Optimizing RIP Routing System using Artificial Neural Network

Saven Hasan Rasool

Department of Computer Science and

,IT, Collage of Science

Salahaddin University-Erbil

Erbil, Iraq

saven.rasool@su.edu.krd

Hangaw Qader Omar

Department of Computer and Network Paytakht Technical Institute

Erbil, Iraq

hangaw.omar@pti.edu.krd

Ferman Ibrahim Kareem

Department of legal administration , Erbil technical administrative institute,Erbil Polytechnic University

Erbil , Iraq

Ferman.kareem@epu.edu.iq

Abstract— Network routing is still considered one of the critical factors affecting overall network efficiency, leading to the invention of many dynamic routing protocols such as RIP, EIGRP, and OSPF. Routing Information Protocol (RIP) is one approach that depends on the hop count, which is the number of routers between the source and destination. The RIP system finds the least expensive next hop for a network to determine the best path in a dynamic routing system, but it does not consider metrics or the physical environment of router devices. Metrics such as delay, bandwidth, and reliability of the router can affect overall packet tracing.

This paper proposes a solution that uses an artificial neural network to contribute to these metrics. Neural networks automatically weigh factors like delay and bandwidth as inputs to an artificial neuron provided for each router to generate a heuristic value, and use reliability as a bias for the neuron. The heuristic value is then added to a routing distance table of all the neighborhood nodes of the router. The optimized Bellman-Ford algorithm is then run to produce the routing table for the entire network.

This results in a more optimized and balanced network that takes into account four network metrics (hop count, bandwidth, delay, and reliability), which makes the network more sensitive to certain environmental attributes.

Keywords— (Artificial Neural Networks; Optimization; Algorithms; RIP)

I. Introduction

With the rapid development and increasing use of the Internet in almost all aspects of life, there is a need for a faster network with more capacity for new Internet applications. Some traditional network technologies encounter bottlenecks when it comes to meeting the growing demand for network usage. In a computer network, data transmission is based on the routing protocol, which selects the best route between any two nodes [1]. A routing protocol is a language that routers use to communicate with each other and share information about the reachability and status of the network. It includes a procedure to determine the best path based on its reachability information and record this information in a routing table. A routing metric is used to select the best path, and a routing algorithm computes it [2].

Different types of routing protocols are used for specific network environments. Three types of dynamic routing protocols are commonly used for a specific network installation: RIP, OSPF, and EIGRP. Routing Information Protocol (RIP) is one of the earliest routing protocols still in use. It is a distance vector protocol that uses hop count as its primary metric. The network size that RIP can support is limited, while the Open Shortest Path First (OSPF) protocol is based on bandwidth and the Enhanced Interior Gateway Routing Protocol (EIGRP) is based on bandwidth, load, and delay, ignoring the hop count [3]. In this case, there is a need for AI algorithms to produce a better routing solution through learning in the field of computer networking. Network routing, which determines the route taken by packets from the source to the destination, plays a critical role in networking by selecting the path of packet transmission. It is the main factor in increasing network efficiency and using network parameters such as delay, hop, load, bandwidth, etc. to balance the load on the network [4].

An artificial neural network (ANN) is a collection of processing elements called neurons that are interconnected with one another. These patterns of neurons work together to solve complex problems that require learning or periodically changing states. When trying to solve optimization problems, it is important to have tools such as methods that can successfully minimize or maximize objective functions. It is often necessary to approximate a solution in order to solve a problem involving one of these functions, especially when the function is neither linear nor polynomial. Some methods use the application of entire or partial derivatives to linearize these functions at certain points [5, 6]. The approximation of the objective function allows for the use of various approaches for artificial intelligence, such as non-linear regression, to solve optimization problems. If the derivative of the objective function can be expressed as a polynomial, then it is possible to compute the solution to an optimization problem. AI algorithms are often used to improve factors such as weights, network design, learning rules, neuron activation function, and bias on the metrics of each router to create a single heuristic value for each router.

The heuristic value is a method that involves adopting an approach to a problem based on some additional factor or criteria, or based on previous solutions to similar problems.

We propose using neural networks to optimize the RIP routing system. Our approach allows for the automatic discovery of cost functions, eliminating the need for custom heuristics. By effectively using context information and describing complicated trajectory features, this method combines the benefits of using weight strategies for input with the current distance vector used in standard RIP.

