EasyChair Preprint

№ 94

Whale Optimization Algorithm for Solving the Maximum Flow Problem

Raja Masadeh, Abdullah Alzaqebah and Ahmad Sharieh

April 25, 2018

WHALE OPTIMIZATION ALGORITHM FOR SOLVING THE MAXIMUM FLOW PROBLEM

RAJA MASADEH [1], ABDULLAH ALZAQEBAH [1], AHMAD SHARIEH [2]

[1]Computer Science Department, the World Islamic Science and Education University, Amman, Jordan

[2]Computer Science Department, the University of Jordan, Amman, Jordan

{raja.masadeh@wise.edu.jo , abdullah.zaqebah@wise.edu.jo, sharieh@ju.edu.jo }

ABSTRACT

Maximum Flow Problem (MFP) is deemed as one of several well-known basic problems in weighted direct graphs [9]. Moreover, it can be applied to many applications in computer engineering and computer science. This problem is solved by many techniques. Thus, this study presents a possible solution to the max flow problem (MFP) using a Whale Optimization algorithm (WOA), which is considered as one of the most recent optimization algorithms that was suggested in 2016. The experimental results of the "MaxFlow-WO" algorithm that were tested on various datasets are good evidence that the s technique can solve the MFP and reinforce its performance. It aims to solve the MFP by clustering the search space to find the MF for each cluster (local MF) in order to find the overall solution (global MF) of the desired graph. The WOA is used to solve the MFP by supposing the graph is the search space that the whales looking to reach the prey. Here, the prey is the sink in the network (represented by the graph) and other whales are represented in other nodes in the graph.

Keywords: Whale Optimization, Maximum Flow Problem, Maximum Flow, Meta-Heuristic, Optimization.

1. INTRODUCTION

In this study, we introduce a solution for the maximum flow problem based on a meta-heuristic approach called whale optimization algorithm Mirjalili et al [8]. The maximum flow problem (MFP) is considered as one of the most well-known issues of optimization in weighted directed network [29, 30]. Moreover, it can be applied to many applications such as networks, engineering, and transportations [18]. Many solutions to the MFP were presented and suggested by researchers using different techniques [9, 11]. The problem of MF is to determine the optimum solution for a directed and weighted graph; where the weight indicates to the flow capacity at each edge that communicates two vertices [29, 30]. Based on that, the main target is to obtain the maximum overall of flow from the source to the sink [29, 30].

A directed and weighted flow network is defined as G= (V, E), where V indicates to the set of vertices and E points to set of edges. Thus, a flow capacity (Cuv) is represented by the weight at each edge which is non-negative C (u, v) ≥0; where u and v belong to V [6]. Moreover, the network has two main vertices: the source and the sink. The source is the beginning vertex and the sink is the objective vertex [6]. Assume a graph G(V, E) contains a vertex called X, and all edges connected with vertex X are called E (X), suppose that F= max{Cuv: (u,v) ϵ E}. So the main issue is to get the optimal solution for a specific directed graph wherever every edge has a capacity. According to these constraints, the target is to get the maximum aggregate flow to the sink [7]. This scenario is mathematically modeled by equation (1) [6].

Where Xuv represents the flow on the edge (u, v).

$$Max\ f(x) = \sum_{(u,v)\in E(s)}^{n} X_{uv} \quad (1)$$

where

$$\sum_{\{v:(u,v)\in E\}} X_{uv} - \sum_{\{v:(v,u)\in E} X_{vu} = 0; For\ all\ u \in V\{s,t\}$$
$$0 \le X_{uv} \le F_{uv};\ For\ all\ (u,v) \in E.$$

The whale optimization algorithm is one of meta-heuristic optimization algorithms. These algorithms are becoming even more popular and play important role in many various applications because they are based on simple notion as they imitate simple concepts and ideas from the nature [1]. In addition, they are considered simple to implement and can be used in many various problems. Many meta-heuristic optimization algorithms are emerged last decades [8]. It is categorized into three main classes: evolutionary,

physics-based and swarm intelligence algorithms. Some studies utilized evolutionary meta-heuristic mechanisms which are inspired from the nature [2, 3], while other researchers used physics- based mechanisms that were imitated physical rules [4]. Whereas, others employed swarm intelligence algorithms were imitated the social intelligent attitude of swarms [5].

The whale optimization algorithm (WOA) imitates the humpback whales' behaviour. The mechanism imitates the hunting behaviour of this kind of whales in nature. The main phases of this behaviour are exploration phase searching for the prey, encircling the prey phase and finally exploitation phase which indicates to bubble-net method and attacking the prey.

The reminder of this paper is organized as follows. Section II illustrates related work. Section III describes the whale optimization algorithm and how it works. Section IV outlines the suggested algorithm "MaxFlow-WO". The experimental results are presented in Section V. Finally, Section VI draws the conclusion and future work.

2. RELATED WORK

Many researchers noticed that the Maximum Flow problem (MFP) includes many issues to be studied in different manners [9, 10]. The first method was used in 1956 and called Ford and Fulkerson method [28], which helps to get the Maximum Flow (MF) from a source to a destination by using augmenting path algorithm. [11]. A study of Dinic (1970) and Cormen et al. (2009) deduced that the algorithm's performance is O (mn) augmentation steps if each augmenting path in the shortest one; where the number of nodes is denoted as n and m indicates to the number of arcs [6, 12]. A close results to Dinic's algorithm where presented by Edmonds and Karp (1972) using new algorithm [12, 13]. The new algorithm conceded that the shortest path where the length of the arc equals one; such results were obtained using of Breadth First Search [14, 15]. As time goes on, more algorithms have been developed. A new algorithm was improved to compute the shortest augmenting path algorithm [15, 16].The study of Orlin (2013) on sparse network showed ameliorated polynomial time algorithm [17]. Moreover, the researcher presented a new solution of how solving the max-flow problem in O(mn); where m = O(n), as well as, solving the max-flow in O(nm + $m^{31/16}$ $\log^2$ n) time, where m= O($n^{1.06}$). An improvement of this algorithm is shown by King et al. (1994) who fixed the MFP in O(nm log m/(n log n) n) time.

