International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 11, November 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

Adapted A* Algorithm for the Shortest Path Problem with Position-Based Learning Effects

Mr. N.Mahendra[1], Mrs. Chokkakula Devi[2]

[1,2]*Assistant Professor,*

Dept.of CSE, WISTM, Pendurthy, Visakhapatnam,A.P (India)

ABSTRACT

The shortest path problems (SPPs) with learning effects (SPLEs) have numerous potential and intriguing applications. In any case, in the meantime they are exceptionally unpredictable and have not been contemplated much in the writing. In this pape0072, we demonstrate that learning impacts make SPLEs totally not quite the same as SPPs. An adjusted A (AA*) is proposed for the SPLE issue under examination. In spite of the fact that worldwide optimality infers neighborhood optimality in SPPs, it isn't the situation for SPLEs. . As all sub paths of potential most brief arrangement ways should be put away amid the pursuit procedure, a search graph is embraced by AA* instead of a search tree utilized by A. Monotonicity and consistency of the heuristic elements of AA* are re-imagined and the relating properties are analyzed . Consistency and monotonicity connections between the heuristic elements of AA* and those of A* are investigated Their effects on proficiency of looking systems are hypothetically investigated and tentatively assessed.*

Key Words: A* Search, Optimality, Learning Effect, Shortest Path.

I. INTRODUCTION

A computer network or data network is a telecommunications network which allows nodes to share resources. In computer networks, networked computing devices exchange data with each other using a data link[1,2,3]. The connections between nodes are established using either cable media or wireless media. The best-known computer network is the Internet. Network computer devices that originate, route and terminate the data are called network nodes. Nodes can include hosts such as personal computers, phones, servers as well as networking hardware. Two such devices can be said to be networked together when one device is able to exchange information with the other device, whether or not they have a direct connection to each other. Computer networks differ in the transmission medium used to carry their signals, communications protocols to organize network traffic, the network's size, topology and organizational intent. Computer networks support an enormous number of applications and services such as access to the World Wide Web, digital video, digital audio, shared use of application and storage servers, printers, and fax machines, and use of email and instant messaging applications as well as many others. In most cases, application-specific communications protocols are layered over other more general communications protocols. This formidable collection of information technology requires skilled network management to keep it all running reliably.

The shortest path problem is the problem of finding a path between two veracities or nodes in a graph such that sum of weights of its constituent edges are minimized. Shortest path problems (SPPs) are widespread in practical applications (e.g., logistics, transportation, robot path planning[4,5], vehicle routing[6], no-wait flow shop scheduling[7], etc.). SPP consists of finding the shortest path from the source node to the sink node in a graph. Generally, the distance, time or price of traversing of each arc is called cost. There are a large number of paths in the graph.

The shortest path is the one with the minimum total cost. Traditionally, costs of all arcs are assumed to be known in advance and the SPP is called static SPP (SSPP)[6]. However, there area lot of dynamic SPP (DSPP) in practical systems in which the cost of each arc changes according to some factors (e.g., traffic status and learning experiences). Though there are a lot of studies on DSPPs (especially in traffic systems), the DSPP problem with learning experiences or effects (SPLEs) has not been considered yet. In practice, costs of arcs in a path usually change with learning experience or "learning effect"[8]. Learning effect was first observed by Wright[9]. Nowadays, there are a lot of topics associating with learning effects. The SPPs in robot soccer matches (robot space exploration, robot rescue in hostile environments, etc.) are typical DSPP problems with learning experiences in which robots obtain learning experiences through interaction with environments using reinforcement learning. More experiences imply shorter possible paths robots can find. In logistic systems, there are a lot of items to be sent to different distribution centers. Finding optimal paths for all items is a typical SPP problem. Workers become more and more experienced after they do the pick-up and drop-down operations many times, which reduces the costs (times for transporting items) as a result of the learning effects. The no-wait flow shop scheduling problem is another typical example. The processing time of a job becomes shorter if it is scheduled later in a sequence because the worker is more and more proficient in the setup, cleaning, operation, control, or maintenance of the machines. The shortest path problem is a problem of finding the shortest path or route from a starting point to a final destination. Generally, in order to represent the shortest path problem we use graphs. A graph is a mathematical abstract object, which contains sets of vertices and edges. Edges connect pairs of vertices. Along the edges of a graph it is possible to walk by moving from one vertex to other vertices. Depending on whether or not one can walk along the edges by both sides or by only one side determines if the graph is a directed graph or an undirected graph. In addition, lengths of edges are often called weights, and the weights are normally used for calculating the shortest path from one point to another point. In the real world it is possible to apply the graph theory to different types of scenarios. Experience-based learning describes processing times by "S"-shaped functions which includes three phases namely Inception stage, Learning stage and development stage. The cost on each arc in the SPLE changes with its position in the path.