We use a hybrid model that considers both of these aspects in the computation of the cost of evaluation. The efficiency and robustness of the proposed model have been demonstrated through multiple runs on different datasets.

II. Literature Review

Numbers of Artificial Intelligence and Neural Network schemes have been proposed to optimize computer networks and the following are some of the papers that have been studied in this regard.

This paper [7] suggests a novel training method based on the previously proposed Whale Optimization Algorithm (WOA). It has been demonstrated that this method outperforms the existing algorithms and can tackle various optimization issues. For the first time in the literature, a set of 20 datasets with varying degrees of complexity are selected to evaluate the suggested WOA-based trainer.

In [8] the researchers said that the comparative analysis reveals that IMBO consistently outperforms existing algorithms and offers highly competitive outcomes. It is used to train neural networks as well, illustrating how useful IMBO is for resolving difficult real-world issues. 15 well-known classification datasets from the University of California, Irvine (UCI) Machine Learning Repository are used to evaluate the IMBO-based trainer. The outcomes are contrasted with a number of methods from the literature, such as the original MBO and GCMBO. It has been found that IMBO considerably enhances neural network learning, demonstrating the algorithm's benefits for taking on difficult issues.

In [9] the performance of the proposed evolving neural network is benchmarked using 13 classification datasets, three function approximation datasets, and one real-world case study (Tennessee Eastman chemical reactor challenge). The results demonstrate that the suggested strategy delivers extremely competitive performance compared to well-known conventional and evolutionary trainers.

In [10] the mechanism for artificial intelligence enabled routing (AIER) with congestion avoidance in SDN is presented. It can not only minimize the impact of monitoring ranges with dynamic routing but also provide learning capability and superior route choices by trying to introduce artificial intelligence (AI) technology. By integrating three more modules—topology discovery, monitoring interval, and an artificial neural network—into the control plane, researchers could assess the overall efficiency of the proposed AIER mechanism on the Mininet simulator. Performance evaluation, including average throughput, packet loss ratio, and packet delay vs data rate for various monitoring times in the system, show the effectiveness and superiority of our suggested AIER mechanism.

In [11], the research suggests a routing mechanism built on reinforcement learning that takes the benefit of node information to generate a stable and fast route. The proposed approach aims to improve throughput by reducing energy use, transmission latency, and packet delivery. By validating the presented method's performance in the based simulation II in terms of energy consumption, transmission latency, and packet delivery ratio, its utility is assessed.

In [12], the paper proposed a delay and energy well-organized machine learning based routing protocol for P2P networks, which permits the network engineers to expand upon the network QoS without compromising on the routing efficiency of the network. The outcomes show that the suggested procedure is enhanced in terms of delay and energy efficiency compared to the existing ones. It also affords another routing path in case of node or link-down circumstances. They also suggest some additional work which can be taken up by using the protocol to make the system more protected and efficient in terms of complete privacy.

III. methods

A. *Artificial Neural Network:* A content analysis of the relevant literature was performed in order to present, identify, analyze, categorize, and review the distinguished ANN-based optimization strategies for various applications and, more specifically, in RIP routing. This was done in order to demonstrate, discover, assess, and review the strategies[13].

B. *Routing Information Protocol (RIP*) is a distance vector protocol that uses hop count as its primary metric. RIP specifies how routers should exchange information while transferring traffic among a collection of linked local area networks (LANs)[14]. In the distance-vector routing, routers learn the routing information from directly connected neighbors, and these neighbors may have learned these networks from other neighboring routers. As result, the distance-vector routing is also known as the routing by rumor.

C. Distance Vecor Algorithm / Bellman-Ford

Each router notifies a list of distance-vectors (route with cost) to each neighbors periodically. Every router selects the route with smallest metric. Metric is a positive integer– The cost to reach a destination: number of hops– Hop-count is limited to 1-15, 16 is “infinity” [15] figure 1 illustrates an example of a specific network that consists of 6 vertices (nodes) each indicate a router with a specific properties and metrics, and 7 edges that indicate the cost of transmitting routed package in the network.

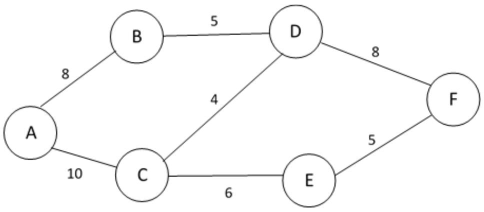

Figure 1 a specific network that consists of 6 routers

Such network has a 6 different routing tables for each router. Table 1, illustrates the routing table of vertex A, (router A) that indicate the neighbor costs and next hop suggestion.

