in flow splitting and moving they used a neighbor search to improve the solution aimed at obtaining a satisfactory solution. [26], applied WFA to solve TSK-type interval-valued fuzzy system (TIVFS) optimization. The results showed a valuable performance of the algorithm. Other research has been done by [29], [30] WFA has shown its good performance in TSP, by using nearest neighbor heuristic to improve the initial solution, and 3-opt neighborhood and 2-opt neighborhood to get a near optimal solution. The results of these studies show that the WFA has good potential for solving different combinatorial optimization problems.

The differences between WFA for the CVRP (WFA-CVRP) and basic one [23] which solved bin packing problem are solution representation, splitting and moving operation, and precipitation operation. In this work, the solutions are represented in one diminution array as in figure 2. Whereas in the basic WFA the solutions are represented in two diminution array. In fact, the solution representation is a problem dependent [11]. Since splitting and moving operation is also problem dependent, it is designed by using three neighborhood operators which are move, swap, and 2-opt operators to find new location for the flow (neighbor). Whilst, the basic WFA used one step move for bin packing problem (refer to [23]). Moreover, in this work the precipitation operation used a solution shacking (swap) operator in order to modify all locations of the pour-down flows.

4. WFA FOR CAPACITATED VEHICLE ROUTING PROBLEM

In this paper, we developed WFA logic and designed heuristic algorithm for solving CVRP problems. Due to the features of this algorithm that are dynamic, there are multiple numbers of solutions. Figure 3, show the flow chart of the WFA-CVRP, the processes with the dark background shape contain the differences with basic WFA. The movements of water flow (location changes) are influenced by the gravitational force and the energy conservation law. At each iteration, water continually moves to lower altitudes, correlative with improvements in the solution search. Starting with one solution (flow) with an initial momentum, the WFA begins searching in the problem space. Afterward, when the flow encounters rugged terrains and the momentum of the flow has exceeded the splitting amount, the flow will split into sub-flows. A flow with higher momentum will generate more sub-flow streams than one with less momentum, otherwise the flow that has limited momentum yields to the landform and maintains one flow. Numbers of flows will merge into one flow if they are shared at the same location. To avoid redundant searches, WFA reduces the number of solutions to get the same objective value.

Water flows are subject to water evaporation in the atmosphere. By rainfall, this evaporation of water will return to the ground. WFA manually removes part of the water flow to mimic the water evaporation. Later, a precipitation operation is implemented in WFA to simulate natural rainfall to explore a wider area. Table 1 shows the analogy between real water flow and WFA.

Journal of Theoretical and Applied Information Technology
10th July 2015. Vol.77. No.1

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

Table 1. The analogy between real water flow and WFA algorithm.

Real water flow	WFA Algorithm	CVRP
Water flow	One solution	Trucks path
Flow move	Neighbor solutions	Move, swap
Flow altitude	Objective function	Solution distance
Geographic terrain	Solution space of a problem	Available paths

Depending on the above behaviors of WFA, the WFA consists of four base operations: 1- flow splitting and moving, 2-flow merging, 3- water evaporation and, 4- water precipitation.

