International Journal on Advanced Science Engineering Information Technology
Vol.9 (2019) No. 5
ISSN: 2088-5334

A Hybrid Water Flow-Like Algorithm and Variable Neighbourhood Search for Traveling Salesman Problem

Mohamed Rafique Othman[#1], Zulaiha Ali Othman[#2], Ayman Ibraheem Srour[*], Nor Samsiah Sani[#3]

[#]*Center for Artificial Intelligence and Technology, Faculty of Information Science and Technology, UKM, Bangi, Selangor, Malaysia*
E-mail: [1]rafiqueothman@gmail.com; [2]zao@ukm.edu.my; [3]norsamsiahsani@ukm.edu.my

[*] *CODeS, Department of Computer Science, KU Leuven, Belgium*
E-mail: aymanih.srour@kuleuven.be

Abstract*—Various metaheuristic methods have been proposed earlier and applied for solving the Travelling Salesman Problem (TSP). Water Flow Algorithm (WFA) is one of the recent population-based metaheuristic optimization techniques used for solving this problem. Past research has shown that improving WFA local search strategy has a significant impact on the algorithm performance. Therefore, this paper aims to solve TSP by enhancing WFA searching strategy based on a Variable Neighbourhood Search (VNS) known as hybrid WFA-VNS. It is a mixture of the exploration of WFA and the exploitation capability of VNS. This study is conducted in two stages: Pre-experiment and initial experiment. The objective of doing pre-experiment is to select four neighborhood structures to be used for the initial experiment. At the first stage, three instances are used, and there are five neighborhood structures involved. Those neighborhood structures are two opt, three opt, four opt, swapping, and insertion move. Because of pre-experiment, it discovers four best neighborhood structures, which are two opt, three opt, exchanging and insertion move. These neighborhood structures will be used in the initial experiment, which an improvement approach is employed. In an initial experiment, the performance of the proposed hybrid WFA-VNS is further studied and tested on 26 established benchmarked symmetric TSP datasets using four neighborhood structures selected in pre-experiment earlier. The TSP datasets involved are categorized into three types: small datasets, medium datasets, and large datasets. Selected neighborhood structures obtained in pre-experiment are applied and generated randomly to intensify the initial solution achieved at an earlier stage of hybrid WFA-VNS. The results of the comparison show that this hybrid approach represents an improvement and able to produce competitive results.

***Keywords*—hybrid algorithm; water flow; variable neighbourhood search; travelling salesman problem; metaheuristic.**

I. INTRODUCTION

Travelling Salesman Problem (TSP) is a widespread combinatorial optimization problem that falls into a non-deterministic polynomial-time category (NP-hard) [1], [2]. A company in the United States to resolve the path problem using linear programming, a well-known problem in the computer science field presently [3], introduces TSP. TSP is widely used in real-world applications such as vehicle routing, scheduling problems, integrated circuits designs, graph theory, and gene ordering [4]. It is a classic problem in combinatorial optimization in the field of computer science that has been solved for decades. It is typically used as a standard problem for testing the efficiency of a newly proposed optimization algorithm [5]. TSP also is recognized as one of the combinational optimization problems of discovering the best solution out of a finite set of promising resolutions [6]. TSP has primarily been used for demonstrating many real-life routine cases as well [7]. The objective of any optimization algorithm used for solving TSP is to find the minimum length of the tour among all possible cities with a minimum cost of visiting all available cities efficiently and require returning safely to the starting point. Technically speaking; finding the best solution within reasonable execution time is crucial. The scenario is given a set of cities and each city associated with a cost. The objective is to travel each city and back to the starting city with the best route or least cost (least total distance or most efficient) [8]. It requires finding a tour of minimum weight (costs) that visits each city (vertex) only once.

Solving TSP is vital since the applications of TSP fall in many fields. It includes computer wiring, frequency assignment in communication networks and machine sequencing, vehicle routing, and modern engineering [9].

TSP is one of the significant sources of impact from various other disciplines as well. The last three decades speak for it [10]. Until today, the search for an effective technique of solving TSP is still a phenomenal effort. In recent years, many metaheuristic approaches were proposed and applied for solving TSP [32]. Some approaches are population-based. Population-based approaches can be categorized as either evolutionary algorithms (EA) or swarm intelligence (SI) [11], [12] that utilize the obtained solutions and begin to improvise it [13]. One of the popular methods in population-based is a Water-Flow Algorithm (WFA) [8], [14].

Naturally, water flow is controlled by gravity force. It benefits water flows from a higher altitude to lower altitude. The water flow capacity of splitting or merging depends on the surface scenery. With a proper amount of velocity, it could overcome or breaks the obstacle as well and continuously flow into stream, rivers, and lakes make its way toward its final point. Ultimately, the water will evaporate or vaporize to the atmosphere and return to earth through precipitation and condensation. Water always moves through this natural cycle, and WFA is designed to simulating this natural event [8].

