Designing Optimal Network for Rural Electrification using Multiplier-accelerated A* Algorithm

Jerry C.F. Li
Department of Systems Engineering
Colorado State University
Fort Collins, CO, USA
jerry.li@colostate.edu

Daniel Zimmerle
Energy Institute
Colorado State University
Fort Collins, CO, USA
dan.zimmerle@colostate.edu

***Abstract*—This analysis proposes a modified A* routing algorithm for the routing of networked rural electrification systems in areas with substantial topological variation. Due to geographical complexity, using standard A* algorithm for optimal routing is very inefficient. This new algorithm utilizes a modified heuristic to reduce computational time while achieving near-optimal routing results. The modified algorithm is also more suitable for use in routing studies where inputs are uncertain, requiring Monte Carlo simulations to assess the robustness of proposed routes.**

***Index Terms*--A* algorithm; rural electrification; path finding.**

I. INTRODUCTION

Identifying optimal connection path is a common problem for developing networked systems such as an electric grid, transportation network, and even integrated circuits. The A* algorithm is most widely used for this purpose. However, different applications have specific constraints and researchers have modified the standard algorithm to fulfill these requirements. This paper proposes a modified A* algorithm and compares it with the standard A* algorithm when finding the optimal path for distribution networks between villages in rural electrification applications.

The problem originates from the difficulty of correct planning for small scale renewable energy generation for villages or small towns. While solar and wind generations are technologically mature, designing a near-optimal system for a small area remains challenging:

(a) Planning – for an individual village, resource (i.e. solar radiation and wind) forecasts based on large-area statistics are often unreliable due to the small geographical area and local topology and ground cover. Similarly, forecasting short-term demand is also challenging for a small number of users. Mid-to-long term demand forecast relies heavily on accurate projection of the village population and activities, which is generally unknown at the planning stage. Thus, over-planning and under-planning are quite common for such small-scale projects [1,2,3].

(b) Resources utilization – village locations are historical and chosen for access to water, good soil, food storage, etc, rather than strong solar and wind resources. As such, the villages may not be located in good locations for effective solar and wind power generation. Therefore, if generation development is restricted to village locations, more equipment and land will be required to generate the same amount of energy. [3,4,5].

These two problems can be addressed if development considers a cost-effective network that connects the villages together. Each village will then be supplied by (i) centralized generation sites with strong solar and/or wind resources, and supplemented by (ii) local generation modules that can be added or removed based upon local demand (Figure 1.1).

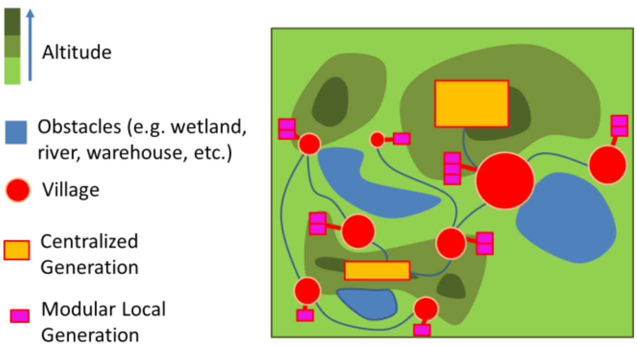

Figure 1.1 Proposed configuration

In addition to optimizing capital investment and resource utilization, this approach can also improve reliability by interconnecting loads with multiple resources. However, the cost-effectiveness of this approach substantially depends on the cost of building the network, which, in turn, depends on selecting an optimal, or near-optimal, connection topology that considers the topology of the area. The two steps for designing such a network are, respectively, (i) finding (near) optimal paths (i.e. lowest cost paths) for connecting any two villages and/or connecting any village(s) to centralized generation sites, and (ii) network synthesis based the optimal paths identified.

While standard methods are available for performing the two tasks, modifications can be made to address specific challenges associated with this rural electric network design. This paper focuses on the first task, i.e., optimal path finding. Figure 1.2 is a small digitized map for illustration, show the two cases used for this study. In practice, much higher resolution maps would be utilized, typically created from surveys or aerial (satellite or unmanned aerial vehicle) photography.