Table 1: The routing table of node A is

Distention	Cost	Next Hop
A	0	
B	8	B
C	10	C
D	Infinity	
E	Infinity	
F	Infinity	

The RIP algorithm states that B will be investigated since it has the lowest cost, regardless of other elements that impact packet tracing, such as delay, bandwidth, reliability, and other metrics. This may lead to a bottleneck as B may has the lowest bandwidth or worst reliability due to power shortage in the site or suffering from overheating [14].

D. **Metrics that affect routing**

1-Hop is a metric value used to measure distance based on the number of network datagram traverses. Each time a router forwards a datagram onto a segment, this counts as a single hop. The RIP makes a decision depending on it.

2-Bandwidth is measured in terms of bits per second. Links that support higher transfer rates like gigabit are preferred over lower capacity links like 768kbps for instance, the OSPF make a decision depending on it.

3- Delay is measured in tens of microseconds and denoted by the symbol µ. Delay represents the amount of time it takes for a router to process, queue, and transmit a datagram out an interface.

4- Reliability although this metric may be configured as a fixed value by an administrator, it is generally measured dynamically over a specific time frame, such as five seconds. Routers observe attached links, reporting problems, such as link failures, interface errors, lost datagrams and so on.

5- There are some other metrics like Load, gutter, path length, packet loss, etc. that may affect the routing system and overall network efficiency but adding all these factors to a routing protocol may require extra time to calculate all these numbers and functions; hence it will be very time consuming which causes of increasing the delay [16, 17].

IV. the system implementation

The proposed system generates a heuristic value by using the main metrics of the routing system and combining it with the hop count used mainly in the RIP system through an artificial neuron. The weighted input periodically allows changes in the impact of metrics on the routers and the overall system.

A. *ANN Optimization*

Neural networks can analyze vast amounts of data with various sophisticated properties in a very short time and discover various patterns. [11, 13]. The neural networks modelling technique is not dependent on methods that are founded in physics, which makes it not only versatile but also quick when it comes to find solutions to problems. They are able to incorporate the ANN structural model without much difficulty and can be adjusted based on the physical attributes of the operator. Neural networks are valuable tools for resolving complex non-linear interactions since the inputs they use are stored within the networks themselves rather than in a database line. The system proposes a neuron that is explained in figure 2, which accepts two of the main metrics as weighted input.

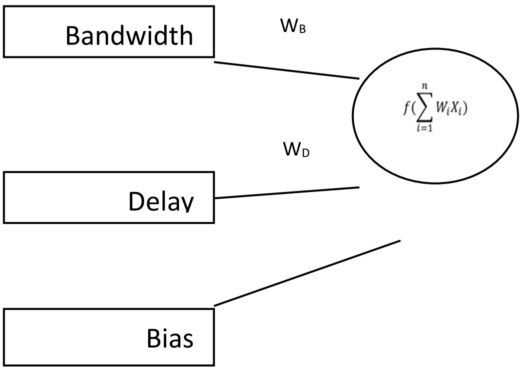

Figure 2 proposed Neurons to associate with each router

The input of the neuron is both bandwidth, delay and bias, these router metrics affect the routing of packages within the network very sharply. The neurons' weights are considered as 0.886 for bandwidth and -0.62 for the delay, as it is a correlation between these two metrics and the routing instability [18]. Table 3 shows the default bandwidth of some routers in kilobit per second, these numbers used by this paper is not all available bandwidths, but it's a sample of the most used one and taken as an input of the proposed neuron.

Table 3 the sample of most used bandwidth

Band width	Kbps
Gigabit Ethernet Interface (1 Gbps)	1000000
Fast Ethernet Interface (100 Mbps)	100000
Ethernet Interface (10 Mbps)	10000
DS1 (1.544 Mbps)	1544
3 Kbps)	

Delay is the second metric that affect the network routing and used an input of the neuron, this paper considers that delay is ranged between 10 to 76 microseconds.

The input of both metrics input value must be mapped into the same range; hence the maximum bandwidth is 1000000 and the maximum delay is 76 microseconds. Multiplying these numbers (1000000 and 76) by their weights will cause a huge unbalancing of the result. Here the following formula is used to map the current input number into arranging between [1 and 3].