WFA has shown a good achievement in several optimization problems such as bin-packing problem (BPP) [15], manufacturing cell fraction problem [16], nurse-scheduling problem (NSP) [17], and traveling salesman problem (TSP) [18], which is the primary concern of this paper. Based on past studies, it shows that the WFA not only has promising potential for solving several optimization problems but also reveals there is plenty of room for improvement exists in WFA. One of those is enhancing the WFA searching strategy.

Variable neighborhood search (VNS) is metaheuristic algorithms that generate based upon the searching strategy of using more than one neighborhood structure [19]. It can change these structures methodically throughout the neighborhood structures, thus preventing it from being a trap at local optima since the solution obtained from one neighborhood is not belongs to other neighborhood. It is an efficient method for searching for approximated answers [20]. VNS has been applied to several NP-hard problems such as TSP, location-allocation [21], course-timetabling problem [19], scheduling problems [20], the p-median [23] and vehicle routing [22]

Recently, a different kind of algorithm has been hybridizing with WFA to tackle TSP. For example, WFA-TS [8] and WFA-SA [24]. The obvious conclusion from these studies indicates that WFA can integrate well with another metaheuristic.

II. Materials and Method

In this study, a proposed hybrid technique is applied to TSP to observe the algorithm performance. The aim of this hybrid approach is to enhance the searching strategy performance of WFA by combining the exploration of WFA and the exploitation capability of VNS. This section explaining a dataset used the related algorithm and a description of the proposed method.

A. Set Data

The performance of the proposed WFA-VNS is evaluated by using the similar standard benchmark TSP datasets used in [25]. The TSPLIB is collected by the Institute of Information of the University of Heidelberg, German in 2009. It consists of 26 established symmetric TSP datasets.

These datasets have 3 categories which small-scale, medium-scale and large-scale of TSP data containing two coordinates, *x* and *y*. Small-scale datasets are berlin52, bier127, ch130, ch150, eil101, eil51, eil76, kroA100, kroA150, kroA200, kroB100, kroB150, kroB200, kroC100, kroD100, kroE100 and lin105. Medium-scale datasets include lin318, rat575, rat783, u1060 and fl1400. While d1655, u1817, d2103 and fl3795 representing large-scale datasets. These datasets vary by city size as shown in Table I to select the best neighborhood structures in pre-experiment.

The distance used in this experiment is Euclidean distance, where the gap between the cities d_{ij} is measured using the Euclidean distance formula, as shown in equation one below.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

x_i and y_i is a coordinate for city A, while x_j and y_j is a coordinate for city B.

TABLE I
SIZE FOR EACH DATASET CATEGORY

Category	City Size
Small datasets	51-300
Medium datasets	300-1000
Large datasets	1000-4000

B. WFA

Ying and Wang invent WFA in 2007. It was introduced for solving Object Grouping Problems [8]. In WFA, a stream of water flow act as the solution agents. Terrain or landscape represents the objective function. It is a self-adaptive and it has the capability to address the population size dynamically. These are the main attraction of WFA. It featured the ability to explore effectively and potentially able to attain a better solution via its solution agents as the water move, split and merge. The percentage of gaining potential good results slightly increases.

The gravitational force and the law of energy conservation influence stream water. Stream flow has two components; the velocity and the volume of water in the stream. Both of the two components may change during the lifetime of a stream. A stream with massive velocity will able to create more sub-flow while a stream will remain single if the velocity is low. Another factor to analyze is the landscape or the topography of the surrounding. Streams are channelized into many flows once it hits an obstacle or a complex barrier. Thus, it will give greater impact to the water movement and the velocity as well.

The role of iteration is to illustrate the movement of water, which causes the water to move towards a lower altitude simulating the natural water cycle. At some point, some of the water flow will combine as one flow when it reaches flat scenery or its final point or any. It could be a river or ocean. Here, evaporation will transport water back to the atmosphere. It indicates the role of iteration. In the search process, four primary operations include in WFA: Flow and

