Using Moore Dijkstra Algorithm with Multi-Agent System to Find Shortest Path over Network

Basem Alrifai[1], Hind Mousa Al-Hamadeen[2]
Department of Software Engineering, Prince Abdullah Bin Ghazi Faculty of Information Technology, Al-Balqa Applied University,Al-Salt, 19117, Jordan

***Abstract*—finding the shortest path over network is very difficult and it is the target for much research, after many researches get the result in many of algorithm and many a mount based on the performance for these algorithm .Shortest paths problems are familiar problems in computer science and mathematics. In these problems, edge weights may represent distances, costs, or any other real-valued quantity that can be added along a path, and that one may wish to minimize. Thus, edge weights are real numbers and the specific operations used are addition to compute the weight of a path and minimum to select the best path weight.**

In this paper we use the Dijkstra's algorithm with new technique to find the shortest path over network to reduce the time we need to find the best path, in this paper we use node for network with the same value which can be use it to find the shortest path but this depend on the number of transition for every node when the node have high number then the node have the high priority to choose it by using this method we descries the time to find the short path .to make this algorithm more distinguish apply multi-agent system (Automata with multiplicities) to find the short path.

Keywords—multi-agent system; shortest paths problems; Dijkstra Algorithm; Automata with multiplicities

I. INTRODUCTION

The most common operation is finding the short path from vertex to another the shortest path from vertex u to vertex v is path with minimum weight we can define the shortest path algorithm as the following [3] Digraph

G = (V, E) where v is vertex and E is edge in the graph the weight function W: E _R Weight of path p = v1 v2……vk is the sum of the weights of its constituent edges is minimized.

W (p) = ∑ w (vi, vi+1) Formatter will need to create these components, incorporating the applicable criteria that follow.

They are many types for shortest path problem such as single- destination shortest- path problem, single–pair shortest path problem, all pair shortest-path problem.

In this paper we will work on single source shortest path problem from vertex v as the source to all other vertices, we have many algorithm to solve this problem and evaluate the shortest path problem, we will make enhancement to the Dijkstras algorithm by when we have two node have the same value we will added many information to node itself.

In past we choose the node randomly but by using the Dijkstras algorithm enhancement we have decision to choose the node based on the number of transition for the node it self

We added multi agent system as a new technique in addition to the Dijkstras algorithm enhancement we use the automata multiplicities in this method there is addresses for every path

II. PROPOSED METHOD

Dijkstra's algorithm, conceived by computer scientist Edsger Dijkstra in 1956 and published in 1959,[1][2] is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms. The main point for this algorithm it is started at the source vertex s and the tree is T every vertices added to T firstly is start S then the vertices which is closest to S then next closest … etc[11]

A. Dijkstra's Algorithm Enhancement

The enhancement Dijkstra's algorithm is represented when we have more than two edges with the same weight such as (V→U) and (V→W) in this case we choose the vertex which have the maximum transition if (U) has the maximum transition than (W) we use the vertex (U) otherwise we choose (W).

B. Pseudo- code for Dijkstra's Algorithm Enhancement

DIJKSTRA'S ALGORITHM ENHANCEMENT (GRAPH G = (V, E),

```
L = labels for every vertex (weight and path)
For I = 1 to n
     L (V1) = infinity
     L (start) = (0, empty)
     S = {}
While S doesn't contain end {
U = vertex not in S with minimal L (u)
If there are more than two nodes have the same L (u)
      Choose vertex node u which has the maximum transition
       Add u into S
For all vertices v not in S that u is adjacent to v
       If L (U) + W (U, V) < L (V)
       L (V) = L (U) +W (U, V)
          RETURN L (end)}
```

C. Example for Dijkstra's Algorithm Enhancement

If we have the graph contains 8 vertexes as shown in figure 1 we compute the shortest path by two algorithm Dijkstra's Algorithm and Dijkstra's Algorithm Enhancement

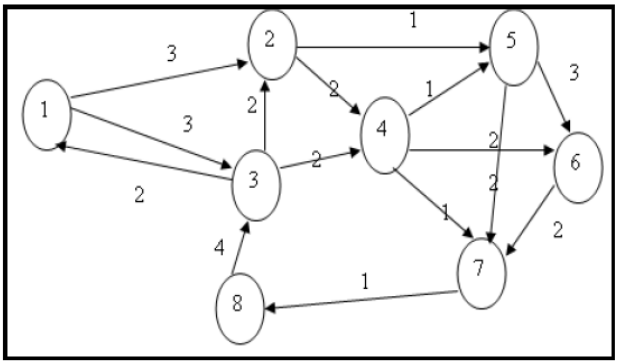

Fig. 1. Example for Dijkstra's Algorithm Enhancement

If we compute the shortest path by using the Dijkstra's Algorithm from source vertex 1 to 8 The path is (1→2→5→7→8) and the shortest path is 7 but if compute the shortest path by using Dijkstra's Algorithm Enhancement we find the path is (1→3→4→7→8) and the shortest path is 7 in this case choose vertex 3 (1 3)because vertex 3 has 3 transition (3→4 ,3→2 , 3→1) whereas vertex 2 has 2 transition (2→ 5,2→ 4) , we have two vertices 2,3 with same value (weight = 3) (1 →3,1 →2)

The difference between two algorithms in Dijkstra's Algorithm the counter for the time is (7) but the Dijkstra's Algorithm Enhancement is (4)

III. Multiagent System and Automata Multiplicities

The multiagent system consists of number of agent, the agent interacts and represent as a user with different goals, and these agents show the ability to cooperate, coordinate and negotiate with each other [1].