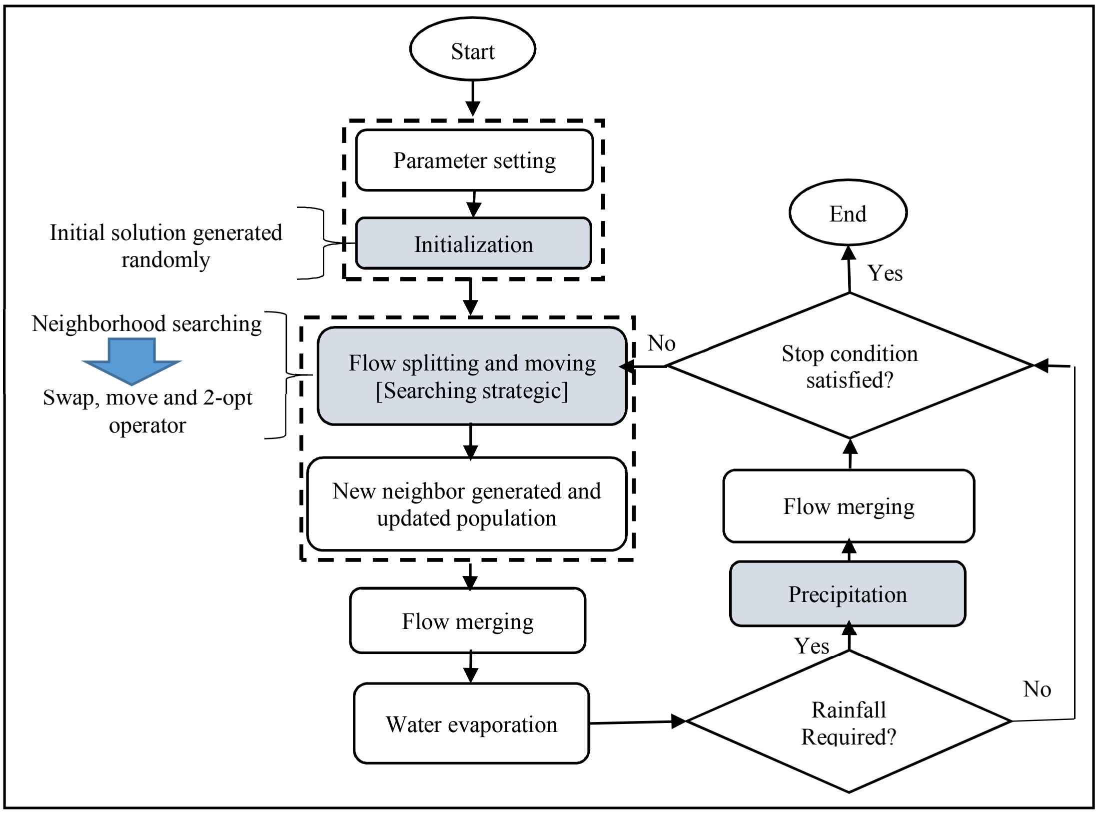


Figure 3: Flowchart of the proposed WFA-CVRP

The pseudo-code of WFA-CVRP is illustrated in figure 4.

WFA parameters setting. *(section 4.1)*
Generate an initial solution *Sol* randomly. *(section 4.1)*
bestSol ← *Sol*
Do
{
For each flow $i \in \{1,2,\ldots..N\}$ if velocity > 0.
{
Calculate the number of sub-flows *ni*. *(section 4.2)*
For each sub-flow $k \in \{1,2,\ldots. ni\}$.
{
Distribute velocity (V) and mass (M).
Applying neighborhood operator
to generate *Sol**. *(section 4.2.1)*
Calculate solution cost.
If (*f(Sol*)* < *f(bestSol)*)
{ *bestSol* ← *Sol** }
} }

Update total number of flows (solutions).
Rank solutions based on the best cost.

If (*sol** has the same cost value) Then run merging the operations *(section 4.3)* and update *Wi* and *Vi*.
Water evaporation. *(section 4.4)*

If (Regular_Precipitation conditions meet)
Then run a regular precipitation operation. *(section 4.5)*
If (Enforced_Precipitation conditions meet).

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

Then run an enforced precipitation operation.

Run flow merging operation.
Update the iteration counter.

} while (iteration counter < max_iteration_value)

Figure 4: Pseudo-code of proposed WFA-CVRP procedure.

4.1 Initial General Solution and Initial Parameter Setting

Before WFA starts its searching process, several parameters should be assigned, such as maximum generation *G*, initial flow *i* (solution), initial mass *W0*, initial velocity *V0*, the base momentum *T* for splitting, the upper limit of sub-flows split from one flow $\overline{n}$, the gravity *g*, and periodical precipitation operation *t*. The WFA-CVRP used similar parameter setting with the basic WFA [23].