Split, Merging, Evaporate, and Precipitation. The initial setting of mass (w), velocity (v) and the initial flow splits are based on its velocity. Mass and velocity are updated using the following formula in equation 2 and 3, while the flow is updated using equation four below correspondingly:

$$W_i = W_i + W_j \tag{2}$$

$$V_i = \frac{W_i V_i + W_j V_j}{W_i + W_j} \tag{3}$$

$$W_i = \left(1 - \frac{1}{t}\right) W_i,\ i = 1,2,..N, \tag{4}$$

C. Variable Neighborhood Search (VNS)

Mladenovic and Hansen [26], [27], introduced variable neighbourhood search (VNS). Metaheuristic algorithms operate based upon the strategy of using more than one neighborhood structure. VNS has the capability to change these structures systematically throughout the neighborhood structures thus preventing from being trap at local optima since the solution obtained from one neighbourhood is not belongs to other neighborhood. It can be considered as continuous re-optimization approach [28]. It is an efficient method for searching approximated solutions [20]. VNS has been applied to several NP hard problems such as the travelling salesman problem (TSP), location-allocation [21], course timetabling problem by Abdullah et al. [19], scheduling problems [20], vehicle routing [22] and the p-median [23]. More details on VNS can be found in Hansen and Mladenovic [21]. VNS has been previously applied in a variety of combinatorial problems [29].

D. The Hybrid: WFA-VNS

In proposed method, we applied four neighborhood structures to this problem for searching a near-optimal solution of a TSP. This study involves two experimental phases which pre-experiment and initial-experiment. The differenced of these two experiments is only the involvement of neighbourhood structures. Five neighborhood structures involved in pre-experiment. Those neighborhood structures are 2-opt, 3-opt, 4-opt, swapping and insertion move. The objective of pre-experiment is to determine the best four neighborhood structures that will be used for the initial experiment.

Exploring entire neighborhood structures in sequence in single iteration is an extremely extensive task because it will consume time. Therefore, we identified one method to overcome this situation. These four neighborhood structures generated randomly instead in sequence. We believe, by incorporate WFA with this mechanism, it might improve the overall computational time. Whenever one neighborhood generates a better solution, the VNS will starts over executing another neighborhood structure, which is selected randomly to further improvise the current better solution. At this point, it might refine the solution using the same neighborhood structure used at previous iteration or it might use others because there are added three neighborhood structures ready for the task. The pseudo code of the proposed WFA is illustrated in figure 1.

```
Start
Generating Initial Solution (Initial Flow)
For number of flow (flow size) //1. Flow and Split Operation
If (split required) {
Calculate number of sub-flow required based of flow momentum;
For each sub-flows {
Find best neighborhood solution using 2-opt;
Distribute mass of flow to its sub-flows;
Calculate the improvement in objective function ;}
End For;
update mass and velocity for each flow
put updated sub-flow in sub-flow array}
Else        //Flow Operation
put back the flow in the population
Put all the sub-flow into population
Parameter setting for neighborhood structures
Initial Solution = the best solution from the population
for n iteration;
For (int n = 1; n <= 5; n++)
 {
 switch (n)
   { //each neighborhood structure will execute 3 instances: instance1, instance2, instance3
    case 1 : neighborhood structure1
     break;
    case 2 : neighborhood structure2
     break;
    case 3 : neighborhood structure3
     break;
    case 4 : neighborhood structure4
     break;
    case 5 : neighborhood structure5
     break;
      }
    return result_instance1;
    return result_instance2;
    return result_instance3;
   }
End For;
Until population number is reached;
//2 Merging Operation
Merge sub-flows with same objective function;
End If
Update total no. of water flows;
if evaporation condition met;    //3. Evaporation Operation
for each sub-flow;
Perform Water evaporation;
end for
End If;
If Regular Precipitation condition met;
// 4. Precipitation Operation
Perform Regular Precipitation
End If;
If (Enforce Precipitation condition met)
Perform Enforce Precipitation
End If;
Until required number of generations are generated;
End
```

Fig. 1 Pseudo-code for the WFA applied in pre-experiment

```
Generating Initial Solution randomly;
while (termination condition is not met) do
(a) Shaking: Neighborhood_curr;
(b) Local search: x ← Local Search (x);
(c) Move or not: if (cost(x) < cost(best));
 best ← x;
 Break;
```

```
Total++;
End
Return (best solution found)
```

Fig. 2 Pseudo-code for the VNS applied in initial experiment

III. Result and Discussion

This study involved two stage of experiments. The first stage is the pre-experiment that focusing on selecting the best four neighborhood structures. At this stage, five neighborhood structures are used (2 opt, 3 opt, 4 opt, swapping and insertion move) on three type of instances: eil101 (small dataset), lin318 (medium dataset) and fl1400 (large dataset). In the initial experiment, the performance of the proposed hybrid WFA-VNS is studied and tested on 26 established benchmarked symmetric TSP datasets using four neighborhood structures selected in pre-experiment. These neighborhood structures applied randomly to intensify the initial solution obtained at an earlier stage of enhanced WFA-VNS.

A. Phase 1: Pre-Experiment

In pre-experiment, five neighborhood structures involved two opt, three opt, four opt, swapping, and insertion move. Executions of three instances accomplished separately. Each instance represents small dataset, medium dataset, and large dataset. The solution obtained via split and flow of WFA enhanced by a neighborhood structure. Instances involved eil101 (small instance), lin318 (medium instance) and fl1400 (large instance).