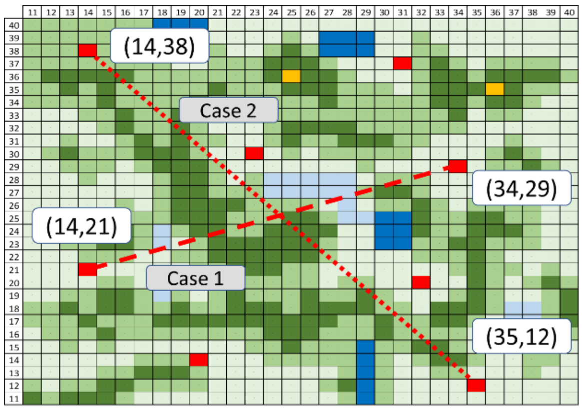

Figure 1.2 Illustrative map

II. A* AND MULTIPLIER-ACCELERATED A* ALGORITHMS FOR OPTIMAL PATH FINDING

A* algorithm is proven and widely used for optimal path finding. Briefly, the algorithm maintains a list of paths originated from the starting point, and, at each iteration, extends a chosen path by one movement until the end point is reached or some termination criteria are satisfied. The algorithm selects a path for extension basing on the minimization of an estimated cost. In standard A* algorithm, this estimated cost is the sum of two components. The first component, denoted by G, is the actual cost for moving from the starting point to the current location on the path. The second component, denoted by H, is a heuristic estimation of the cost for moving from the current position to the end point [6].

To apply A* algorithm to this specific rural electrification problem, the cost function has to be defined accordingly. The first component, G, consists of:

(i) Path length - directly proportional to cost;

(ii) Incremental costs due to changing elevation - In actual project implementation, changing elevation will not only increase the path length, but also increases cost due to additional works required;

(iii) Accessibility – some locations are more difficult to access and traversing them increases the cost. For example, it is more costly to route cables through a wetland.

Thus, the cost of the (i+1)-th movement in the A* search is represented by:

$$C_{i+1} = [(h_{i+1} - h_i)^2 + (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{1/2} + w_1 \,|\, h_{i+1} - h_i \,|\, + w_2\, a_{i+1} \qquad (1)$$

where

x_i, y_i, h_i, a_i are, respectively the coordinates, altitude and accessibility of the i-th point on paths in Figure 1.1. For this analysis, x_i, $y_i \in \{11\text{-}40\}$, $h_i \in \{1, 2, 3\}$ and $a_i \in \{0, 3, 10\}$;

w_1, w_2 are the weighting coefficients. Their selection can be affected by technical capability, equipment availability, material and labor cost, etc. For this analysis, $w_1 = 6$ and $w_2 = 3$.

Heuristic cost H is simply the Euclidean ($\mathbf{R}^2$) distance in the *x,y* plane between the end point and the current point of the search path.

A. *Limitations of standard A* Algorithm*

On each iteration, A* algorithm switches search paths and makes movement based on comparing the estimated cost at the current search position to the estimated costs of a list of previous explored positions.

In similar A* path-finding applications, such as computer games, the 'land' is flat, coordinates typically marked as either "walkable" or "non-walkable", and elevation change happens at a small number of coordinates. These constraints reduce search complexity, and typically only a small number of coordinates on the map need to be explored [7,8]. In contrast, for this problem elevation and accessibility (see Figure 1.2) can change drastically within a small subset of the map. Furthermore, there are very few "non-walkable" coordinates because remote villages are, almost by definition, difficult to access, yet all need to be included in the distribution system. Therefore, all coordinates are accessible but the cost to access each coordinate varies substantially.

These differences challenge A* path-finding and common A* acceleration methods when applied to this rural electrification problem. Given these map characteristics, the standard A* algorithm has a high probably of switching search paths on each iteration. In many cases, the search must be reinitiated from relatively early positions of previously explored paths. Although the algorithm will eventually converge, frequent switching increases the computation burden, and as we will show, makes direct application of the standard algorithm intractable. Common acceleration techniques (e.g. quadtree, navigation mesh, jump point search, etc. [7,8,9,10]) are, as many of them are formulated for computer games, also inapplicable due to frequent and potentially large changes in elevation and accessibility.

To synthesize the final network, optimal paths connecting all point-pair on the map have to be determined. While computation burden may not be critical for one single optimal path finding exercise on a small map such as Figure 1.2, it is an important factor in practice when there are large number of locations on a realistically sized map (Figure 2.1).