In most graph problems, the genetic algorithm (GA) is rated as difficult algorithm to be applied in MFP, since the graph problems known with its various rare characteristics [18]. However, the genetic algorithm was applied in a weighted directed graph in order to get the MF from the source to the sink by [18]. In the previous study, each solution is executed by a flow matrix. There are two main features, in the fitness function - balancing nodes and the saturation rate of the flow. The first generation population is chosen randomly. After applying the GA, the resulted generation will be enhanced. After a particular number of iterations of applying genetic algorithm, the results will be optimal or near optimal.

Barham et al. [19] submitted chemical reaction optimization algorithm (CRO) for MFP in O(I E2) time, where I indicates to the number of iterations and E denotes to the number of arcs in flow graph. The CRO is suggested by Lam and Li [21] that is a meta-heuristic algorithm which was designed for fixing combinatorial optimization problems. The MaxFlow-CRO algorithm is offered to detect the best MF that can be transferred from the source to the sink in a flow network with no constraints for the capacity and violation where the flow in every arc rests within the upper bound value of the capacity.

Masadeh et al [20] presented grey wolf optimization (GWO) to solve the MFP in O(|N| + |E|2) time. GWO is suggested by Mirjalili [22], which is a meta- heuristics algorithm that is intended to resolve combinatorial optimization problems too [31, 32, 33]. The "Maxflow-GWO" algorithm is suggested in order to fix the MFP as similar to MaxFlow-CRO algorithm.

Surakhi et al [26] presented a parallel genetic algorithm (PGA) in order to solve MFP by calculating each augmenting path in the graph from the source to the sink in parallel stages during each iteration. PGA is implemented by using message passing interface (MPI) library. In addition; the study is implemented and conducted results from real distributed system IMAN1 supercomputer. Moreover; the study' results present good improvement in terms of speedup efficiency; whereas PGA speedup the running time by achieving up to 50% parallel efficiency.

Khanafseh et al [27] presented a comparison study between the performances of genetic algorithm (GA) and chemical reaction optimization (CRO) algorithm in solving MFP with the Ford-Fulkerson algorithm' performance which is a popular algorithm for solving MFP. In addition to that, the main objective of the

study is finding the suitable algorithm that will give outcomes closer to Ford-Fulkerson with same accuracy. Thus, the results of both algorithms that presented in this study can solve MFP with accuracy close to Ford-Fulkerson outcomes, but superior when applied GA in terms of time and accuracy.

3. WHALE OPTIMIZATION ALGORITHM

Whale Optimization Algorithm (WOA) is suggested by Seyedali Mirjalili and Andrew Lewis in 2016 [8]. Whale is chosen because it is considered as a highly smart animal with feeling. Whales are similar to human beings from the following aspects [23]. Whales have combined cells in specific parts of their brains comparable to those of human beings that named Spindle cells which are accountable for ruling, feeling and public attitude in human beings. Realize that the major reason of whales cleverness that they have twice quantum of these cells than an adult human. In addition; Whales can impart, understood, rule, communicate and sentimental similar to human but with less intelligent [8].

Humpback whale is one of the massive baleen whales, where their preferable preys are krill and small fish herds. The most motivating matter about humpback whales is their private hunting technique which is called bubble-net feeding technique [24].This kind of whales prefers to chase and hunt their preys that are near to the surface of the ocean. It has been observed that looking for preys is accomplished by making special bubbles along a circle or spiral shape as illustrated in Figure 1.

The mathematical model of the Whale optimization algorithm is presented in this section.

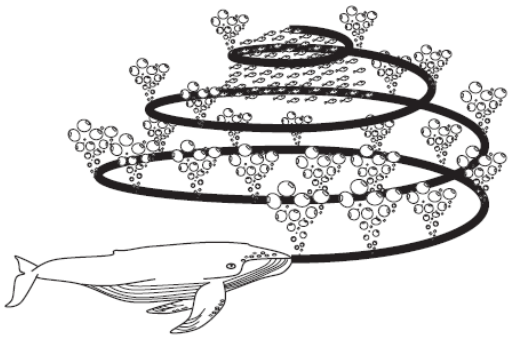

Figure 1: Bubble-net feeding behaviour of humpback whales [8].

3.1 Encircling Prey

Humpback whales can realize the location of prey and surround them by bubble-net feeding technique [24]. Since the location of the optimal value in the search space is unknown before, the WOA supposes that the best objective prey or is close to the optimum denote as the current candidate solution. When the best agent is determined, the other agents will change their positions at the best search agent. This manner is modelled by equations (2) and (3) [8].

$$\vec{D} = |\vec{C}.\overrightarrow{X*}(t) - \overrightarrow{X(t)}| \quad \text{..... (2)}$$

$$\vec{X}(t+1) = \overrightarrow{X*}(t) - \vec{A}.\vec{D} \quad \text{..... (3)}$$

Where (t) indicates the present iteration, X* denotes as the position vector of the best solution acquired so far, $\vec{X}$ is the position vector, | | denotes to the absolute values and the coefficient vectors denoted as A & C , and D is the distance that calculated in (2).

The vectors A and C can be computed as in equation (4) sand (5) respectively [8].

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \quad \text{..... (4)}$$

$$\vec{C} = 2.\vec{r} \quad \text{..... (5)}$$

Where $\vec{a}$ is linearly decreased from 2 to 0 over the range of iterations, and $\vec{r}$ is a random vector in [0, 1] according to [5].

3.2 Bubble-Net Attacking Method (Exploitation Phase)

Two approaches are prepared for the Bubble-net behaviour technique of the Humpback whales:

- Shrinking encircling mechanism: This manner is obtained by minimizing the value of a $\vec{}$ in equation (4). In this approach, the new position can be updated by the historical position and obtain the best one.