An Automaton is advice with permit to assign to every word a coefficient in a smearing and this in an implementable form mainly using matrix computation [8].

We will be able to build effective operation on such automata using of the algebraic structures of the output data [6].

Let k a smearing then an automaton is the data a five up let (Q, A, μ, λ, γ) with :

- Q: the finite set of states
- A: a finite set Alphabet
- μ: A→k Q*Q
- λ € K Q*1 : the set of initial states together with initial values
- γ € k Q*1

Where T = {(q1,a μ(a)q1,q2,q2)} q1,q2 € Q, μ (a) q1,q2# 0[11].

I = {(q € Q :λ (q)#0)}is the set of initial states

F = {(q € Q :λ (q)#0)}is the set of final states

Label (f) =a

Tail (f) =q1

Head (f) =q2

Weight (f) = a

Path c = f1……fm is an element of T

Now we associate the support of T for the weighted graph with edges [6].

Edges (TT) = {(q, a, α, r) € Q* A*K*Q}

This means that every edges (q→r) is superscripted by the pair such that

T(q, a, r) =α #0

The automaton can be observed by means the function it generates this function will be called the behavior of the automaton.

The local behavior of A between two states p;q € Q for the label w € A* is the product of the initial weight λ (p) the total weight of the set of path between p and q with label w and the final weight γ(q) it read [8]

A(p,q) (W) =∑p.q € Qap,q(W)

A. An Agent Modeling Framework Based on Automata Multiplicities

The formalism which is used in our work for the representation of agent behavior produced by perception and action is automata with output the finite inputs alphabet corresponds to the actions set from this output alphabets we build a smearing corresponding to the polynomials over this output alphabet [8]. As we described an automaton with multiplicities over a finite alphabets ∑ and a smearing K is a 5 tuple (∑; Q; I; T; δ) with Q a finite set of states and I, T, δ being mapping such that [8].

I: Q →K
T:Q →K
δ: q* ∑*Q →K

Where I is the set of initial states and T is the set of final states and is δ the transition function Such a structure is useful when transition have outputs to each input word of ∑ is associated an output element of K thus the behavior of an automaton with multiplicities is a series

S = ∑ w€∑*(S │W) W

Where (S │W) is the output elements associated to input word

Example

We have the same graph

Q = {1, 2, 3, 4, 5, 6, 7, 8}
A={a ,b , c}

Now we compute the shortest path by using two algorithms

No.	Counter time for Dijkstra's Algorithm	Counter time for new vision Dijkstra's Algorithm	δ & A
1	9	5	δ={(1,(A,4),2),(1,(b,6),3),(1,(a,4),4),(2,(d,1),5),(3,(b,3),5),(4,(c,2),5),(4,(d,3),6),(5,(b,2),6),(5,(a,4),7),(5,(b,3),8),(6,(d,1),8),(7,(d,2),8),(8,(c,2),9),(9,(b,4),10)} . A={a,b,c,d}
2	7	4	δ={(1,(a,3),2),(1,(b,3)3),(2,(1,c),5),(2,(a,2),4),(3,(c,2),5),(4,(d,3),6),(5,(b,2),6),(5,(a,4),7),(5,(b,3),8)(7,(d,2),8),(8,(c,2),9),(9,(b,4(,10)} A={a,b,c }
3	6	3	δ={(1,(b,2),3),(2,(b,2),6),(3,(b,4),4),(3,(c,4),5),(3,(b,4),2),(4,(c,1),5),(4,(b,3),7),(4,(c,4),6),(5,(c,3),6),(6,(c,1),7),(7,(c,1),8),(8,(b,4),3)} A={a,b,c }
4	11	5	δ={(1,(b,5),2),(1,(c,5),4),(2,(c,2),3),(3,(c,2),5),(4,(b,3),5),(4,(b,2),6),(5,(b,6),6),(5,(b,2),7),(6,(c,3),8),(7,(b,2),8),(8,b,5),10),(9,(c,2)10),(9,(b,3),3),(10,(b,6)11),(10,(b,5),12),(11,(c,2),12),(12,(b,1(,11),} A={a, c }
5	10	6	δ={(1,(c,3),2),(1,(a,3),3),(2,(b,1),4),(2,(b,3),5),(3,(a,4),1),(3,(b,2),4),(3,(c,2),6),(4,(a,1),6),(4,(c,3),7),(5,(a,2),7),(6,(b,2),7),(7,(a,2),8),(8,(c,4),10),(8,(b,3),9),(9,(a,2),10),(10,(a,3),1),(11,(b,4),9),} A={a,b,c }