II. METHODOLOGY

Among existing methods for solving SPPs, A* is one of the most popular algorithms. The A* algorithm was originally presented by Hart et al. [10], which was extended from the Dijkstra algorithm [11]. Heuristics are key

to the time performance of A*. The A* algorithm usually outperforms other traditional exact algorithms for SPPs [12]. Recently, many variants of A* have been presented for SSPPs, such as new approach to multiobjective (NAMOA*) [13], explicit estimation search [14], and short-sighted probabilistic planner [15]. Though there are many A* algorithms for DSPPs, most of them are for irregular ones (i.e., arc costs change stochastically). The one-to-all DSPP (finding the shortest paths between a start node to all the other nodes in a graph) for a given departure time can be transformed into an SSPP [16]. Adapted A* algorithm is used to find the shortest path from the start node to some other node but also some other subpaths need to be put in the list of solution paths to be explored. AA* for the SPLE is a best-first heuristic search algorithm.

AA* starts the search process with the start node n0. Initially,n_0 is set as the only node in the search graph SG and the sub path tuple (n_0, g, fl(n_0, 0)) is introduced into the list OPEN.

In every iteration, AA* selects the path P with the smallest f_l (randomly select one to break ties if there are any), which is determined by gl +hl. Each successor of P is expanded by deleting P from OPEN and moving the corresponding g from Gop to Gcl. At the same time, a backward pointer is established from each successor to the last node of P in order to show the optimal path in the final search graph.

Three operations are carried out.

PRUNE is performed if there are dominations

FILTER is performed if there are some bad path(s)

Otherwise, ENTER is performed by inserting the tuples of Otherwise, ENTER is performed by inserting the tuples of P` into both Gop(n_i) and OPEN. The process is repeated until OPEN is empty, i.e., no path can be expanded. The shortest paths from the start node to the goal nodes are constructed by backtracking from SG.

AA* algorithm, a labeled directed graph is shown in the figure 1, where n_0 is the start node and γ is the only goal node. The graph contains 8 nodes and 14 arcs. Because no cycle is included on the path from the start node n0 to Ґ , there are at most ρ = min{8 − 1, 14}= 7 arcs. The learning effect factor α takes a value of −0.2. C is initialized as +∞. c(n,γ) is the cost of arc (n,γ) if it exists in G, otherwise it is +∞. In the graph G without learning effects, the heuristic value h is defined as

$$h(n) = \min\{c(n,\gamma), \min_{n_s \in Sucessor(n)} c(n,n_s) + \min_{n_p \in Predecessor(\gamma)} c(n_p,\gamma)\}$$

In the graph G with learning effects, the heuristic value hl(n) = ρ^{α} ×h is used to evaluate the value of h_l^*. h_l is computed by ρα ×h = 7−0.2 ×h in this example h and h_l for each node are tabulated and shown in the table 1.

International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 11, November 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

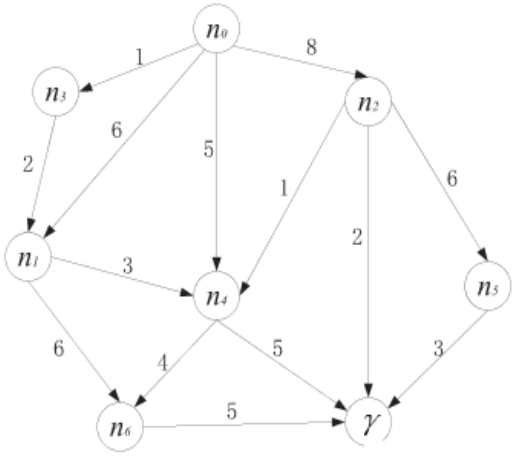

Fig.1 Figure showing the sample Graph

Table 1. Table showing the Heuristic Function

n	n_0	n_1	n_2	n_3	n_4	n_5	n_6	γ
h(n)	3	5	2	4	5	3	5	0
$h_l(n)$	2.033	3.338	1.335	2.710	3.388	2.033	3.388	0

AA* is applied to the example[18], and show the operations PRUNE, FILTER, and the updating operations on Gop and Gcl sets and on OPEN.

AA* records the nondominated sub paths in the search graph. While similar to NAMOA* [17], there are several differences between them.

1. AA* is for a single objective while NAMOA* is for multiple objectives, i.e., AA* returns the shortest solution paths while NAMOA* gives the optimal Pareto solution set.
2. Along one optimal solution path, all the elements of g increase simultaneously in NAMOA* whereas it is not the case in AA*.
3. NAMOA* considers the static SPPs but AA* deals with the regular dynamic shortest path version.