The objective of conducting pre-experiment is to determine the value of the standard deviation of all neighborhood structures against each instance: eil101, lin318, and fl1400. Figure I illustrate the flow of pre-experiment conducted in this study. Results obtained are gathered and observed to investigate and analyze the stabilization of each neighborhood structure for all instances. The critical aspect of validating is the value of standard deviation of each instance executed by neighborhood structures.

TABLE II
Results of Standard Deviation of Each Neighbourhood Structure for Each Instance Processed

Neighborhood structures	**Standard Deviation**		
	Small Instance	**Medium Instance**	**Large Instance**
	eil101	**lin318**	**fll400**
2 opt	1.2293	188.2676	27.9333
3 opt	1.9889	152.4735	47.4478
4 opt	2.9609	230.5573	65.0197
Insertion	2.3664	161.1658	51.1756
Swapping	2.5734	102.9682	44.5298

Table II shows the observed results. According to the standard deviation values in Table II, 4 opt show a higher value for all type of instances. Which means, 4 opt will be automatically excluded since the initial experiment requires only four neighborhood structures. Phase II requires only four neighborhood structures.

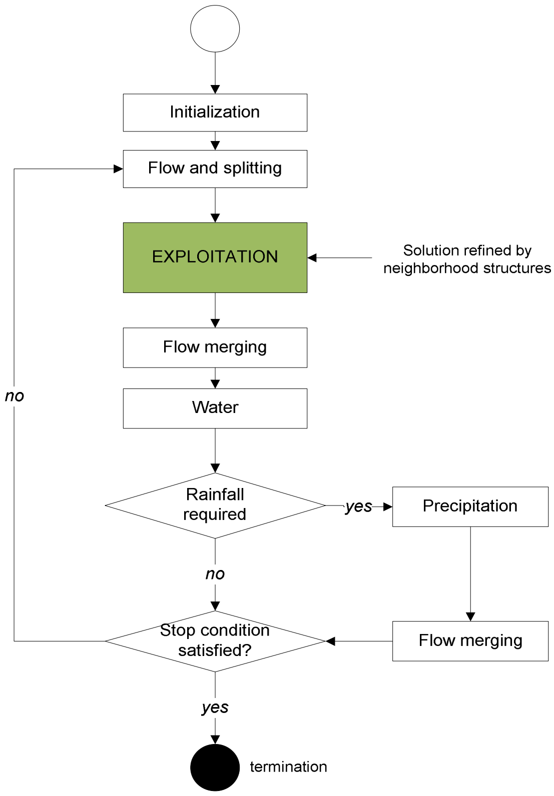

Fig. 3 Results of standard deviation of each neighborhood structure for each dataset processed

B. Phase II: Initial Experiment

At this phase, selected neighborhood structures were starts by randomly generating the initial population in the search space. The goal is to obtain the better solution in the search space. There is a process called split and move. There, the stream of water acts as a solution agent for diversification or exploration.

The split of water and depending on the velocity and mass, it furthers the exploration process. The feasible solution obtained is enhanced by the neighborhood structures namely 2-opt [24]. In our work, we modified the algorithm [24] to improve the feasible solution obtained. After the process of flow and split, we applied VNS to generate the best solution obtained. The aim is to examine if we can improve the result obtained in the initial experiment. For WFA, we are using the parameters adapted from [15, 25]. The intensification of WFA is implemented by integrating VNS: a set of neighborhood structures applied by using four neighborhood structures. In [31], four different neighborhood structures were applied randomly during execution for searching a near-optimal solution of a TSP datasets. As mentioned earlier, the role of VNS in this initial experiment is to exploit the information obtain from flow and split and use it to enhance the searching process. The neighborhood structures involved two-opt, three-opt, swapping and insertion move are randomly chosen and executed.

At the beginning, the initial feasible solution is set to be feas_Sol. In a while loop, VNS will execute one of the neighbourhood randomly to refine feas_Sol to obtain Temp_Sol. The best solution among Temp_Sol is identified and become New_Sol. The incumbent solution New_Sol moves until no improvement is detected. The search for an improvement of a current New_Sol is continued in the next iteration and using one of these four neighborhood structures regardless the improvement is found or not in a previous neighborhood search. Simply describe, for each iteration VNS will returns the best individual found from the selected neighborhood. The current solution is updated, and begins to execute next iteration within VNS. As the iterations continue, it can be expected that all solutions will aggregate towards the best global best. This whole search procedure is repeated until the termination criterion is met. In this algorithm, the termination criterion is set as a number of iterations. Figure 3 represents the flow chart of our approach.