The initial solution in this work is generated by selecting the customers randomly and adding it to the current route without violating the capacity of the vehicle. If the violation occurs, the customer will be removed from the route and another customer selected until there are no customers to add to this route, then we will create a new route and repeat this procedure until all customers are routed.

The flows with higher kinetic energy have more potential to split into sub-flows, the number of sub-flows will be decided from the kinetic energy. Hence, the value of M0V0=2*T* ~ 3*T* was suggested by [23], and to avoid the escalating increase of the number of sub-flows.

4.2 Flow Splitting and Moving

A WFA starts with single water flow (a single solution generated randomly), and its location is assigned based on the initial solution value. The flow starts by moving to scrutinize new locations in the solution space based on fluid momentum and potential energy. The flow might be split into sub-flows depending on the results of equation (1) that divides the momentum of the flow (W_iV_i) by the base momentum *T*. Additionally, the flow with zero momentum will stay where it is, and will be considered as a stagnant solution. On the other hand, a flow can split into sub-flows when the momentum of the flow exceeds a base momentum *T*. If the flow momentum is between *0* and *T*, ($0 < W_iV_i < T$), the flow will move to a new location as a single stream solution without forking in order to avoid increasing the sub-flow that might cause unnecessary resources consumption. [23] Each iteration determines the upper limit $\overline{n}$ as the number of sub-flows that fork to form a flow. Equation (1) calculates the number of sub-flows.

$$n_i = min\left\{\max\left\{1, int\left(\frac{W_iV_i}{T}\right)\right\}, \bar{n}\right\} \quad (1)$$

In the WFA-CVRP, the splitting and moving of any flow is associated with a neighborhood search of the current solution, where the flows can move from one location (solution) to a new location (neighbor) based on a small change of the current solution. A neighborhood for a given solution is defined as any other solution obtained by a pairwise exchange of one or more nodes in the solution. This always guarantees that any neighborhood, for a feasible solution, is always a feasible solution. Figure 5 illustrates the flow splitting and moving the mechanism during the solution searching.

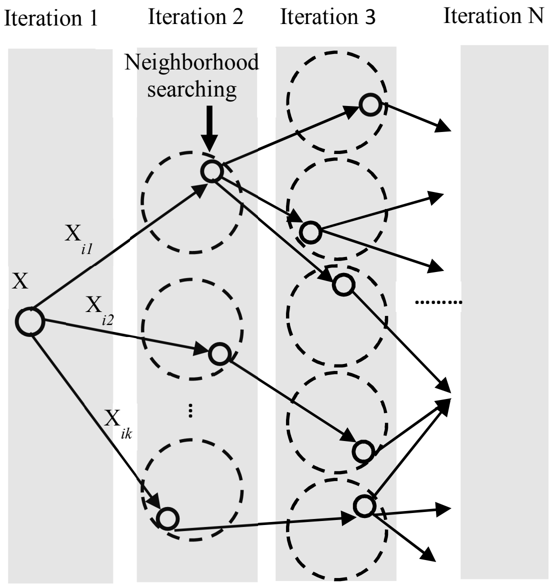


Figure 5: WFA flows splitting and moving mechanism.

4.2.1 Neighborhood operators

A neighborhood operator is employed to obtain a new neighbor *x'* from the current solution *x* in the WFA algorithm. From the literature, three common neighborhood operators are chosen for this work. From these neighborhood, each neighborhood operator is selected randomly to generate new neighbor, which are:

Random swap

Two positions are selected randomly in this operation (in the solution vector) *i* and *j* where *i*

Journal of Theoretical and Applied Information Technology
10th July 2015. Vol.77. No.1

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

$\neq j$ and swap the customers located in positions i and j. if it is feasible. See figure 6, where i=2 and j=6.