III. RESULTS AND DISCUSSIONS

SPLE has not been studied extensively in the literature. Node ni [identified by its coordinate (x, y)] has several neighbors as successors. Each node on the angle points has two neighbors, each node on the edges has three and each inside node has four. Assume that the learning index α is −0.2. The involved algorithms search one of the shortest paths from the start node (0, 0) to the goal node (s1 − 1, s2 − 1) with learning effects. An extreme case

International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 11, November 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

is that the longest path has s1 × s2 nodes. The algorithm is implemented using the Linux OS and java and the results are shown in the diagrams from 2 to 6.

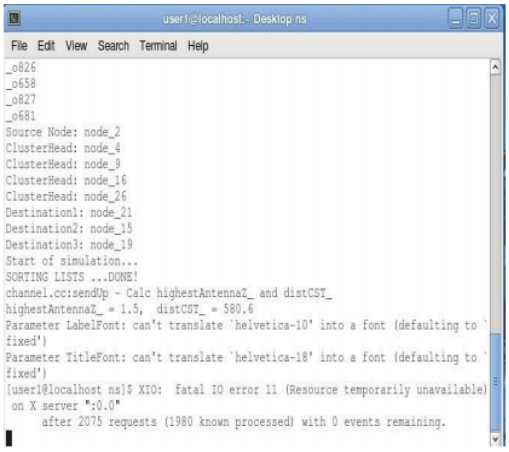

Fig.2. Figure showing the start of a local host

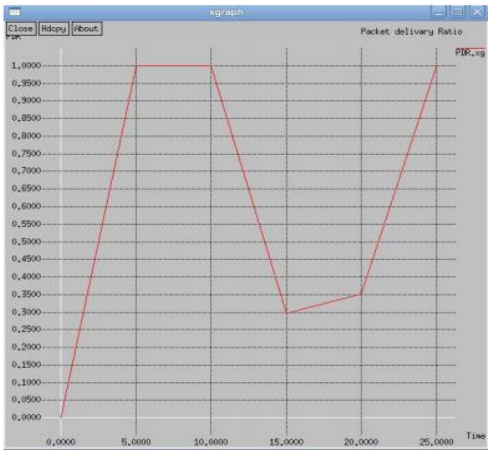

Fig.3. Graph showing the Packet delivery ratio

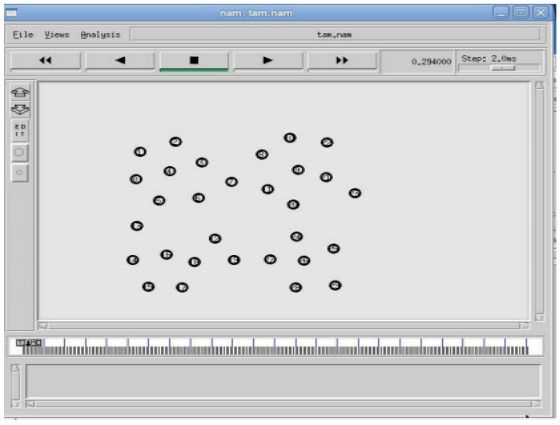

Fig.4. Figure showing the initial nodes

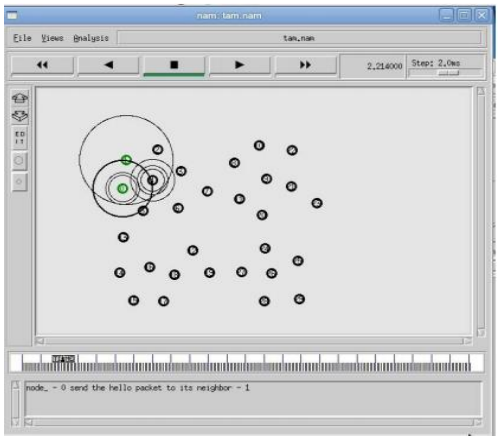

Fig.5. showing the Initializing path between nodes

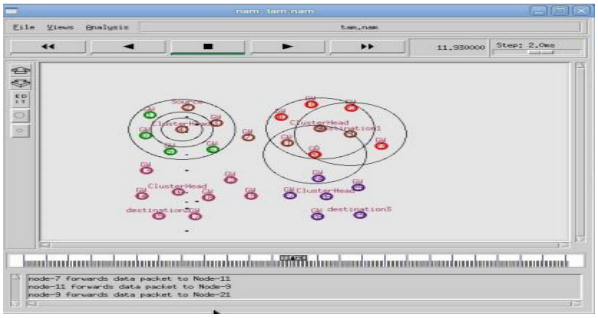

Fig.6. Figure showing communication between nodes

IV. CONCLUSION

In this paper, we have considered the SPLE, of which each arc cost changes dynamically. The cost of an arc in a path is determined by a function of the arc's position in the path because of the learning effects. The shortest sub paths in SPLE have been demonstrated to be unnecessary sub paths of the final shortest path, which is different from the case in SPP without learning effects. The AA* method has been proposed for SPLE problems. A search graph rather than a search tree is adopted to store candidates because there would be more than one candidate sub path for the final shortest path.

