Presenting a Model for the Affordable Choice of Wiring Route in the Electrical and Telecommunications Networks in the Residential Areas Based on the Artificial Intelligence A-STAR Algorithm

Mohammad Reza Gholami Dehbalaei**.** Efaf Delshad

Department of Electrical and Computer Engineering, Payame Noor University, Tehran, Iran

Abstract: One of the most important issues that should be strictly considered when wiring the electrical and telecommunications distribution networks in residential areas is choosing the appropriate direction to conduct the wiring project. Lack of attention to this significant issue causes huge economic and technical costs including reduced efficiency of electrical and telecommunications distribution networks. There are several factors in choosing the appropriate direction in each of the residential areas, whose checking can be very time-consuming and full of human errors. Using the methods of artificial intelligence (AI) might be instrumental. In this article, at first all the possible zones in a residential area to be wired are going to be calculated and equated in terms of a tree with roots and nodes with the functions of specific costs. Then, using the A-STAR iterative deepening search algorithm (IDA*) the best routes between the points of origin and destination are going to be determined, and finally the results of a simulated sample with the considered algorithm is going to be presented.

[Mohammad Reza Gholami Dehbalaei**.** Efaf Delshad. **Presenting a Model for the Affordable Choice of Wiring Route in the Electrical and Telecommunications Networks in the Residential Areas Based on the Artificial Intelligence A-STAR Algorithm.** *Life Sci J* 2013;10(1):1068-1070] (ISSN:1097-8135). http://www.lifesciencesite.com. 165

Keywords: wiring, A-STAR algorithm, function of cost, IDA*, artificial intelligence (AI)

1. Introduction

The optimization operations of the electrical and telecommunications distribution networks require implementing the fast and exact algorithms to verify these operations. One of the most significant factors needs to be considered is routing the locations of power and telecommunications cables [1]. To determine the suitable place for cabling in the networks, it is important to route for several times among different posts. These operations are very time-consuming and full of human errors. Furthermore, in some cases the solution found cannot be implemented practically. The artificial intelligence algorithms can present appropriate solutions. In this method, we try to approximate the target correctly without examining all the probable modes.

One of the important search methods in the artificial intelligence is the A-STAR algorithm (A*) [2]. In this method, the cost function is used as the following:

$$F(n)=g(n)+h(n)$$

In this function, g (n) stands for the route cost from the beginning node to the node (n), and h (n) stands for the direct cost (the cheapest route) from the node n to the target node. Therefore, the route is acceptable if f (n) is in the least amount in it. Of course, other methods such as fringe-saving A*, (FSA*) [3], and generalized-adaptive A* (GAA*) [4] are also considerable. The problem happens during using the search algorithm A* is the need for a huge capacity of memory [5]. The simplest way of decreasing the need for memory in A*, is suing the iterative deepening search algorithm (IDA*) [6]. IDA* operates the depth first-search [7] by cutting and shortening the routes that are more than one cost threshold, successively. The first amount of this boundary is the first node found in the considerable routing. If no solution is found up, the amount of the threshold cost is updated, and through a relative increase compared to the previous stage, the algorithm iterates until the optimized solution is found up. In this article, the researcher first creates the search tree and determines the amounts of the tree casts with the functional definitions, and extracts the optimized answer with the IDA* algorithm. Finally, he presents the results of a simulated sample.

1.1. Creating the Search Tree

One of the problems in working with routing programs and searching the optimized route is to extract and collect the data and present them in a database to use the routing programs and algorithms [8]. The primary raw geographical data include only some points and lines that show the available streets and routes. One street or simple route, which includes no crossroads, triode, or roundabout and extends directly, includes two or four information points, have their own horizontal and vertical specifications. Each street is called a root; for example, in a residential area, which includes 200 streets and each street includes 50 crossroads, or triodes, or squares, 6000 roots might be existed. If we save the information of each crossroads, the triode, or the square as a node in the memory, and summarize the data of each front root in one root, the volume of the required data will be reduced to about one-third, i.e. 2000 roots [9].

2.1. Determining the Functions of Costs of Roots and Nodes

To use the A-STAR iterative deepening search algorithm (IDA*) from the search tree, we have to calculate the cost of real and geographical routes between the two nodes, i.e. f-cost (root) as well as the cost of the direct the ideal route between the two nodes, i.e. f-cost(node). In this regard, the cost function of the route should at least include the following effective factors:

- Route length
- The cost of drilling
- Route slope
- Hardness of work

The route length affects directly the cost of laying cables. The cost of drilling changes depending on the type of soil. The route slope increases the costs. Finally, it is easier to work in the wider streets than in narrow and crowded ones. The cost function, f-cost (root) can be considered as the following:

F-Cost (root) = L (K_d× D + H + K_s× S)

In which:

L: the root length of the real route

K_d: coefficient of the drilling cost (depending on the type of soil)

D: the drilling cost per unit length

K_s: coefficient of the extra cost, related to the route slope

S: the real route slope, (unit percentage)