- Spiral updating position: the Humpback whales plunge about 12 meters then originate bubbles in spiral shape around the prey, then hunt them [8, 24]. This mechanism first computes the distance between the Humpback whale and the prey.

Then, a spiral equation is originated between the location of the prey and Humpback whale to imitate the spiral shape in equation (6) [8].

$$\vec{X}(t+1) = \vec{D} \cdot e^{bl} \cdot Cos\,(2\pi l) + \overrightarrow{X^*}(t) \quad \text{..... (6)}$$

Where $\vec{D}$ = |(X*) $\vec{t}$ - $\vec{X}$ (t)|, b is a constant for determining the form of the logarithmic spiral and l indicates to a random number in [−1, 1] according to [5].

It has been observed that the Humpback whales swim round the prey within a contracting range and along a helix shaped path at the same time. To shape this simultaneous behaviour, two probabilities are considered to choose from. The mathematical model is as in formula (7) [8].

$$\vec{X}(t+1) = \begin{cases} \vec{X}*(t) - \vec{A}.\vec{D} & if\ p < 0.5 \\ \vec{D} \cdot e^{bl} \cdot Cos\,(2\,\pi\, l) + \vec{X}*(t) & if\ p \geq 0.5 \end{cases} \quad \text{..... (7)}$$

Where p (probability) is a random number in range [0, 1].

3.3 Search For Prey (Exploration Phase)

Moreover to the bubble-net technique, the Humpback whales investigate for the prey randomly in the iteration. The mathematical model is as in functions (8) and (9) [8].

$$\vec{D} = |\,\vec{C} \cdot \overrightarrow{Xrand} - \vec{X}\,| \quad \text{..... (8)}$$

$$\vec{X}(t+1) = \overrightarrow{Xrand} - \vec{A}.\,\vec{D} \quad \text{..... (9)}$$

Where ($\vec{X}$) is a random position vector, ($\overrightarrow{Xrand}$) is the best search agent.

4. ALGORITHM "MAXFLOW-WO"

The whale optimization algorithm (WOA) is improved to resolve the maximum flow problem. In this study, the developed algorithm is theoretically analysed; and it is implemented and tested on various datasets. The run time of both the MaxFlow-WO algorithm and the Ford-Fulkerson algorithm are applied on the same datasets, and their performance is compared. To get the optimal flow for the maximum flow problem, the whale optimization algorithm is applied. Figures (2 – 6) present the suggested pseudo code for "MaxFlow-WO" algorithm.

MaxFlow-WO Algorithm (Main)

1- Initialize the whales population X_i (i = 1, 2, 3…… n) for n whales
2- Initialize C, r and a/ C is coefficient vector, r is random vector in [0, 1], and a is linearly decreased from 2 to 0 over the course of iterations.
3- Select the whale (source) and the prey (sink) randomly
4- Calculate the distance between each whale (i) and all X_{rand} by equation (1)
5- If whale (i) is not assigned
6- Assign whale (i) to its closet X_{rand}
7- Calculate the Fitness for each whale (i)
8- Invoke Cluster function
9- Global Max-Flow = Summation of all local max-flow
10- Return the best solution

Figure 2: Pseudo – Code for "MaxFlow-WO" algorithm

4.1 Initialization Stage

Figure 2 shows main program and initialization of the whales' population. One of these whales is chosen randomly for the source. However, before that, the prey is chosen at random too for a sink.

At the beginning, the distances between each whale and all X_{rand} are calculated, then the whale will designate the nearest X_{rand} and associate that cluster to be a member of its group.

4.2 Fitness Function

According to equation (10), X_{rand} has creates bubble-net, when it sees the prey, all whales in the search space (search agents) that catch sight of the bubble-net will join the cluster. Then, the other whales that designated the cluster should update their position towards the position of X_{rand} as presented in Figure 3. In this case, the position of any whale will be updated according to the value of A. If A is less than 1, then this whale is still joining this cluster and update its position by using equation (6). In other words, that whale is nearby to the prey. If A is greater than or equals to 1, then this whale will not join the cluster and it will look for another X_{rand} to join by using equation (9). This behaviour is represented mathematical by equations (6, 9 and 10).

$$\boldsymbol{V = h * r * \pi\,(\frac{1}{3})} \quad \text{..... (10)}$$

Where r is the radius of the bubble-net, h is the height of bubble-net, which is selected randomly between 6 and 12 according the whale's bubble-net behaviour in nature [24], this value represents the depth that the whale swam under

the prey [8], and π is constant value that is approximated to 3.14.

Fitness function to update whale's positions

1- X_{rand} creates Bubble-net by equation (10)
2- For each whale (i)
3- Update a, A, C and L
4- Calculate the distance between each whale (i) and X_{rand} by equation (8)
5- If (A<1)
6- Update the position of the current whale (i) by the equation (6)
7- Else If (A >=1)
8- Select new X_{rand} randomly
9- Update the position of the current whale (i) by the equation (9)
10- End If
11- End for

Figure 3: Fitness Function for update whale's positions to join in cluster

4.3 Clustering

Function for clustering the whales and calculate the local MaxFlow for each cluster

1- For each cluster K
2- Select X_{rand} randomly
3- While (I < = max_iteration)
4- Invoke the fitness function
5- Invoke the max_flow function
6- End While
7- Return the local max_flow for each cluster
8- End for

Figure 4: Cluster Function for clustering and calculating the local MaxFlow

As mentioned above, each cluster K has Xrand that chosen randomly, and after specifying the Xrand in the cluster, the fitness function will be computed for each whale to check if it sees the bubble-net from the Xrand. In other words, this whale will stay in this cluster and if not it will search for another Xrand to join. Thus, each cluster will compute its local maximum flow. And finally to get the global max-flow, calculate the summation of all locals max-flow as represented in Figure 4.