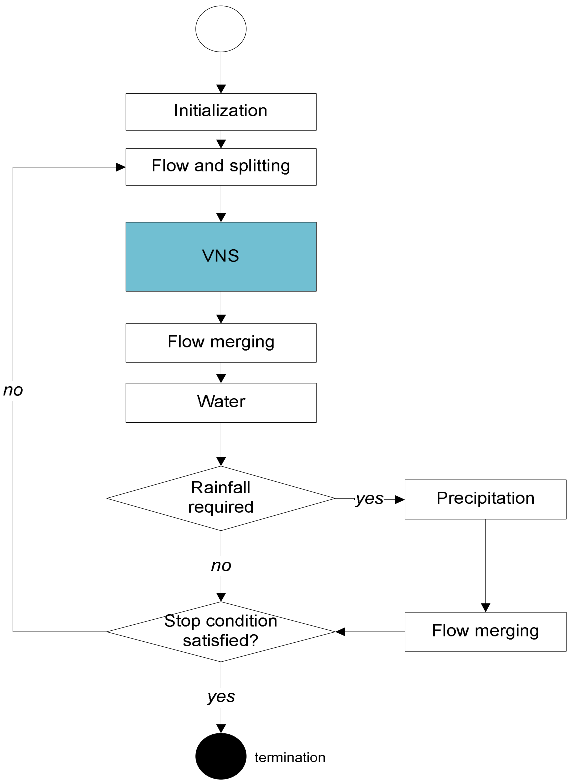

Fig. 4 The block diagram illustrates the flow of initial experiment

In this experiment, WFA use the parameter setting that has been implemented in several successfully experiments of WFA [15, 25] while VNS require termination criterion. Table III show the parameter used in WFA-VNS.

TABLE II
PARAMETER SETTING OF WFA-VNS

	Symbol	**Value**
Base Velocity, T	T	20
Initial Mass, Wo	Wo	8
Initial Velocity	Vo	5
Sub flow number limit	ň	3
VNS Attribute	-	Movement
Objective	-	Accepting rightest solution (least value)
Termination Criteria	-	Iteration

We began validating the strength of our approach by evaluating the results on 26 established benchmarked symmetric TSP datasets. The algorithm is set to execute 10 times for each dataset. Each execution consists of 10,000 iterations.

Table IV shows the comparison between WFA-TSP [15] and proposed algorithm WFA-VNS. The result justified hybridizing WFA with VNS furthering improve the solutions.

TABLE III
RESULTS COMPARISON BETWEEN WFA BASIC AND WFA-VNS

No	**Datasets**	**BKS**	**WFA**		**WFA-VNS**	
			Best Solution	**Time**	**Best Solution**	**Time**
1	berlin52	7542	7542	0.03	7542	0.07
2	bier127	118282	118282	3.56	118282	**0.55**
3	ch130	6110	6110	3.45	6110	**3.43**
4	ch150	6528	6528	3.53	6528	8.09
5	eil101	629	629	1.23	629	**0.5**
6	eil51	426	426	0.22	426	**0.05**
7	eil76	538	538	0.55	538	**0.34**
8	kroA100	21282	21282	0.56	21282	0.65
9	kroA150	26524	26524	4.41	26524	**2.7**
10	kroA200	29368	29368	9.81	29368	**6.15**
11	kroB100	22141	22141	0.99	22141	**0.72**
12	kroB150	26130	26160	4.19	**26130**	**2.88**
13	kroB200	29437	29634	7.44	**29508**	13.18
14	kroC100	20749	20749	0.53	20749	**0.17**
15	kroD100	21294	21342	1.08	**21294**	**0.79**
16	kroE100	22068	22106	1.14	**22068**	**1.01**
17	lin105	14379	14379	0.49	14379	**0.10**
18	lin318	42029	42278	20.88	**42225**	34.27
19	rat575	6773	6971	59.48	**6900**	107.81
20	rat783	8806	9126	112.25	**9113**	183.94
21	u1060	224094	230481	231.99	**229639**	418.37
22	fl1400	20127	20365	1297.39	**20340**	**741.2**
23	d1655	62128	64313	746.34	**63980**	1049.19
24	u1817	57201	59713	980.08	**59445**	1307.56
25	d2103	80450	81769	1240.28	**81640**	1684.40
26	fl3795	28772	29400	4971.05	**29350**	6795.17

At glimpse we can identify WFA-VNS is able to produce best solutions on all datasets. According to the Table IV, 1-17 represent small datasets, 18-22 represent medium datasets, and 23-26 represent large datasets.