Before:

Swap point Swap point

0	7	4	0	5	8	2	0	3	6	1	0

After:

0	8	4	0	5	7	2	0	3	6	1	0

Figure 6: Random swap

Random move

In this operation, one costumer at position i is selected randomly then moved to another position j which is selected randomly where $i \neq j$ and relocating the customer from position i to position j. See figure 7, where customer *3* is relocated from position *9* to position *3*. Where number *4* and *3* represent the customer *id*, and the feasible move only is accepted.

Before:

Move position Move point

0	7	4	0	5	8	2	0	3	6	1	0

After:

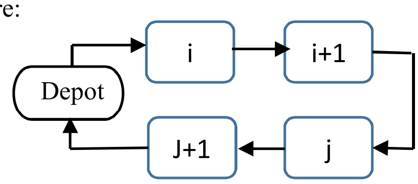
0	7	3	4	0	5	8	2	0	6	1	0

Figure 7: Random swap

2-opt operator

The 2-opt operator aims to cut two edges between sequent costumers thus leading to divide the route to many routes, then reconnect these route in some different ways in order to reduce the traveling distance. The operator is shown in figure 8, in which the route direction between customer's i-(i+ 1) and j-(j+ 1) are interchanged.

Before:



After:

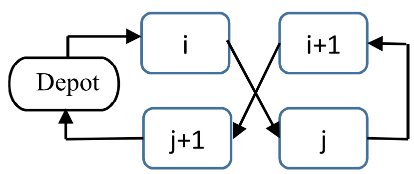


Figure 8: 2-opt operator

When the flow i splits into sub-flows n_i, the original mass of flow will be accordingly distributed to its sub-flows based on their ranks using equation (2).

$$w_{ik} = \left(\frac{n_i + 1 - k}{\sum_{r=1}^{n_i} r}\right) W_i, \quad k = 1,2,\dots,n_i \qquad (2)$$

In case: if $W_i = 5$ and $n_i = 3$

$$w_{i1} = \left(\frac{3+1-1}{1+2+3}\right)5, w_{i2} = \left(\frac{3+1-2}{1+2+3}\right)5,$$

$$and \quad w_{i3} = \left(\frac{3+1-3}{1+2+3}\right)5$$

The velocity of each sub-flow is computed using the equation of energy conservation (3). μ_{ik} is the velocity of sub-flow k, split from flow i.

$$\mu_{ik} = \begin{cases} \sqrt{V_i^2 + 2g\delta_{ik}}, & if\ V_i^2 + 2g\delta_{ik} > 0 \\ 0, & otherwise \end{cases} \qquad (3)$$

Where g is the gravitational acceleration, and δ_{ik} is the altitude drop from flow i to its sub-flow k, which is, the improvement in objective value of the current solution i to its neighborhood solution k. If $V_i^2 + 2g\delta_{ik} < 0$, means no improvement in the solution i, and it stuck at local optima without splitting or moving. Such stagnant flow will gradually evaporate into the atmosphere, returning to the ground by precipitation later on. At the end of the splitting and moving operation, the original flow is discarded because sub-flows have been generated. Information regarding the current number of sub-flows and solution sets is then recorded.

4.3 Flow Merging

When more than one flow of water moves to the same location (objective value), they merge into one flow with more mass and momentum to reduce the number of solutions at that location. The merging operation assists to prevent redundant searching by solution with the same objective value. Flow merging might help the stagnant flows to get out from the trapped location. This operation systematically checks current flows if they move to the same location. Assuming there are two flows i and j sharing the same location, flow j is merged into flow i, then the mass W_j and velocity V_j of flow j will be added to flow i using equations (4),(5), and delete flow j.

$$W_i = W_i + W_j \qquad (4)$$

Journal of Theoretical and Applied Information Technology
10th July 2015. Vol.77. No.1