4.4 Maximum-Flow Function

As shown in Figure 5, each cluster will compute its local max-flow by invoking max-flow function. This function was introduced by Ford- Fulkerson [11, 28]; which depends on augmentation paths in residual graph in order to finding the maximum flow from source to sink node.

Maximum Flow Function

1- Select all paths from the whale (source) to the prey (sink)
2- For each edge (i, j) in the graph G
3- Flow of edge =0
4- While there is a path (p) from the whale to the prey in the remaining graph
5- Remaining_Capacity (p) = min (Remaining_capacity (i, j) for this edge in the path (p).
6- Flow= Flow + Remaining_Capacity (p)
7- For each edge (i, j) in the path (p)
8- If (i, j) is a forward edge
9- Flow (i, j) = Flow (i, j) + Remaining_Capacity (p)
10- Else
11- Flow (i, j) = Flow (i, j) – Remaining_Capacity (p)
12- End if
13- End for // For each edge (i, j) in the path (p)
14- End while
15- End for // For each edge (i, j) in the graph G
16- Return the maximum flow

Figure 5: Maximum- Flow Function

In the proposed method, each cluster will appears as a separate subnetwork and the maximum flow for it is calculated by using Ford- Fulkerson method. This method will return the maximum flow for that cluster and this value represents the local maximum flow for that cluster.

The clustering technique will result N number of clusters (K1….., N) and each cluster will find its local maximum flow, so the numbers of local maximum flows will be a result of the algorithm. Since the MFP tries to get the maximum flow of the network instead of a cluster, the term Global max-flow is used to represent the solution of Max Flow problem. Equation (11) represents the global max-flows where K is a cluster and N is the number of clusters.

$$\boldsymbol{Global\ Max\ Flow} = \sum_{k=1}^{N} (\boldsymbol{local\ max\ flow})_k \qquad \textbf{(11)}$$

A whale may leave the cluster X and belongs to another cluster Y in the same iteration. In this case, the path will be counted as a path for Y cluster.

For the initial MF of the population the computing effort is approximated by O(|V| |E| F), in line 1; where |V| represents the number of humpback whales (vertices), |E| indicates the number of edges between these whales and F represents the Maximum Flow value in the network. The complexity calculation for each function and method is shown as follows:

- The time complexity of the fitness function is O(|V|), in line 7, and the cluster function, in line

8, is O(|V|). The |V| indicates to the number of humpback whales (vertices).

• The time complexity of max-flow function is $O(|E|^2)$, in line 9. The while loop, in line 4 in maximum flow function, is repeated Fm times at most; where Fm points to the maximum flow. Furthermore O(|E|) is the cost of finding the augmenting path, in line 8 in maximum flow function. When the graph is complete, the Fm = |E|. Therefore, the run time complexity of Maximum Flow function is $O(|E| *Fm) = O(|E|^2)$.

• The total time complexity of MaxFlow-WO is $O(|V| + |E|^2)$.

5. EXAMPLE OF MAXFLOW – WO ALGORITHM

Let us clarify how the proposed approach is deployed the whale optimization algorithm for solving the MFP. The following example is based on the graph shown in Figure 6. A graph G has a set of nodes |V| and a set of edges |E| that represents a MFP from source to destination, for simplicity, the capacity for each edge was assumed as a random number. The aim of this example is to show how MaxFlow–WO algorithm works.

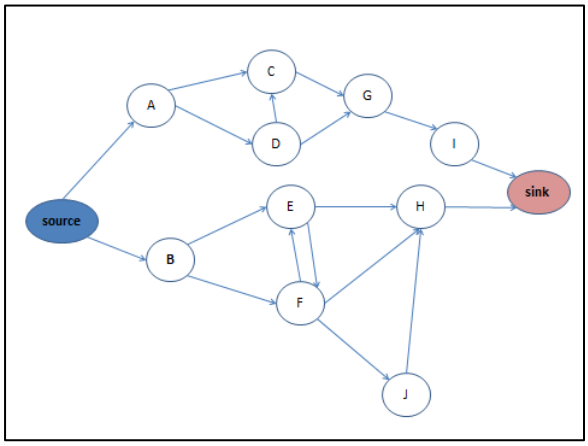

Figure 6: General Graph to represents the MFP

5.1 Initializing Phase

In this phase, the source and the sink were chosen randomly since the source represents the main whale and the sink is the prey. In Figure 6, it is clearly shown that the nodes A and B have the minimum distances to the source, so each one will represent a cluster as shown in Figure 7-(a). Each one of them will use the bubble net hunting technique and it is represented mathematically in equation (10) by using a cone shape. Figure 8-(b) shows the bubble net for each selected whales for each cluster.

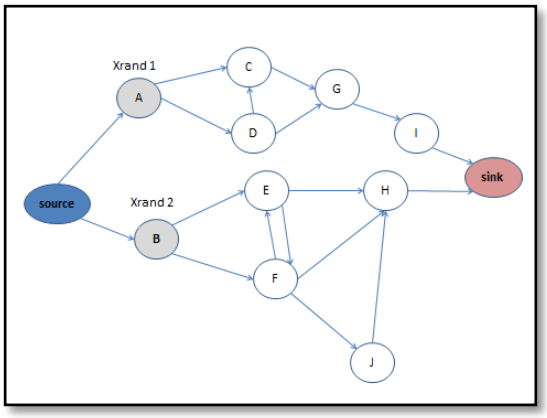

Figure 7: (a) - Choose the nearest node to the source to identify clusters.