Comparison and analysis with WFA revealed that WFA-VNS could obtain new best results in term of solutions for large datasets and medium datasets. For small datasets, the WFA-VNS manages to retain all the best value of best-known results. Comparing to the basic WFA [22], it seems WFA-VNS obtained new best solutions for the remaining 4. It increases from 0.11% to 0.42%. It occurs on kroB150 (0.11%), kroE100 (0.17%), kroD100 (0.22%) and kroB200

(0.42%). The basic WFA (Ayman 2015) unable to obtain the best-known results on these datasets. For medium datasets, WFA-VNS capable to find feasible solutions for all datasets, which are lin318 (0.12%), fl1400 (0.12%), rat783 (0.14%), u1060 (0.36%) and rat575 (1.0%). The performances of WFA-VNS increases 0.12% to 1.0%. Performance of WFA in terms of solution continues on large datasets. It increases its performance from 0.16% to 0.52%: d2103 (0.16%), fl3795 (0.17%), u1817(0.46%), d1655 (0.52%).

The initial standard deviation is computed and illustrated via histograms. A low standard deviation indicates that the data points tend to be close to balance or steadiness. As illustrates by Figure 4, WFA-VNS indicates a lower standard deviation for nine small datasets (bier127, ch130, ch150, eil101, eil51, kroB100, kroB150, kroD100, kroE100). Meanwhile in Figure 5, WFA outperform WFA-VNS for all medium datasets (lin318, rat575, rat783, u1060, fl1400). As for Figure 6, WFA-VNS has reduced its standard deviation on all large datasets (d1655, u1817, d2103, fl3795).

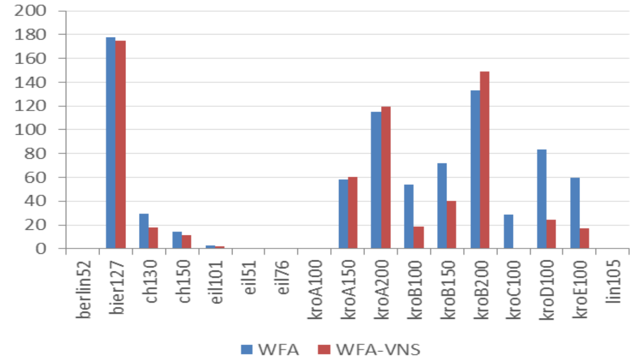

Fig. 5 Standard deviation comparison between WFA and WFA-VNS for small dataset

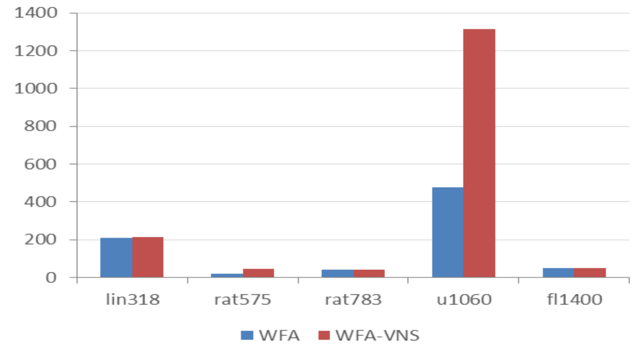

Fig. 6 Standard deviation comparison between WFA and WFA-VNS for the medium dataset

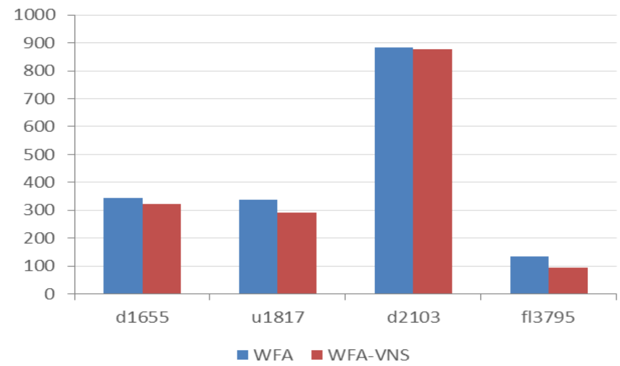

Fig. 7 Standard deviation comparison between WFA and WFA-VNS for large dataset.

We further compare our results with recent Hybrid WFA as shown in Table V. The methods in comparisons are: Hybrid water-flow algorithm and tabu search (WFA-TS) [8] and Hybrid water-flow algorithm and simulated annealing (WFA-SA) [24]. Table V shows that our approach is able to obtain competitive results with other hybrid WFA approaches in the literature.