$X= (input-r_{min})/(r_{max}-r_{min})\ x(t_{max}-t_{min})\ +t_{min}$

Where r_{min} and r_{max} are the minimum and the maximum numbers of the metrics and t_{min} and t_{max} are the minimum and the maximum number of the new map.

Bandwidth [1000000, 768] will be mapped to [3, 1]

And the delay [76, 10] will be mapped to [3,1].

Applying the formula of neural network mentioned in figure 2 is as follows:

Net. Weigh = $X_C*W_C+X_D*W_D$ + Bias

By applying the formula on the case of the best given metric (minimum delay and maximum bandwidth) and worst given metrics case (maximum delay and minimum bandwidth) table 4 and table 5 can be observed.

Table 4 Applying worst inputs

Metrics	Value	Input	Net. Weight
Bandwidth	768 kbps	X_C= 1	-0.974
Delay	76 Microsecond	X_D= 3	

Table 5 applies the best case input

Metrics	Value	Input	Net. Weight
Bandwidth	1000000 kbps	X_C= 3	2.038
Delay	10 Microsecond	X_D= 1	

The best case Net weight generated by the neuron is (2.038) and the worst case Net weight generated by the neuron is (-0.974), which are ready to pass to the activation function. Table 6 shows the use of several none linear activation functions in the neuron and their results.

Table 6 different nonlinear activation functions

Net. Wight	Best case	2.038
	Worst case	-0.974
Sigmoid	Best case	0.734777766
	Worst case	0.380600545
Tan Inverse	Best case	1.11463478756195
	Worst case	-0.772227699034323
Tan Hyperbolic	Best case	0.966616235206401

	Worst case	-0.750456796738985

As shown in table 6, the Net weight has a better fit to the distance vector provided in figure 1, which is approximately between -1 and 2 which makes this paper proposes this number after adding a bias.
This research proposes the reliability metric to generate a net weight which has a better performance and better fit to
the distance vector, so the reliability also must map from the current range of [0, 255] to a new range [1, 3] by the same mentioned formula for bandwidth and delay.
A Heuristic value Solution for RIP Routing
The problem of determining the most efficient route for data packets is a search problem that can be modeled as a graph. In the context of this discussion, the network of nodes is modeled as a graph, and the question at hand is how to determine the path that leads from a specific origin node to a specific destination node.
A new routing table will be generated by adding the generated heuristic value from the proposed neural network to the routing table. Let's apply on the following case:
From figure 1, let's assume that router B has a delay of 69 microseconds, the reliability is 40 and it uses an Ethernet Interface (10 Mbps) the result is 0.487612319 which can approximately be set as 0.49.
Furthermore, assume that router C has a delay of 25 microseconds, the reliability is 220 and it uses a Fast Ethernet Interface bandwidth (100 Mbps), the result is 2.885646266 which approximately can set as 2.89.
The two heuristic values are now used to modify the routing table of router B and router C neighbor's using the following heuristic function:
$f(n) = g(n) + h(n)$
Where the f(n) is the next hop cost, g(n) is the current cost of the next hop and h(n) is produced value from the last available metrics updated from the neighbor node. Here the lowest cost is the better effect, which makes the following formula used instead.
$f(n) = g(n) + (-h(n))$
According to this, the routing table of node A is updated, as shown in Table 7.
Table 7 the routing table of node A after adding a heuristic

Distention	Cost	Next Hop	Heuristic value	Net Cost
A	0			
B	8	B	0.49	7.51
C	10	C	2.89	7.11
D	Infinity			
E	Infinity			
F	Infinity			