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

$$V_i = \frac{W_i V_i + W_j V_j}{W_i + W_j} \quad (5)$$

4.4 Water Evaporation

In the neutral behavior of water lifecycle process, the water is evaporated into the atmosphere then returns as drops of liquid during precipitation. WFA used the concept of water evaporation and precipitation in water flows after moving from one location to another to help flows escaping from the local optima and explore within a greater solution search space. In the WFA, each flow is subjected to water evaporation, where part of the water evaporates into the atmosphere. After a prescribed number of iterations *t*; it is determined that a flow will be completely removed, meaning that the masses of all flows are decreased by the ratio of $\frac{1}{t}$ every time evaporation occurs, as illustrated in equation (6).

$$W_i = \left(1 - \frac{1}{t}\right) W_i, \ \ i = 1,2, \ldots \ldots, N. \quad (6)$$

4.5 Precipitation

There are two types of precipitation in WFA to mimic the natural behavior of water, enforced and regular precipitation, [23]. *Enforced precipitation* is used when the velocity of all flows is grounded with zero. There is no improvement for all flows in solution search after *t* iterations. In this case, all flows must evaporate and then return to the ground as precipitation with the same number of current flows, and the location of these poured flows are deviated randomly from the original ones. In WFA-CVRP, we used random swap between two customers in the solution to reassign new locations randomly to the flows. After relocation, the new position of all flows, the initial mass W_0 is proportionally distributed to flows based on their original mass with the same initial velocity V_0 using equation (7).

$$W_i' = \left(\frac{w_i}{\sum_{k=1}^{N} W_k}\right) W_0 \, , i = 1,2 \ldots , N \quad (7)$$

Regular precipitation is applied after *t* iteration (same *t* value as in evaporation) returns the evaporated water back and thereafter aims to balance with water evaporation. It can be noted that, the accumulated mass of the evaporated water is $W_0 - \sum_{k=1}^{N} W_k$ that is reassigned to ground flows, as in equation (8). However, after precipitation is applied, the number of current solutions will increase due to the newly poured flows joining the current solution set. Additionally, both the enforced and regular precipitations may add several new flows at the same locations (objective values) as those of other flows. Hence, the merging operation will be applied to remove possible redundant solutions.

$$W_i' = \left(\frac{w_i}{\sum_{k=1}^{N} W_k}\right) W_0 - \sum_{k=1}^{N} W_k \quad (8)$$

Three neighborhood structures (swap, move and 2-opt operator) were selected randomly to generate a number of neighbors as sub-flows from the original solution. The new neighbors have been generated randomly then added to the population if it is feasible then sort the population. A flow movement is a solution search from the current solution to a new solution. A new solution will not be generated from the original solution if its momentum is zero, and considered as a stagnant solution.

5. EXPERIMENTAL RESULTS

The proposed algorithm WFA-CVRP was coded in Java platform JDK 1.6, a Windows environment and a personal computer with an Intel core i7 (2.20 GHz CPU speed and 8 GB RAM). Use 14 CVRP datasets with the size ranging from 21 to 261 customers, were selected form http://www.coinor.org/SYMPHONY/branchandcut/VRP /data/, in order to measure the effectiveness and scalability of the proposed algorithms for solving CVRP.

WFA involves many parameters that need to be tuned such as stopping criteria (iteration limit), initial mass, initial velocity, base momentum, and the total number of water flows. Since the parameters in WFA are dynamic, we set the initial value for each of them. Then the algorithm adjusts these parameters in a dynamic way during the optimization process. WFA parameters are set founded on preliminary experiments that are based on the recommendation of the original paper [23]. See Table 2.

Table 2: WFA parameters

	Parameter	Value
1	Base momentum T	20
2	Initial mass W_0	8
3	Initial velocity V_0	5
4	Limit number of sub-flows $\overline{n}$	3

Journal of Theoretical and Applied Information Technology
10th July 2015. Vol.77. No.1

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

We compared the results of different iterations to set the proper stopping conditions of the algorithm. The algorithm ran with 31 independent runs for each iteration limit.