TABLE V
WFA-VNS COMPARED TO OTHER RECENT WFA: WFA-TS AND WFA-SA

No	**Dataset**	**WFA-VNS**	**WFA-TS**		**WFA-SA**
			2-opt	**3-opt**	
1	berlin52	7,542	7,542	7,542	-
2	bier127	118,282	118,282	118,282	118,282
3	ch130	6,110	6,110	6,110	6,110
4	ch150	6,528	6,528	6,528	6,528
5	eil101	629	629	629	629
6	eil51	426	426	426	426
7	eil76	538	538	538	538
8	kroA100	21,282	21,282	21,282	21,282
9	kroA150	26,524	26,524	26,524	26,524
10	kroA200	29,368	29,368	29,368	29,368
11	kroB100	22,141	22,141	22,141	-
12	kroB150	26,130	26,130	26,130	-
13	kroB200	29,508	29,439	29,437	-
14	kroC100	20,749	20,749	20,749	-
15	kroD100	21,294	21,294	21,294	-
16	kroE100	22,068	22,068	22,068	-
17	lin105	14,379	14,379	14,379	-
18	lin318	42,225	42,155	42,126	42,029
19	rat575	6,900	6,869	6,854	6,843
20	rat783	9,113	9,021	8,967	8,942
21	u1060	229,639	228,298	229,613	228,321
22	fl1400	20,340	20,210	20,186	20,229
23	d1655	63,980	63,902	63,113	63,371
24	u1817	59,445	59,375	59,267	59,394
25	d2103	81,640	81,287	80,875	81,463
26	fl3795	29,350	29,348	29,333	29,151

IV. CONCLUSIONS

The aim of this paper is to investigate the potential of hybrid water-flow algorithm (WFA) with variable neighborhood search (VNS). The performance of the approach is tested on 26 established benchmarked symmetric TSP datasets. The result indicates that the WFA-VNS performance demonstrates improved result compare to a basic WFA in term of solution. WFA-VNS technically improved the solution produced in the population. However, WFA-VNS did not yield much improvement in terms of time but still there is an improvement. Meanwhile, in our further comparison with other algorithm and hybrid WFA algorithms, it appears that proposed algorithm showed a competitive result. We will make a further improvement in our neighborhood structures. The VNS part requires further revision and adjustment, which includes the method of executing neighborhood, whether to do it at random or by sequence. We will look deeper, sorting possibilities to improve the quality of the solutions including the strategy of reducing the time taken. The parameters aspects of WFA will be analyzed as well. For future work, the enhanced algorithm should be tested on different optimization problem. To conclude, it is worth to point out that our approach is successful in producing (near) optimal solutions than WFA-TSP. Given extra iteration, we are confident our algorithm is capable of finding better solutions. Thus, it has the credibility to solve difficult problems within this domain.

ACKNOWLEDGMENT

This research was funded by Ministry of Higher Education grant (FRGS/1/2016/ICT02/UKM/02/8).

REFERENCES

[1] Da Silva, R. F., & Urrutia, S. (2010). A General VNS heuristic for the traveling salesman problem with time windows. Discrete Optimization, 7(4).

[2] Frieze, A. M., Galbiati, G., & Maffioli, F. (1982). On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. Networks, 12(1), 23–39.

[3] Xiong, N., Wu, W., & Wu, C. (2017). An Improved Routing Optimization Algorithm Based on Travelling Salesman Problem for Social Networks. Sustainability, 9(6), 985.

[4] Lin, B. L., Sun, X., & Salous, S. (2016). Solving travelling salesman problem with an improved hybrid genetic algorithm. Journal of computer and communications., 4(15), 98-106.

[5] Xue-hui, L., Ye, Y., & Xia, L. (2008). Solving TSP with Shuffled Frog-Leaping Algorithm. In Proceedings - 8th International Conference on Intelligent Systems Design and Applications, ISDA 2008.

[6] Sakiyama, T., & Arizono, I. (2017). Can the Agent with Limited Information Solve Travelling Salesman Problem?. Complexity, 2017, Article ID 9562125, 6 pages.

[7] Papalitsas, C., Karakostas, P., Andronikos, T., Sioutas, S., & Giannakis, K. (2018). Combinatorial GVNS (General Variable Neighborhood Search) Optimization for Dynamic Garbage Collection. Algorithms, 11(4), 38.

[8] Bostamam, J. M., & Othman, Z. (2016). Hybrid water flow-like algorithm with Tabu search for traveling salesman problem. In AIP Conference Proceedings (Vol. 1761). American Institute of Physics.

[9] Bingüler, A. Hande Erol and Bulkan, Serol (2015). New Variable Neighborhood Search Structure for Travelling Salesman Problems. British Journal of Mathematics & Computer Science 6(5): 422-434, 2015, Article no.BJMCS.2015.088.

[10] Zia, M., Cakir, Z., & Seker, D. Z. (2018). Spatial Transformation of Equality–Generalized Travelling Salesman Problem to Travelling Salesman Problem. ISPRS International Journal of Geo-Information, 7(3), 115.