Table 7 illustrates that the metrics can change the routing path from node B to node C according to the new metrics generated from node B and node C neurons as (7.11) is greater than (7.51).
In table 7, the A is zero because there is no estimated cost from A to A and the infinities mean no direct link between A and nodes (D, E and F). The following Bellman-Ford algorithm is then applied to generate the routing table of all routers.

```
function OptimizesBellmanFord(list vertices, list edges, vertex source) is

    // This implementation takes in a graph, represented as
    // lists of vertices (represented as integers [0..n-1]) and edges,
    // fills three arrays (Heuristic, distance and predecessor) holding
    // the shortest path from the source to each vertex

    distance := list of size n
    predecessor := list of size n
    Heuristic := list of size n

    // Step 1: initialize graph
    for each vertex v in vertices do

        distance[v] := inf      // Initialize all vertices to infinity
        predecessor[v] := null   // And having a null predecessor
        distance[source] := 0  // Distance from the source to itself is 0
        Heuristic [v] := neuralNetwork(delay(v), bandwidth(v),
                                                  reliability(v) )
     //Step 2: updating the distance value, according to metrics.
     for each vertex v in vertices do
         distance[v] :=  distance[v] - Heuristic [v]

    // Step 3: relax edges repeatedly
    repeat |V|-1 times:
        for each edge (u, v) with weight w in edges do
           if distance[u]+w< (distance[v] then
                 distance[v] := (distance[u] + w)
                 predecessor[v] := u

       //Step 4: updating the heuristic value, the metrics may change.
     for each vertex v in vertices do
            Heuristic [v] := neuralNetwork(delay(v), bandwidth(v),
                                                   reliability(v) )
             distance[v] :=  distance[v] - Heuristic [v]


        // Step 5: check for negative-weight cycles
     for each edge (u, v) with weight w in edges do
         if distance[u] + w < distance[v] then
             // Step 4: find a negative-weight cycle
             negativeloop := [v, u]
             repeat |V|-1 times:
                 u := negativeloop[0]
                 for each edge (u, v) with weight w in edges do
                     if distance[u] + w < distance[v] then
                         negativeloop := concatenate([v], negativeloop)
             find a cycle in negativeloop, let it be ncycle
             // use any cycle detection algorithm here
             error "Graph contains a negative-weight cycle", ncycle

     return distance, predecessor
 function neuralNetwork(delay, bandwidth, reliability)is
```

Figure 3 the Optimized Bellman-Ford algorithm

It does this by beginning at a source node in a network and attempting to find the shortest and most cost-effective route to a target node in the graph. It remembers a tree of pathways that start at the source node and expand when more edges are added, and it does this up until the point where its termination condition is satisfied .

The added heuristic value to optimize the Bellman-Ford algorithm will increase the probability of minus distance cost, which makes it impossible to use for Dijkstra's shortest path algorithm that is used in OSPF dynamic routing. The Dijkstra negative weights will cause this algorithm to produce incorrect results. On the other hand, the Bellman-Ford algorithm, which is used in the RIP algorithm, contains a negative weight solution part in its algorithm that make subtracting the effect of metrics on its distance not produce any error routing but instead will enhance the routing protocol and decrease the bottleneck that happens due to low bandwidth of a specific node for example .[٢٠ ,١٩]

IV ConclusionThis paper presents a new technique for optimizing the efficiency of the Routing Information Protocol (RIP), one of the earliest dynamic routing network protocols. The original RIP protocol only takes the hop count into account to determine the routing path between the source and destination.

The paper combines several key metrics that increase network efficiency, such as bandwidth, delay, and reliability, as weighted input to a cost graph of a network to generate a better routing table for network routers.

As a result, a better routing table is generated compared to the regular RIP routing, which gives higher priority to routers with higher bandwidth or lower delay, as well as taking into account the reliability and physical environment of the router to avoid bottlenecks. The Bellman-Ford algorithm is chosen for this purpose due to its immunity to negative weights, unlike the Dijkstra algorithm, which produces a wrong routing with the presence of negative weights.

In the future, additional metrics and weight updates based on throughput can be applied to this proposed system to create a fully optimized RIP protocol.

References

[١]Pablo PavÃ³n Mari, “Optimization of Computer Networks: Modeling and Algorithms”, Wiley; 1st edition (May 2, 2016), ISBN-10 :   9781119013358, ISBN-13 :   978-1119013358