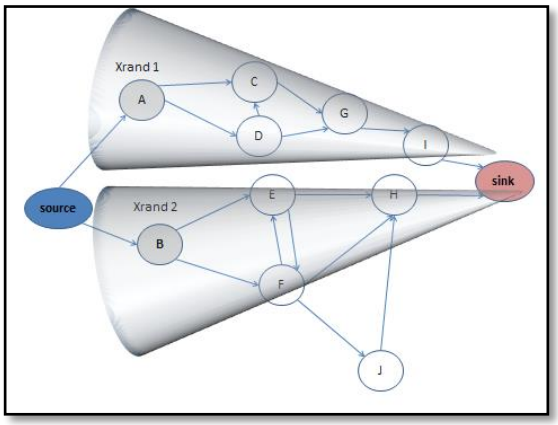

Figure 8: (b) - Each chosen node creates a bubble net to the prey (sink)

5.2 Clustering Phase

After the selected nodes create bubble net, it represents a clusters in the proposed work, each node will join the nearest cluster (inside the cone shape as shown in Figure 8-(b)) according to the fitness function. After this process, each node inside the cone shape will be in that cluster. The cone volume is calculated as in equation (10). Here, some nodes may be not joining any clusters as shown in Figure 9; node "J" was not joining any cluster.

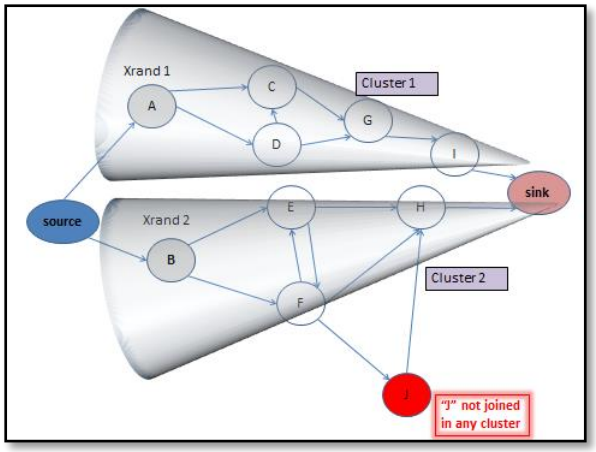

Figure 9: Clustered graph with a node out of bubble net (cluster)

Hence, any node that does not belong to any cluster, will update its position to join the nearest cluster in order to cover all nodes in the search space. In Figure 9 and Figure 10, node "J" has an empty cluster so it will update its position to join the nearest cluster, node "J" will join cluster number 2. The updated position and joining the nearest cluster is shown in Figure 11.

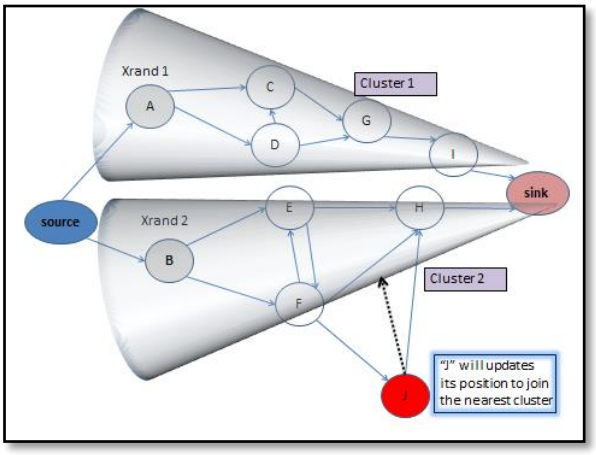

Figure 10: Clustered graph with a node out of bubble net (cluster) which looking for a nearest cluster

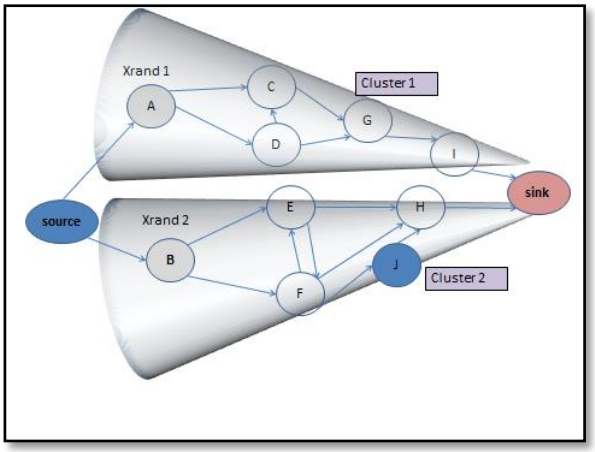

Figure 11: update position and joining the nearest cluster for empty cluster node.

5.3 MaxFlow -WO Algorithm Phase

After updating positions and clustering phase, each node in the graph will be in a specific cluster. In the proposed method, the MF will be calculated for each cluster from source to sink in order to find the MF for that cluster by using equation (1). The final result of calculation the max flow for each cluster is called the local Maximum Flow for that cluster.

This process will be repeated for each cluster, and the result of this process is a set of local MFs as a number of clusters. From this set of locals MF, the global MF will be found by using equation (11).

Figure 12 and Figure 13 show the MaxFlow-WO algorithm by calculation MF for each cluster in order to find the global MF for the all graph.