For statistical analysis, Wilcoxon test has been used to evaluate the effect of the iteration limit on the performance of the algorithm. The results of the statistical analysis are significantly different for most of the instances. This rejects the null hypothesis that the iteration limit does not affect the performance of the algorithm. It has been used to evaluate the effect of the parameters on the performance of the algorithm. This results answer the second research question raised in section 1 (introduction).

Table 3 shows the minimum distance result of the WFA with different numbers of iterations. From Table 3, 500000 iterations resulted in the most number of minimum distances compared to other iterations. 500000 iterations is more efficient solution compared to 1000000 since it produces comparable minimum distances with 1000000 iterations with half the number of iterations.

***Table 3**: The minimum distance result for different iterations during 31 runs*

Instance	Different Number of Iteration				
	1000	*50000*	*100000*	*500000*	*1000000*
E-n22-k4	375	375	375	375	375
E-n23-k3	594	568	569	568	568
E-n33-k4	924	837	847	837	837
E-n51-k5	764	575	574	545	541
E-n76-k8	1272	805	808	774	770
E-n76-k10	1337	922	907	876	899
E-n101-k8	1767	1013	937	878	854
E-n101-k14	1849	1214	1172	1144	1120
att-n48-k4	59628	42728	40795	40471	40472

We compared between WFA and other recent algorithms from literature, such as: ant colony optimization (ACO) [20], [21]. Most the recent algorithms are enhancement algorithm such as hybrid genetic algorithm (HGA) [19], a hybrid ant colony system (HACS) [10], [22]. Table 4 shows the comparison results between of WFA with algorithms reported in the literature for some instances. From Table 4, it can be concluded that the basic WFA obtained the best result in four instance (E-n22-k4, E-n23k3, E-33-k4 and att-n48-k4). Furthermore, it produced better solution compared with basic ACO and ACO+2opt [20] in all instances and better than HAA [20] in most instances except three instances (E-n51-k5, E-n-101-k8 and E-n-101-k14). WFA obtained the same results as HGA in one instance which is E-n23-k3 and better than HGA in instance E-n33-k4. WFA obtained better results compared with basic ACO [21] in two instances (E-n76-k10 and M-n200-k17).

From Table 4 it can be noted that even though WFA did not manage to obtain the best results for the other 9 instances, however the quality of these solutions are competitive, where the differences are very little. The results reveal that WFA is able to obtain the promising results. The main research question in section 1 (introduction) was answered in this experiment.

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

Table 4: Comparing the results of WFA with basic ACO and Hybrid Ant Algorithm (HAA).

Instances	BKS	ACO 2012 [20]	ACO+2-OPT 2012 [20]	HAA 2012 [20]	HGA 2009 [19]	ACO 2004 [21]	SW+ACS 2012 [10]	HACS 2013 [22]	WFA
E-n22-k4	**375**	418.3	376.5	376.5	-	-	-	-	**375.28**
E-n23-k3	**569**	618.7	579.3	577.4	**568.56**	-	-	-	**568.56**
E-n33-k4	**835**	880.4	865.2	842.5	845.24	-	-	-	**837.67**
E-n51-k5	**521**	639.9	606.6	535.9	524.61	**521**	**521**	524.6	545.24
E-n76-k8	**735**	944.9	911.5	783.1	**750.48**	-	-	-	774.41
E-n76-k10	**830**	1113	1065	895.7	853.05	877	**838**	843.5	876.78
E-n101-k8	**817**	1109	1103	844.5	847.5	845	839.2	**835.3**	878.55
E-n101-k14	**1071**	1433	1431	**1103**	1121.3	-	-	-	1144.11
att-n48-k4	**40002**	44970	43010	40891	-	-	-	-	**40471.08**
M-n101-k10	**820**	-	-	-	-	838	823.7	**819.6**	864.95
M-n121-k7	**1034**	-	-	-	-	1189	**1050**	1090	1285.32
M-n151-k12	**1053**	-	-	-	-	1105	**1030.5**	1054.9	1149.37
M-n200-k17	**1373**	-	-	-	-	1606	**1325.6**	1377.7	1462.22
G-n262-k25	**6119**	-	-	-	-	-	-	-	**6403.05**