[٢]Michal Pioro, Deepankar Medhi, “Routing, Flow, and Capacity Design in Communication and Computer Networks” ELSEVIER Morgan kaufmann publiesher, 1st Edition - July 1, 2004, Hardcover ISBN: 9780125571890, eBook ISBN: 9780080516431
[٣]B. Mao et al., "Routing or Computing? The Paradigm Shift Towards Intelligent Computer Network Packet Transmission Based on Deep Learning," in IEEE Transactions on Computers, vol. 66, no. 11, pp. 1946-1960, doi:10.1109/TC.2017.2709742, 2017.
[٤]N. Kato et al., "The Deep Learning Vision for Heterogeneous Network Traffic Control: Proposal, Challenges, and Future Perspective," in IEEE Wireless Communications, vol. 24, no. 3, pp. 146-153, doi: 10.1109/ MWC.2016.1600317WC,2017.
[٥]Vukelic, Djordje, et al. "Multi-objective optimization of steel AISI 1040 dry turning using genetic algorithm." Neural Computing and Applications 33.19 (2021): 12445-12475.
[٦]Solarte-Pardo, Bolivar, Diego Hidalgo, and Syh-Shiuh Yeh. "Cutting insert and parameter optimization for turning based on artificial neural networks and a genetic algorithm." Applied Sciences 9.3 (2019): 479..
[٧]Aljarah, Ibrahim, Hossam Faris, and Seyedali Mirjalili. "Optimizing connection weights in neural networks using the whale optimization algorithm." Soft Computing 22.1 (2018): 1-15.
[٨]Faris, Hossam, Ibrahim Aljarah, and Seyedali Mirjalili. "Improved monarch butterfly optimization for unconstrained global search and neural network training." Applied Intelligence 48.2 (2018): 445-464.
[٩]Amirsadri, Shima, Seyed Jalaleddin Mousavirad, and Hossein Ebrahimpour-Komleh. "A Levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training." Neural Computing and Applications 30.12 (2018): 3707-3720..

[١٠]Yan-Jing Wu, Po-Chun Hwang, Wen-Shyang Hwang and Ming-Hua Cheng, “Artificial Intelligence Enabled Routing in Software Defined Networking” published at applied science MDPI, Received: 25 August 2020; Accepted: 17 September 2020; Published: 20 September 2020.
[١١]Dr. M. Duraipandian, “PERFORMANCE EVALUATION OF ROUTING ALGORITHM FOR MANET BASED ON THE MACHINE LEARNING TECHNIQUES” Journal of trends in Computer Science and Smart technology (TCSST) (2019) Vol.01/No.01, Pages: 24-35, https://irojournals.com/tcsst/, DOI: https://doi.org/10.36548/jtcsst.2019.1.003
[١٢]Amruta Deshmukh, M. A. Pund, “Improving the Performance of P2P Routing” using Adaptive Machine Learning” International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8 Issue-8S3, June 2019
[١٣]Jiang, J., & Fan, J. A. (2020). Simulator-based training of generative neural networks for the inverse design of metasurfaces. Nanophotonics, 9(5), 1059-1069.
[١٤]Karim, Syed Shujaat, et al. "Future advances and challenges of nanomaterial-based technologies for electromagnetic interference-based technologies: A review." Environmental Research 205 (2022): 112402.
[١٥]Oris K. Sulaiman, Amir M. Siregar, Khairuddin Nasution and Tasliyah Haramaini (2018) “Bellman Ford algorithm - in Routing Information Protocol (RIP)” April 2018Journal of Physics Conference Series 1007(1):012009, DOI:10.1088/1742-6596/1007/1/012009
[١٦]Younes Ben Chigra, Abderrahim Ghadi, Mohamed Bouhorma, “Novel metric for performance evaluation of routing in MANETs”, SCA '18: Proceedings of the 3rd International Conference on Smart City Applications October 2018 Article No.: 78 Pages 1–4 https://doi.org/10.1145/3286606.3286855
[١٧]Patrik Carlsson, Doru Constantinescu, Adrian Popescu, Markus Fiedler and Arne A. Nilsson, “Delay Performance in IP Routers”, Dept. of Telecommunication Systems, School of Engineering, Blekinge Institute of Technology, 2004 availaable at:
https://www.diva-portal.org/smash/get/diva2:837930/FULLTEXT01.pdf
[١٨]Ghazanfar Abbas, Saman Iftikhar and Nadeem Asif, “Effects Of Bandwidth And Delay On The Routinginstability” European Scientific Journal, April edition vol. 8, No.8 ISSN: 1857 – 7881 (Print) e - ISSN 1857- 7431, available at https://core.ac.uk/download/pdf/328023394.pdf
[١٩]Peter Norvig, Stuart J Russell , “Artificial Intelligence”, PRENTICE HALL SERIES, 3e, ISBN: 0136042597 ©2010
[٢٠]Nyan Dawood Salman, Hanan Kamal AbdulKarim, “DYNAMIC SHORTEST PATH ANALYSIS OF ROUTING TECHNIQUE VIA GENETIC ALGORITHM” jouranl of enginering and sustainable development, Vol. 22, No. 4, July 2018, ISSN 2520-0917, https://doi.org/10.31272/jeasd.2018