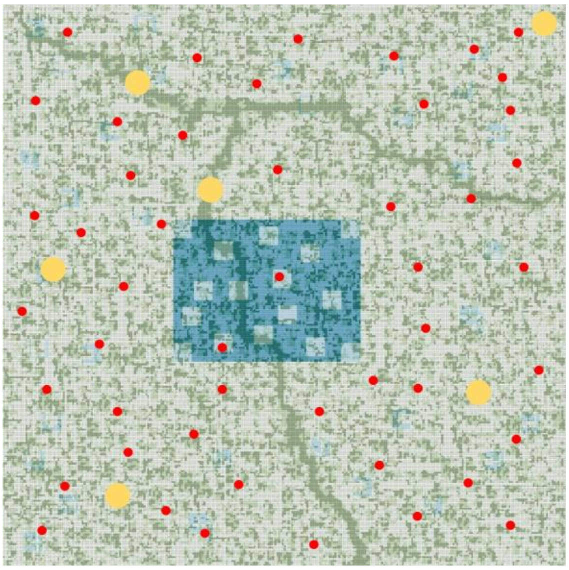

Figure 2.1 Example of a practical map

B. Multiplier-accelerated A algorithm*

While many known acceleration techniques such as quadtree and JPS are not applicable to this analysis, the A* algorithm can be accelerated if we reduce the probability that the algorithm will jump back to earlier locations during later stages of the search. This can be done by exaggerating the heuristic cost for the earlier points using a multiplier. Figure 2.2 illustrates the structure of the proposed multiplier-accelerated A* algorithm.

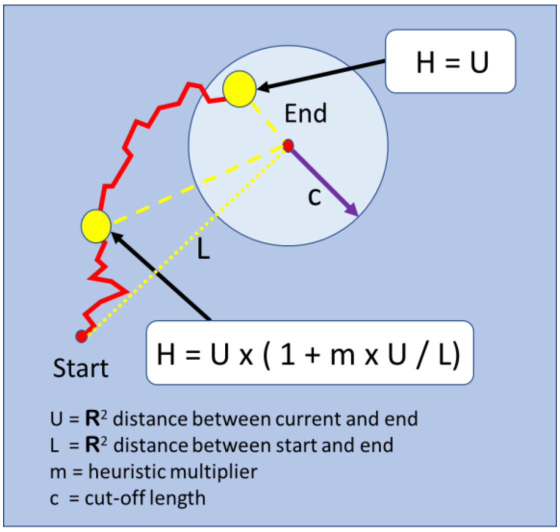

Figure 2.2 Multiplier-accelerated A* algorithm

As shown, the heuristic cost (H) is, instead of simply the Euclidean distance on plane, modulated by a coefficient that decreases as the current point is getting closer to the end point:

$$H = U \times (1 + m \times U / L) \qquad (2)$$

where U is the $\mathbf{R}^2$ distance between current and end points

L is the $\mathbf{R}^2$ distance between start and end points

m is the heuristic multiplier

Equation (2) is applied when the current point of the search path is outside of the cut-off region, shown as the circle in the figure. Choosing appropriate multiplier *m* and the cut-off length c, the heuristic cost function *H* has been artificially inflated and A* algorithm is less likely to jump back to earlier starting points as the *H* becomes larger. When the search path enters the cut-off region (circle defined by radius *c*), *H* reverts to the normal heuristic completes the routing as the standard A* algorithm.

While the proposed weighting for *H* accelerates the A* algorithm, the new heuristic may violate the fundamental constraint of A* that the heuristic is never higher than the optimal path cost between the current and destination locations. That is, heuristic must not be overestimated. Therefore, the modified algorithm may not converge to the global optimal path for higher values of *m*. The question, then, is whether the modified algorithm produces sufficient computational savings, and produces nearly-enough optimal paths, to warrant its use.

Another important aspect of this problem is that the input map, as shown in Figure 1.2, is not perfectly known. Since there is uncertainty in the values at each grid square, it is often useful to use Monte Carlo methods to analyze the robustness of a solution relative to variations in input data. This type of analysis typically needs to perform hundreds, if not thousands, of routings, to analyze whether a proposed routing would be substantially modified for expected variations in the input data.