6. CONCLUSION

This work has presented a basic WFA algorithm for CVRP, which differs from the basic meta-heuristic algorithm due to dynamic and self-adaptive population size of solution and tuning parameter during the optimization process. It can overcome the drawbacks of both single and multiple solutions based algorithms. The WFA-CVRP uses random steps for initialization, applies three neighborhood search strategies for splitting and moving which are (move, swap and 2-opt operator), and they have been selected randomly to generate a number of neighbors as sub-flows from the original solution. The experimental results demonstrate that WFA-CVRP is competitive compared with other algorithms in literature based on the quality of the solution. The WFA for CVRP has many room for further improvement especially on exploration on decision of the splitting and exploitation after flow merging. For future work, we suggest to use nearest neighbor heuristic to generate the initial solution that lead the algorithm to start to good quality solution. Furthermore, WFA can be hybridized with other local search algorithm to make a balance between intensification and diversification of search process.

ACKNOWLEDGEMENT

This work was funded by the Fundamental Research Grant FRGS/1/2013/ICT07/UKM/02/3 from the Ministry of Education, Malaysia.

REFERENCES:

[1] Y. Zhou, J. Xie, and H. Zheng, “A Hybrid Bat Algorithm with Path Relinking for Capacitated Vehicle Routing Problem”, *Mathematical Problems in Engineering* 2013.

[2] C. Qi, and P. Li, “An Exponential Entropy-based Hybrid Ant Colony Algorithm for Vehicle Routing Optimization”, *Applied Mathematics & Information Sciences* Vol. 8. No. 6, 2014, pp. 3167-3173.

[3] G. B. Dantzig, and J. H. Ramser, “The truck dispatching problem” *Management science* Vol. 6, No. 1, 1959, pp. 80-91.

[4] P. Toth, and D. Vigo, “The vehicle routing problem”, *Siam,* Vol. 9, 2002.

[5] A. S. Bjarnadóttir, “Solving the vehicle routing problem with genetic algorithms”. Diss. Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2004.

[6] J. Rust, “Dynamic Programming” *London, UK, Palgrave Macmillan, Ltd,* 2008.

[7] D. Vigo, “Branch-and-Bound Algorithms for the Capacitated VRP”, *the vehicle routing problem* 29, 2001.

[8] J. E. Mitchell, “Branch-and-cut algorithms for combinatorial optimization problems”, *Handbook of Applied Optimization,* 2002, pp. 65-77.

[9] M. Dell’Amico, G. Righini, and M. Salani, “A branch-and-price algorithm for the vehicle routing problem with simultaneous pick-up and delivery”, *Transp Sci*, Vol. 40, 2006, pp. 235-247.

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

[10] M. Yousefikhoshbakht, and E. Khorram, “Solving the vehicle routing problem by a hybrid meta-heuristic algorithm”, *Journal of Industrial Engineering International,* Vol. 8, No. 1, 2012, pp. 1-9.

[11] E.-G. Talbi, “Metaheuristics: From Design to Implementation”, *John Wiley and Sons,* Vol. 74, 2009, 624.

[12] S. W. Lin, Z. J. Lee, K. C. Ying, and C.Y Lee. “Applying hybrid meta-heuristics for capacitated vehicle routing problem.” *Expert Systems with Applications,* Vol. 36, No. 2, 2009, pp. 1505-1512.

[13] G. U. Clarke, and J. W. Wright, “Scheduling of vehicles from a central depot to a number of delivery points.” *Operations research,* Vol. 12, No. 4, 1964, pp. 568-581.