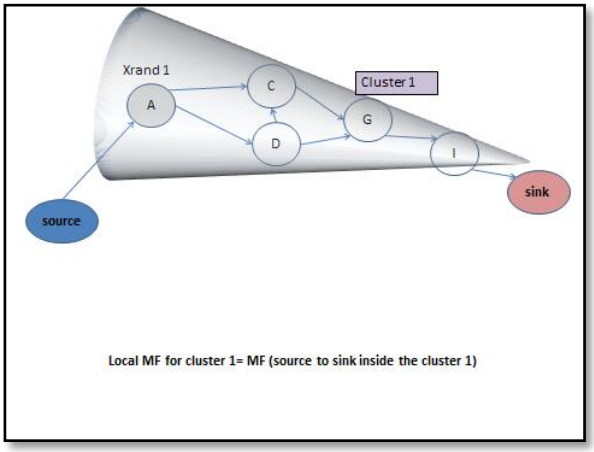

Figure 12: (a) - local MF for cluster 1

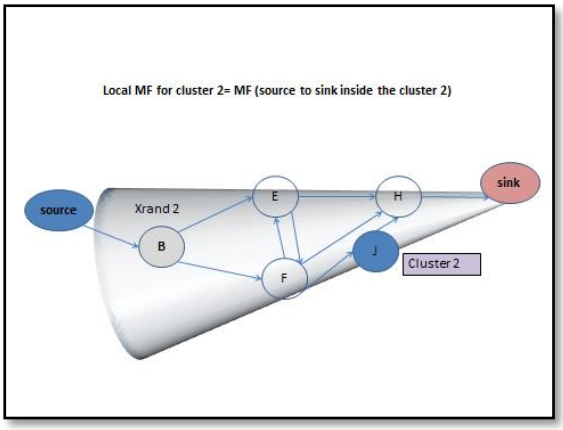

Figure 13: (b) - local MF for cluster 2

As shown in Figure 12 and Figure 13, the local MF was found for each cluster, and by using equation (11) the global MF will be found for that graph as in equation (12).

Global MF (G) = Local MF for cluster1 + Local MF for cluster2 …… (12)

6. EXPERIMENTAL RESULTS AND ANALYSIS

MATLAB based simulation program was developed to assess the performance of the MaxFlow-WO algorithm, by utilizing datasets of various network sizes. In addition, the sizes of the used datasets were between 50 and 1000 based on the prior studies [4, 5]. Every scenario was generated 10 experiments to have their average run time. The simulation program was tested on portable computer with following specifications: Intel (R) core (TM) i7-4510U CPU with 2.40 GHz, 16 GB RAM and Windows 8.1, 64-bit operating system.

6.1 Run time results

Table 1 illustrates the average run time in seconds for MaxFlow-WO algorithm compared to Ford-Fulkerson (FF) algorithm for various datasets. The MaxFlow-WO algorithm is faster than the FF algorithm for dataset of size 500 nodes by 7.69 times, for example.

It is obvious from Figure11 that the run time complexity is quadratic polynomial. In other words, it increases with the number of vertices in the graph. Table 2 shows the Theoretical and experimental run time of MaxFlow-WO with relative error rate, and this rate was calculated using equation (13).

Relative error = abs (experimental run time – analysis run time)/abs (analysis run time) …… (13)

Ford-Fulkerson algorithm [25] is selected in this study, in order to compare its performance with the performance of the suggested algorithm as shown in Figure 14 and Figure 15. Furthermore, the same specifications were utilized for the suggested algorithm as presented in Table 1. As mentioned above, it was reiterated 10 experiments for every dataset. Thus, according to the comparison of both algorithms results, it is obvious from Figure 12 that the suggested algorithm accomplished better performance, especially for networks with large sizes. Figure 16 shows the performance enhancements that had been accomplished in this research compared with Ford-Fulkerson algorithm.

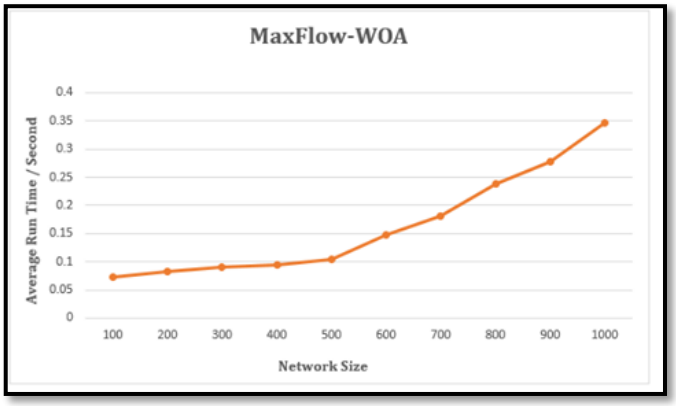

Figure 14: Average running time for experimental results for MaxFlow-WO algorithm

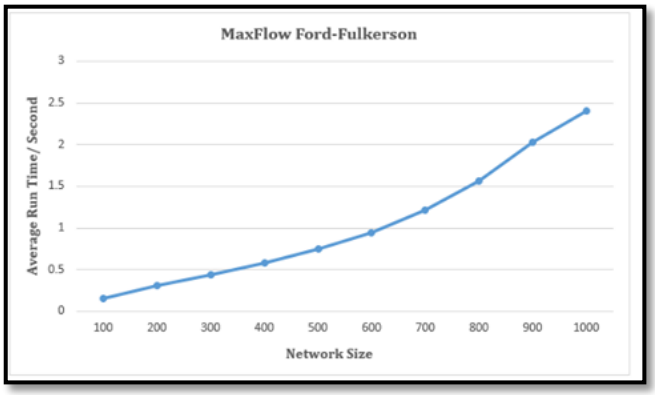

Figure 15: The experimental results for Ford-Fulkerson Algorithm

Table 1: Run time of MaxFlow-WO compared to Ford-Fulkerson for various datasets

Network Size	Average run times/ Seconds (Ford-Fulkerson)	Average run times/ Seconds (MaxFlow-WO)	MaxFlow-WO faster than FF
50	**0.06721**	**0.05783**	1.16
100	**0.15626**	**0.0719**	2.17
150	**0.21564**	**0.07658**	2.78
200	**0.31252**	**0.08282**	3.7
250	**0.36408**	**0.08751**	4.17
300	**0.43283**	**0.09063**	4.76
350	**0.49376**	**0.09377**	5.0
400	**0.57502**	**0.09533**	5.88
450	**0.7**	**0.09689**	7.14
500	**0.75002**	**0.1047**	7.69
550	**0.8469**	**0.12032**	7.14
600	**0.94219**	**0.14844**	6.25
650	**1.09063**	**0.15939**	6.67
700	**1.2109**	**0.18127**	6.67
750	**1.38908**	**0.22033**	6.25
800	**1.56563**	**0.23908**	6.67
850	**1.72502**	**0.25002**	7.14
900	**2.02657**	**0.27814**	7.14
950	**2.16719**	**0.30939**	7.14
1000	**2.40099**	**0.34621**	7.14