Fig. 2. computing the shortest path by using two algorithms

The value of shortest path for this graph from vertex (1) to (8) by using Dijkstra's Algorithm is (7)= (3+1+2+1) in order where the path is (1→2→5→7→8) without any addressing because this algorithm applied without using automata with multiplicities but if we apply new algorithm we find that path is (1→3→4→7→8) with the same value of shortest path (7) =(3+2+1+1) in order but this method we distinguish this path by (bcbc) addressing which is obtained from A ={a, b,c} where edge (1,3) has label (b), edge (7,8) has label (c) edge(4,7) has label b and edge (7,8) has label c so the final addressing for this path (concatenated) is (bcbc)

IV. Results

We used different graph to find shortest path by using two algorithms: Dijkstra's Algorithm and Dijkstra's Algorithm enhancement applied on multiagent system. The performance of our proposed vision of Dijkstra's Algorithm is depending on analysis of algorithm and the criteria which are used for prove performance for our algorithm.

Figure 3 and figure 4 display comparison between two algorithms, we see that the counter time for Dijkstra's Algorithm enhancement applied on multiagent system is less than the Dijkstra's Algorithm.

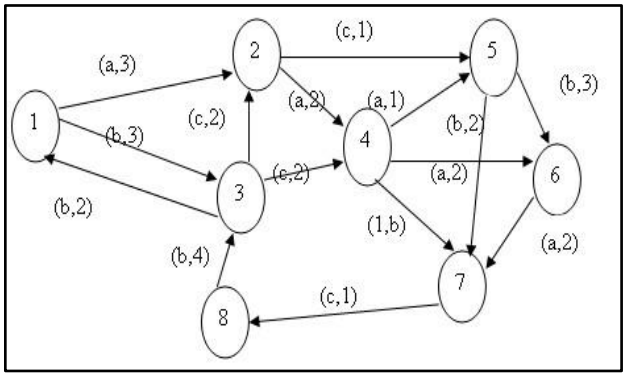

Fig. 3. Comparison between two algorithms

No.	No of vertices	Path in Dj	Path in new vision of Dj	The value of shortest path
1	10	1→2→5→8→9→10	1→4→6→8→9→10	14 from 1 to 10
2	8	1→2→5→7→8	1→3→4→7→8	7 from 1 to 7
3	7	1→3→2→6→7	1→3→4→7	9 from 1 to 7
4	12	1→2→3→8→10→12	1→4→6→8→10→12	20 from 1 to 12
5	10	1→2→4→7→8→10→11	1→3→6→7→8→10→11	16 from 1 to 11

Fig. 4. display path for graph in figure 3

V. Conclusion and Future Work

In this work, enhancement of Dijkstras algorithm for solving single source shortest path problem is discussed.Algorithm tries to solve the problem when have more than two vertices with the same weight and use Multiagent systems techniques.

The properties of this algorithm when we compared with other algorithms can summarized in the following points:

- It is simple and easy to implement over any application
- Uses Q apriority queue ADT

- It is a very Efficient Algorithm to calculate the Shortest Path.
- Uses multi-agent system (simulated by automata with multiplicities) which enhances overall system performance specifically along the dimensions of computational efficiency reliability extensibility maintainability flexibility and reuse

The time required for implementation over any graph by using this new vision of Dijkstras algorithm is less than that by using Dijkstras algorithm .The main shortcoming of this algorithm is this only works if the edges of the graph are nonnegative (Negative weights are not allowed)

In this future we can apply the idea of this method on other algorithm which are used to find shortest path such as(Bellman –Ford algorithm Floyd-Warshall algorithm and Johnsons algorithm) Also we can use other technique for multiagent system with these algorithm which is a specifically representation of automata with multiplicities can be used represent a deterministic agent behavior which is driven by perceptions that induce internal state transitions and can lead to specific action from the agent .(it know transducers) as finite state automata

REFERENCES

[1] Bordini, R,H bner, and Wooldridge , M2007 Programming Multi-Agent Systems in AgentSpeak using Jason.

[2] Buse, D. Wu, Q2007 .IP Network-based Multi agent system for Industrial Automation . Springer-Verlag London Limitied.

[3] Common ,T Leiserson , c, Rivest, R and Clifford S 2001 .Introduction to Algorithm 2nd Edition,MIT press 2001,PP[.492-508].

[4] G Duchamp, M.Flouret and E langerotte 1999.operation over automata with multiplicities .

[5] Harris,S and Ross, J 2006 Begging algorithm.Wiley publishing ,Inc, Indiana.

[6] Jaff ,L, Bertelle , 2008 shift operation and complex system modinling identification and control Vol.3, No 1,2008.

[7] K culick and J. Kari 1995 Finite state transformations of images.

[8] Wooldridge M,2002 An introduction of multiagent system 2nd John Wiley and Son.