H: extra cost related to the hardness of the work

Moreover, the function of cost f-cost (node) is as the following:

F-Cost (node) = L' (K_d× D + H + K_s× S')

In which:

L': the length of the direct route of the two nodes

S': the direct slope of the two nodes, (unit percentage)

3.1. Finding the Optimized Route Based on the A-STAR Iterative Deepening Search Algorithm (IDA*)

When the search tree was completed, postulate that the goal of determining an optimized route between the two nodes A and B is in the tree, where A is the point of origin and B is the point of destination. If we select a node in the route from A to B, and name it n, we can use the following equivalents:

g (n) = f-Cost(root A-n)

H (n) = f - Cost (node n-B)

Then the lowest cost function can be achieved from the A-STAR algorithm, i.e. f(n) = g(n) + h(n), and the optimized answer can be found up [10]. However, since implementing this function asks for a huge memory, we use the A-STAR iterative deepening search algorithm (IDA*). In this algorithm each of the iterations is a depth cost search that constantly uses a boundary of f-cost. Therefore, each of the iterations develops all the internal nodes for the current f-cost. It searches above the area as well to find the next area. When the internal search of an area was completed, it starts a new iteration by the use of a new f-cost for the next area to achieve the goal [11].

The IDA algorithm has been stated in the following:

```
function IDA*(problem) returns a solution sequence
  inputs: problem, a problem
  local variables: f-limit, the current f-COST limit
                   root, a node

  root ← MAKE-NODE(INITIAL-STATE[problem])
  f-limit ← f-COST(root)
  loop do
     solution, f-limit ← DFS-CONTOUR(root, f-limit)
     if solution is non-null then return solution
     if f-limit = ∞ then return failure; end

function DFS-CONTOUR(node, f-limit) returns a solution sequence and a new f-COST limit
  inputs: node, a node
          f-limit, the current f-COST limit
  local variables: next-f, the f-COST limit for the next contour, initially ∞

  if f-COST[node] > f-limit then return null, f-COST[node]
  if GOAL-TEST[problem](STATE[node]) then return node, f-limit
  for each node s in SUCCESSORS(node) do
     solution, new-f ← DFS-CONTOUR(s, f-limit)
     if solution is non-null then return solution, f-limit
     next-f ← MIN(next-f, new-f); end
  return null, next-f
```

Figure 1: The IDA algorithm

4.1. The Results of the Simulation

The IDA algorithm was implemented for a boundary, length of 2 kilometers and a width of 5.1 kilometers, with about 200 routes, crossroads, triodes, and squares. First, for the algorithm, the node of origin (A), and the node of the destination (B) were determined. Then the output of the algorithm was produced as the following:

Figure 2: output of the algorithm

2. Discussions

To calculate the optimization of the electrical and telecommunications distribution networks, fast, exact, optimized, and full search programs are needed. Thus IDA* algorithm can be the best and the most suitable choice. Due to the high volume of information and the street and route connections, it is impossible to search all the modes. Therefore through determining the cost functions and search in the boundary of specific costs, the considered goal can be achieved. This model can be used for determining the optimized route for the water-supply networks.

.

Acknowledgements:

Author is grateful to Department of Electrical and Computer Engineering, Payame Noor University, Tehran for support to carry out this work.

12/25/2012

Corresponding Author:
Mohammad Reza Gholami Dehbalaei
Department of Electrical and Computer Engineering, Payame Noor University, Tehran, Iran

References

1. Willis,H.Lee ,Northcote – Green ,J.E.D , : Comparison of several computerized distribution planning methods , IEEE Transation on power apparatus and systems , VOL .PAS – 104 , NO . 1 , JANUARY 1985.
2. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. Systems Science and Cybernetics, IEEE Transactions on 4(2) (1968) 100-107.
3. Sun, X., Koenig, S.: The fringe-saving a* search algorithm - a feasibility study. In: IJCAI. (2007) 2391-2397.
4. Sun, X., Koenig, S., Yeoh, W.2008. Generalized adaptive a*. In: AAMAS '08, Int. Foundation for Autonomous Agents and Multiagent Systems 469-476.
5. Zhou, R., Hansen, E.A (2002). Memory-bounded a* graph search. In: Proc. of the Fifteenth Int. Florida Artif. Intell. Research Society Conf., AAAI Press. 203-209.
6. Korf, R.E. (1985). Depth-_rst iterative-deepening: An optimal admissible tree search. Artif. Intell. 27 97-109.
7. Burns, E., Lemons, S., Zhou, R., Ruml, W.: Best-_rst heuristic search for multicore machines. In: IJCAI. (2009) 449-455.
8. Koenig, S. 2010. Dynamic fringe-saving a* (June 2010) Retrieved 7.
9. Schildt ,H. : Artificical intelegence using C , Osborne Mc Graw Hill. 1992.
10. Koenig, S., Likhachev, M., Liu, Y., Furcy, D.: Incremental heuristic search in ai. AI Mag. 25(2) (2004) 99-112.