With two assumptions, AA* has been proven to be admissible. The efficiency of AA* is influenced by the heuristic function with consistency and monotonicity properties which are similar to the consistent or monotone properties in the A*. Though it is difficult to analyze the consistency of a heuristic function directly, it can be done by judging the consistency of heuristic functions for graphs without learning effects. A closer to (less than) admissible and consistent heuristic function's real value implies less paths to be expanded. Experimental results illustrate that AA* algorithms are far faster than the backtracking method. Though computation times of AA* algorithms increase fast with the size of the problems, the increase in cpu times are far less than those of enumerative search methods. In addition, a tighter heuristic function with monotonicity and consistency with respect to the real cheapest cost results in faster AA* algorithms. For the future, SPLE problems with experience-based or sum-of-processing-time-based learning effects are promising topics.

REFERENCES

[1.] Abramson, N. and Kuo, F.F. Computer-Communication Networks. Prentice-Hall, Englewood Cliffs, N.J., 1975.

[2.] Baran, P. On distributed communications. Rand Corp. Memo RM-3420-PR, Aug. 1964

[3.] Rustin, R. (Ed.) Computer Networks (Proc. Courant Computer Sci. Symp. 3, December 1970), Prentice-Hall, Englewood Cliffs, N.J., 1970.

[4.] N. Mathew, S. L. Smith, and S. L. Waslander, "Planning paths for package delivery in heterogeneous multirobot teams," IEEE Trans. Autom.Sci. Eng., vol. 12, no. 4, pp. 1298–1308, Oct. 2015.

[5.] P. Brass, I. Vigan, and N. Xu, "Shortest path planning for a tethered robot," Comput. Geometry, vol. 48, no. 9, pp. 732–742, 2015.

[6.] M. Gendreau, G. Ghiani, and E. Guerriero, "Time-dependent routing problems: A review," Comput. Oper. Res., vol. 64, pp. 189–197,Dec. 2015.

[7.] P. J. Kalczynski and J. Kamburowski, "On no-wait and no-idle flow shops with makespan criterion," Eur. J. Oper. Res., vol. 178, no. 3, pp. 677–685, 2007.

[8.] D. Biskup, "Single-machine scheduling with learning considerations,"Eur. J. Oper. Res., vol. 115, no. 1, pp. 173–178, 1999.

[9.] T. P. Wright, "Factors affecting the cost of airplanes," J. Aeronaut. Sci.,vol. 3, no. 4, pp. 122–128, 1936.

[10.] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Trans. Syst. Sci. Cybern., vol. 4, no. 2, pp. 100–107, Jul. 1968.

[11.] E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, no. 1, pp. 269–271, 1959.

[12.] I. Chabini and S. Lan, "Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks," IEEE Trans. Intell. Transp. Syst., vol. 3, no. 1, pp. 60–74, Mar. 2002.

[13.] L. Mandow and J.-L. P. de la Cruz, "Multiobjective A* search with consistent heuristics," J. ACM, vol. 57, no. 5, 2010, Art. no. 27.

[14.] J. T. Thayer and W. Ruml, "Bounded suboptimal search: A direct approach using inadmissible estimates," in Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI), Barcelona, Spain, 2011, pp. 674–679.

[15.] F. W. Trevizan and M. M. Veloso, "Depth-based short-sighted stochastic shortest path problems," Artif. Intell., vol. 216, pp. 179–205, Nov. 2014.

[16.] S. E. Dreyfus, "An appraisal of some shortest-path algorithms," Oper Res., vol. 17, no. 3, pp. 395–412, 1969.

[17.] L. Mandow and J.-L. P. de la Cruz, "Multiobjective A* search with consistent heuristics," J. ACM, vol. 57, no. 5, 2010, Art. no. 27.

[18.] Yamin Wang, Xiaoping Li, and Rubén Ruiz, "An Exact Algorithm for the Shortest Path Problem With Position-Based Learning Effects" ,IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS,pp-2168-2216, 2016.