[11] Yang, X.-S. (2010). Nature-Inspired Metaheuristic Algorithms. Nature-Inspired Metaheuristic Algorithms Second Edition (p. 115).

[12] Dréo, J., Pétrowski, A., & Taillard, E. (2006). Metaheuristics for Hard Optimization. Vasa (p. 382).

[13] Alzaqebah, M., & Abdullah, S. (2015). Hybrid bee colony optimization for examination timetabling problems. Computers and Operations Research, 54, 142–154.

[14] Wang, X., Mu, A. & Zhu, S. (2013) ISPO: A New Way to Solve Traveling Salesman Problem. Intelligent Control and Automation, 4, 122-125.

[15] Yang, F. C., & Wang, Y. P. (2007). Water flow-like algorithm for object grouping problems. Journal of the Chinese Institute of Industrial Engineers, 24(6), 475–488.

[16] Wu, T.H., Chung, S.H., And Chang, C.C., "A Water Flow-Like Algorithm For Manufacturing Cell Formation Problems". European Journal Of Operational Research, Vol. 205, No. 2, pp. 346-360, 2010.

[17] Shahnazari-Shahrezaei, P., Tavakkoli-Moghaddam, R., Azarkish, M., & Sadeghnejad-Barkousaraie, A. (2012). a Differential Evolution Algorithm Developed for a Nurse Scheduling Problem. South African Journal of Industrial Engineering, 23(3), 68–90.

[18] Srour, A., (2014). Water flow-like algorithm for travelling salesman problem and web service selection problem," Ph.D thesis, University Kebangsaan Malaysia.

[19] Abdullah, S., Burke, E. K., & Mccollum, B. (2005, July). An investigation of variable neighbourhood search for university course timetabling. In The 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA) (pp. 413-427).

[20] Burke, E. K., Causmaecker, P. D., Petrovic, S., & Berghe, G. V. (2001). Fitness evaluation for nurse scheduling problems. Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), 2, 1139–1146.

[21] Hansen, P., & Mladenovic, N. (1997). Variable neighbourhood search. Computers and Operations Research, 24(11), 1097–1100.

[22] Crispim, J., & Brandao, J. (2001, July). Reactive tabu search and variable neighborhood descent applied to the vehicle routing problem with backhauls. In Proceedings of the 4th Metaheuristics International Conference, Porto (Vol. 1101, pp. 631-636).

[23] Crainic, T., & Centre for Research on Transportation (Montréal, Québec). (2003). Parallel variable neighborhood search for the p-median. Montréal: Groupe d'études et de recherche en analyse des décisions.

[24] Othman, Z.A., Al-Dhwai, NH, Srour, A., Yi, Wudi., "Water Flow-Like Algorithm with Simulated Annealing for Travelling Salesman Problems''. Int. J. Advanced Science, Engineering and Information Technology, Vol. 7, No. 2, pp. 669-675, 2017.

[25] Srour, A., Othman, Z. A., & Hamdan, A. R. (2014). A water flow-like algorithm for the travelling salesman problem. Advances in Computer Engineering, 2014, 1–14.

[26] Hansen, P., Mladenović, N., & Moreno Pérez, J. A. (2008). Variable neighbourhood search: Methods and applications. 4OR, 6(4), 319–360.

[27] Hansen, P., Mladenović, N., & Moreno Pérez, J. A. (2010). Variable neighbourhood search: Methods and applications. Annals of Operations Research, 175(1), 367–407.

[28] Chen, S., Chen, R., & Gao, J. (2017). A Modified Harmony Search Algorithm for Solving the Dynamic Vehicle Routing Problem with Time Windows. Scientific Programming, 2017, Article ID1021432, 13 pages

[29] Rincon-Garcia, N., Waterson, B., & Cherrett, T. (2017). A hybrid metaheuristic for the time-dependent vehicle routing problem with hard time windows. International Journal of Industrial Engineering Computations, 8(1), 141-160.

[30] Othman, Z. A., Srour, A. I., Hamdan, A. R., & Ling, P. Y. (2013). Performance water flow-like algorithm for TSP by improving its local search. International Journal of Advancements in Computing Technology, 5(14), 126.

[31] Cruz-Chávez, M. A., Martínez-Oropeza, A., del Carmen Peralta-Abarca, J., Cruz-Rosales, M. H., & Martínez-Rangel, M. (2014, June). Variable Neighborhood Search for Non-deterministic Problems. In International Conference on Artificial Intelligence and Soft Computing (pp. 468-478). Springer, Cham.

[32] Kralev, V. (2017). An Analysis of a Recursive and an Iterative Algorithm for Generating Permutations Modified for Travelling Salesman Problem. International Journal on Advanced Science, Engineering and Information Technology, 7 (5), 1685-1692.