[14] B. E. Gillett, and L. R. Miller, “A heuristic algorithm for the vehicle-dispatch problem”. *Operations research,* Vol. 22, No. 2, 1974, pp. 340-349.

[15] R. Tavakkoli-Moghaddam, N. Safaei, and Y. Gholipour, “A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length”, *Applied Mathematics and Computation,* Vol. 176, No. 2, 2006, pp. 445-454.

[16] Y.-J. Kwon, J.-G. Kim, J. Seo, D.-H. Lee, and D.-S Kim, “A tabu search algorithm using the voronoi diagram for the capacitated vehicle routing problem” *Computational Science and its Applications, ICCSA, International Conference on*, 2007, pp. 480-488.

[17] Y. Marinakis, “Multiple Phase Neighbourhood Search-GRASP for the Capacitated Vehicle Routing Problem”, *Expert Systems with Applications,* Vol. 39, No. 8, 2012, pp. 6807-6815.

[18] B. M. Baker, and M. A. Ayechew, “A genetic algorithm for the vehicle routing problem.” *Computers & Operations Research* 30, No. 5, 2003, pp. 787-800.

[19] C. H. Wang, and J. Z. Lu, “A hybrid genetic algorithm that optimizes capacitated vehicle routing problems.” *Expert Systems with Applications* 36.2, 2009, pp. 2921-2936.

[20] Z. Xiao, and W. Jiang-qing, “Hybrid Ant Algorithm and Applications for Vehicle Routing Problem.” *Physics Procedia* 25, 2012, pp. 1892-1899.

[21] S. Mazzeo, and I. Loiseau, “An ant colony algorithm for the capacitated vehicle routing.” *Electronic Notes in Discrete Mathematics* 18, 2004, pp. 181-186.

[22] E. E. Adiba, M. Elhassania, and E. A. Ahemd “A hybrid ant colony system for green capacitated vehicle routing problem in sustainable transport.” *Journal of Theoretical and Applied Information Technology* 54, No. 2, 2013.

[23] F.-C. Yang, and Y.-P. Wang, “Water flow-like algorithm for object grouping problems”, *Journal of the Chinese Institute of Industrial Engineers*, Vol. 24, No. 6, 2007, pp. 475-488.

[24] C. Blum, and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison.” *ACM Computing Surveys (CSUR)* Vol. 35, No. 3, 2003, pp. 268-308.

[25] T.-H. Wu, S.-H. Chung, and C.-C. Chang, “A water flow-like algorithm for manufacturing cell formation problems”, *European Journal of Operational Research,* Vol. 205, No. 2, 2010, pp. 346-360.

[26] C.-H. Lee, and C.-T. Kuo, “Dynamic solution agent algorithm for TSK-type interval-valued fuzzy system optimization”, *Industrial Electronics and Applications (ICIEA), 7th IEEE Conference on,* 2012, pp. 1297-1302.

[27] K. De Jong, “Parameter setting in EAs: a 30 year perspective” Dlm. (pnyt.). *Parameter Setting in Evolutionary Algorithms*, *Springer Berlin Heidelberg*, 2007, pp. 1-18.

[28] C. Neumüller, S. Wagner, G. Kronberger, and M. Affenzeller, “Parameter meta-optimization of metaheuristic optimization algorithms”, Dlm. (pnyt.). *Computer Aided Systems Theory–EUROCAST 2011 Springer,* 2012, pp. 367-374.

[29] Z. A. Othman, A. I. Srour, A. R. Hamdan, and P. Y. Ling, “Performance Water Flow-Like Algorithm for TSP by improving its Local Search”, *International Journal of Advancements in Computing Technology (IJACT),* 2013.

[30] A. Srour, Z. A. Othman, and A. R. Hamdan, “A Water Flow-Like Algorithm for the Traveling Salesman Problem”, *Advances in Computer Engineering*, 2014.