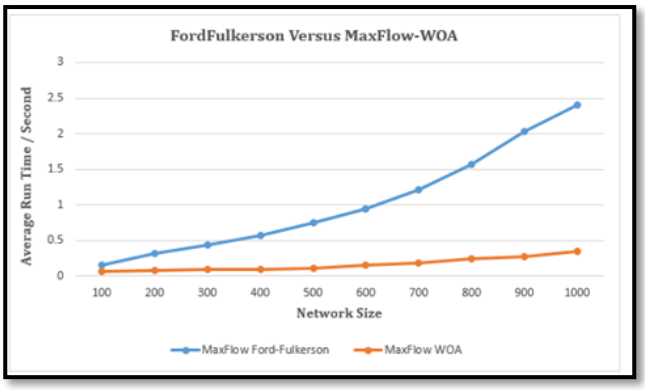

Figure 16: The experimental results for Ford-Fulkerson versus MaxFlow-WOA Algorithms

Table 2: Theoretical and experimental Run Time of MaxFlow-WO algorithm

Size of Dataset	Estimated Theoretical run time (T)	Estimated experimental run time (E)	Error =Abs (T-E)	Relative Error (%)
100	0.01127	0.0719	0.060	5.38
200	0.069344	0.08282	0.013	0.19
300	0.31608	0.09063	0.223	0.71
400	0.81856	0.09533	0.723	0.88
500	1.65506	0.1047	1.550	0.94
600	3.04627	0.14844	2.898	0.95
700	4.26164	0.18127	4.080	0.96
800	5.99504	0.23908	5.756	0.96
900	8.14552	0.27814	7.867	0.97
1000	9.99965	0.34621	9.653	0.97
Average			3.283	1.29

6.2 Results accuracy

Two main factors are taken into account in order to compare between FF algorithm and MaxFlow-WO algorithm; average run time and the accuracy of the results. Average run time comparison is mentioned above. While comparing the two algorithms in terms of accuracy of the results; the MF value was computed for both algorithms in experiments in order to show the accuracy of MaxFlow-WOA algorithm compared with Ford-Fulkerson's MF values on various data sets.

Table 3 shows the MF values for both algorithms with the accuracy between Ford-Fulkerson's value and MaxFlow-WOA's value for each data set that was computed using Equation (14).

$$Accuracy = \left(1 - \frac{|FF_{MFvalue} - MaxFlowWOA_{MFvalue}|}{FF_{MFvalue}}\right) * 100 \quad \ldots\ldots\ldots\ldots (14)$$

Where FF_MFvalue indicates the MF value that calculated using Ford-Fulkerson algorithm, MaxFlowWOA_MF value represents the MF value by using proposed algorithm and the result value is the accuracy of proposed algorithm result compared with Ford-Fulkerson algorithm for each size of graph in the data set.

In order to calculate the overall accuracy of the proposed algorithm; the average accuracy for all graph sizes in the data set was computed using Equation (15).

$$Overall\ Accuracy = \left(\sum_{i=1}^{N} Accuracy(N)\right)/N \quad \ldots\ldots\ldots\ldots\ldots\ldots (15)$$

Where N indicates to the number of graph sizes in the data set, accuracy (N) calculated as Equation (14).

As clearly shown in Table 3; the overall accuracy of the proposed algorithm is 93.8%, since the difference between the MF values for MaxFlow-WOA algorithm and Ford-Fulkerson algorithm comes from the way of taking the graph in the search space, since the proposed algorithm split the graph into subgraphs to works with. Thus, it is possible to find an edge that connects a node (whale) form one subgraph to another subgraph whereas this displays the reason of difference.

Table 3: Results Accuracy for MaxFlow-WOA

Nodes	Ford-Fulkerson MF (E)	MaxFlow-WOA MF (F)	|E - F|	|E - F|/E	Accuracy 1-(|E - F|/E)
50	305	352	47	0.15	84.6
100	621	629	8	0.01	98.7
150	893	883	10	0.01	98.9
200	1195	1261	66	0.06	94.5
250	1592	1512	80	0.05	95.0
300	1865	1982	117	0.06	93.7
350	2206	2234	28	0.01	98.7
400	2467	2628	161	0.07	93.5
450	2844	2971	127	0.04	95.5
500	3162	2990	172	0.05	94.6
550	3402	3536	134	0.04	96.1
600	3710	4148	438	0.12	88.2
650	4168	4174	6	0.001	99.9
700	4538	4635	97	0.02	97.9
750	4711	5217	506	0.11	89.3
800	5136	5676	540	0.11	89.5
850	5471	6034	563	0.10	89.7
900	5744	6485	741	0.13	87.1
950	6058	6593	535	0.09	91.2
1000	6408	6384	24	0.00	99.6
Over all Accuracy					**93.8**

7. CONCLUSION AND FUTURE WORK

This paper introduces a whale optimization algorithm, named MaxFlow-WO, to fix the maximum flow problem. The algorithm is explained, implemented, and tested on datasets of sizes 50, 100, 150, …,1000 nodes. For the initial MF of the population the computing effort is evaluated to be O(N|E| F), where N denotes as the number of whales (vertices), |E| indicates to the number of edges between whales and F represents the Maximum Flow in the graph. The experimental run time is acquired as quadratic polynomial on the experimented various datasets. Thus, the time complexity of MaxFlow-WO is $O(N + |E|^2)$. The MaxFlow-WO outperforms the Ford-Fulkerson algorithm for fixing the maximum flow problem in run time. In addition to that, the accuracy of the results is another main factor that is used in order to compare between FF algorithm and MaxFlow-WO algorithm, whereas the overall accuracy of the proposed algorithm is 93.8%.