III. SIMULATION RESULTS AND DISCUSSIONS

A Python program has been developed to execute both algorithms for tests using both routes on the map in Figure 1.2. Computation time is used as a rough indicator for computation burden and reported for a Pentium 2020m PC with typical settings. While this is not a robust indicator of computational effort (results could be affected by the status other background processes on the computer), it provides a comparable metric provided all trials were conducted on the same machine, with same software conditions. An analysis of the size of the path list (the 'closed list' in A* terminology) also provide computational insights. However, this paper is exploratory in nature, and only the computational measurement is reported here.

In the following analyses, the cut-off length (c) is chosen to be 10. This represents 30-40% of the length of the straight line connecting the start and end points. Therefore, the heuristic is modified for 60-70% of the $\mathbf{R}^2$ distance between start and end points. We first analyze the impact of varying the heuristic multiplier, *m*, and later analyze the impact of varying the cut-off length, *c*.

A. *Case 1 – Connecting (14,21) and (34,29)*

Table 3.1 summarizes the computed paths connecting (14,21) and (34,29) in Figure 1.2. The path from the standard A* algorithm is optimal, and the path from multiplier-accelerated A* algorithms vary in optimality. The physical shape of these paths is shown in Figure 3.1.

Table 3.1 Optimal connection between (14,21) and (34,29)

	Connection: (14,21) - (34,29)			
	Computed Optimal Cost	Computation Time (sec)	Normalized Computation Time	% Over-estimate
Standard A*	47.9	198.6	100.0%	0.0%
Accelerated A*				
m = 0.1	47.9	169.8	85.5%	0.0%
m = 0.2	47.9	146.9	74.0%	0.0%
m = 0.5	48.8	117.7	59.3%	1.9%
m = 1	50.7	20.3	10.2%	5.8%
m = 2	50.7	5.8	2.9%	5.8%
m = 5	58.6	3.9	2.0%	22.3%

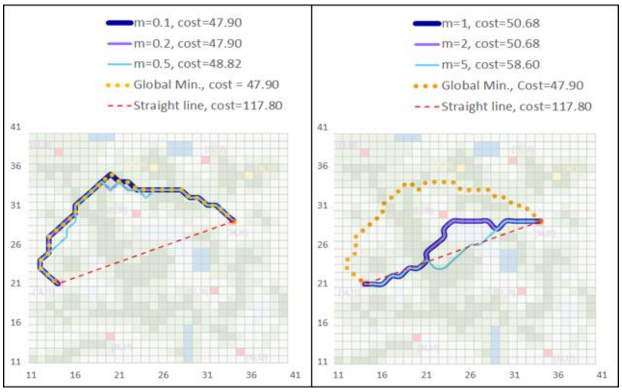

Figure 3.1 Optimal paths connecting (14,21) and (34,29)

The standard A* algorithm identifies the optimal cost as 47.9, a 59.3% reduction from a straight-line between the points. This optimal cost is used as benchmark for evaluating the performance of multiplier-accelerated A* algorithm.

For multipliers of *m=0.1* and *0.2*, the modified algorithm finds paths with the same cost as the optimal path, and these paths track closely to the routing of the optimal path computed by standard A* algorithm. No over-estimation is observed. Computation time, however, has been reduced by 14.5% and 26% respectively.

At a higher multiplier of *m=0.5*, the modified algorithm deviates from the standard algorithm, the path tracks less well, and the resulting path is 1.9% more costly than optimal. However, a reduction in computational time by 40.7% makes the performance tradeoff interesting, particularly for large problems. When multipliers of *m=1* and *2* are used, tracking errors increase further and both lead to an over-estimation of 5.8%. However, computation time has been very significantly reduced by 89.8% and 97.1% respectively.

Finally, at a multiplier of *m=5*, the modified algorithm starts to follow the suboptimal direct path and reports a substantially sub-optimal result that is 22.3% higher cost than optimal, although the computation time has been reduced by 98%.

In practice, small errors in initial routing will be acceptable in most cases, since the problem statement cannot capture all possible impacts on cost in any case. For example, factors such as workmanship deviations, material transportation, consumer preference, changing land conditions, and even local politics, also contribute considerably to the costs of installing a distribution line, and the unknown variances in these costs may well exceed the difference between reported and optimal paths. As such, results from the multiplier-accelerated A* algorithm are acceptable for connecting (14,21) and (34,29) when *m = 0.1, 0.2, 0.5, 1* and *2*. These results cannot be generalized because the results are affected by the actual structure of the area being analyzed, as shown in the following example.