As a future direction, MaxFlow-WO can be improved to acquire better performance by implementing it on parallel computer. Furthermore, a comparison between this suggested algorithm and other metaheuristics which are used to fix the maximum flow problem could be investigated.

REFERENCES

[1]: Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. science. 1983 May 13;220(4598):671-80.

[2]: Holland JH. Genetic algorithms. Scientific american. 1992 Jul 1;267(1):66-73.

[3]: Shaheen A, Sleit A. Comparing between different approaches to solve the 0/1 Knapsack problem. International Journal of Computer Science and Network Security (IJCSNS). 2016 Jul 1;16(7):1.

[4]: Alatas B. ACROA: artificial chemical reaction optimization algorithm for global optimization. Expert Systems with Applications. 2011 Sep 15;38(10):13170-80.

[5]: Dorigo M, Birattari M, Blum C, Clerc M, Stützle T, Winfield A, editors. Ant Colony Optimization and Swarm Intelligence: 6th International Conference, ANTS 2008, Brussels, Belgium, September 22-24, 2008, Proceedings. Springer; 2008 Sep 20.

[6]: Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to algorithms mit press. Cambridge MA. 1990.

[7]: Goldberg AV, Tarjan RE. A new approach to the maximum-flow problem. Journal of the ACM (JACM). 1988 Oct 1;35(4):921-40.

[8]: Mirjalili S, Lewis A. The whale optimization algorithm. Advances in Engineering Software. 2016 May 31;95:51-67.

[9]: Tarjan RE. Data Structures and Network Algorithms. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for industrial and applied mathematics. 1983;33.

[10]: McHugh JA. Algorithmic graph theory. New Jersey: Prentice Hall; 1990.

[11]: Ford LR, Fulkerson DR. Maximal flow through a network. Canadian journal of Mathematics. 1956 Feb;8(3):399-404.

[12]: Dinits EA. Algorithms for solution of a problem of maximum flow in a network with power estimation. Soviet Math. Cokl.. 1970;11:1277-80.

[13]: Edmonds J, Karp RM. Theoretical improvements in algorithmic efficiency for network flow problems. Journal of the ACM (JACM). 1972 Apr 1;19(2):248-64.

[14]: Eppstein D. Finding the k shortest paths. SIAM Journal on computing. 1998;28(2):652-73.

[15]: Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms, the Massachusets Institute of Technology. Cambridge, MA, USA,. 2001.

[16]: Nemhauser GL, Wolsey LA. Chapter vi integer programming. Handbooks in Operations Research and Management Science. 1989 Dec 31;1:447-527.

[17]: Orlin JB. Max flows in O (nm) time, or better. InProceedings of the forty-fifth annual ACM symposium on Theory of computing 2013 Jun 1 (pp. 765-774). ACM.

[18]: Munakata T, Hashier DJ. A Genetic Algorithm Applied to the Maximum Flow Problem. InICGA 1993 Jul 1 (pp. 488-493).

[19]: Barham R, Sharieh A, Sliet A. Chemical Reaction Optimization for Max Flow Problem. International Journal of Advanced Computer Science And Applications. 2016 Aug 1;7(8):189-96.

[20]: Masadeh R, Sharieh A, Sliet A. Grey wolf optimization applied to the maximum flow problem. International Journal of Advanced and Applied Sciences. 2017 Jul 1;4(7):95-100.

[21]: Lam AY, Li VO. Chemical-reaction-inspired metaheuristic for optimization. IEEE Transactions on Evolutionary Computation. 2010 Jun;14(3):381-99.

[22]: Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Advances in Engineering Software. 2014 Mar 31;69:46-61.

[23]: Hof PR, Van Der Gucht E. Structure of the cerebral cortex of the humpback whale, Megaptera novaeangliae (Cetacea, Mysticeti, Balaenopteridae). The Anatomical Record. 2007 Jan 1;290(1):1-31.

[24]: Watkins WA, Schevill WE. Aerial observation of feeding behavior in four baleen whales: Eubalaena glacialis, Balaenoptera borealis, Megaptera novaeangliae, and Balaenoptera physalus. Journal of Mammalogy. 1979 Feb 20;60(1):155-63.

[25]: Chintan J, Garg DG, Goel SG. An Approach to Efficient Network Flow Algorithm for Solving Maximum Flow Problem (Doctoral dissertation).

[26]: Surakhi OM, Qatawneh M, Hussein A. A Parallel Genetic Algorithm for Maximum Flow Problem. International Journal of Advanced Computer Science and Applications. 2017.

[27]: Khanafseh MY, Surakhi OM, Sharieh A, Sleit A. A Comparison between Chemical Reaction

Optimization and Genetic Algorithms for Max Flow Problem. International Journal of Advanced Computer Science and Applications. 2017 Aug 1;8(8):8-15.

[28]: http://www.geeksforgeeks.org/ford-fulkerson-algorithm-for-maximum-flow-problem/ , last visited 12/1/2018

[29]: https://www.topcoder.com/community/data-science/data-science-tutorials/maximum-flow-section-1/ , last visited 12/1/2018

[30]: https://www.topcoder.com/community/data-science/data-science-tutorials/maximum-flow-section-2/, last visited 12/1/2018

[31]: Masadeh, R., Alzaqebah, A., Hudaib, A., & Rahman, A. A. (2018). Grey Wolf Algorithm for Requirements Prioritization. Modern Applied Science, 12(2), 54. ISO 690

[32]: Hudaib, A., Masadeh, R., Qasem, M. H., & Alzaqebah, A. (2018). Requirements Prioritization Techniques Comparison. Modern Applied Science, 12(2), 62.

[33]: Yassien, E., Masadeh, R., Alzaqebah, A., & Shaheen, A. (2017). Grey Wolf Optimization Applied to the 0/1 Knapsack Problem. International Journal of Computer Applications, 169(5).