B. *Case 2 – Connecting (14,38)-(35,12)*

Table 3.2 shows the same data as Table 3.1, for connecting (14,38) and (35,12) in Figure 1.2, and the physical shape of these paths is shown in Figure 3.2.

Table 3.2 Optimal connection between (14,38) and (35,12)

	Connection: (14,38) - (35,12)			
	Computed Optimal Cost	Computation Time (sec)	Normalized Computation Time	% Over-estimate
Standard A*	51.7	308.1	100.0%	0.0%
Accelerated A*				
m = 0.1	51.7	264.6	85.9%	0.0%
m = 0.2	51.7	239.2	77.6%	0.0%
m = 0.5	52.5	134.3	43.6%	1.6%
m = 1	54.9	64.6	21.0%	6.2%
m = 2	57.8	10.2	3.3%	11.9%
m = 5	82.2	8.2	2.7%	59.0%

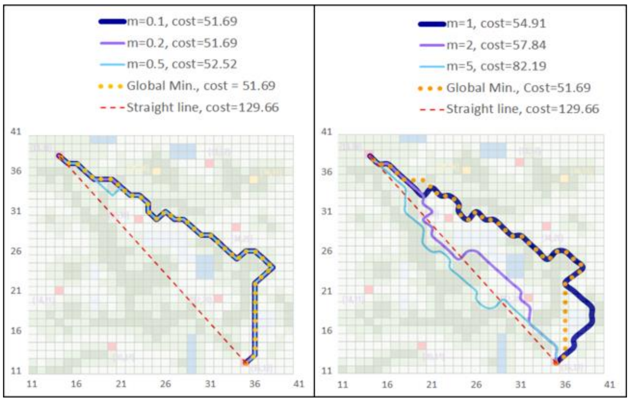

Figure 3.2 Optimal paths connecting (14,38) and (35,12)

For this path, the standard A* algorithm gives an optimal cost of 51.7, which is a 60.1% reduction from the straight-line connection cost. As in the first example, multipliers of *m=0.1* and *0.2* also find the optimal path but reduce computational time by 14.1% and 22.4% respectively.

Similarly, middle values of the multiplier *m=0.5* and *1.0* produce near-optimal paths (cost increased by 1.6% and 6.2% respectively), while reducing computational time 56.4% and 79% respectively. As indicated above, these errors are likely acceptable in practical terms.

For this pair of start and end points, multipliers of *m=2* and *5* exhibit substantial tracking errors and associated suboptimal costs that are 11.9% and 59.0% higher than optimum. Computational savings are significant, however, with reductions of 96.7% and 97.3%, respectively.

Comparing the two cases, similar trends are observed, but suboptimality becomes substantial at higher values of *m*.

C. Impact of Cut-off Value, c

Table 3.3 varies the cutoff, *c*, for two values of the multiplier, *m*. In general, a larger value of *c* will force the algorithm revert to the standard A* algorithm earlier in the search exploration. This may improve results, typically at the cost of higher computational effort.

Table 3.3 Computed optimal cost vs cut-off length

	Connection: (14,38) - (35,12)					
	m = 2			m = 5		
	Computed Optimal Cost	Computation Time	Increase in Computation Time	Computed Optimal Cost	Computation Time	Increase in Computation Time
c = 0	57.84	8.69	-15.1%	82.19	2.2	-73.2%
c = 5	57.84	8.58	-16.1%	82.19	2.52	-69.3%
c = 10	57.84	10.23	0.0%	82.19	8.22	0.0%
c = 15	57.74	16.2	58.4%	82.19	11.69	42.2%
c = 20	56.92	41.15	302.2%	70.38	35.77	335.2%
c = 25	56.09	114.11	1015.4%	69.55	54.72	565.7%

Results indicate that higher values of *c* improve results, but computational time increases rapidly for *c>10*. This is caused by the algorithm investigating larger areas near the end of the search, and possibly traversing regions near the previously chosen near-optimal path. Compared to the nominal results with *c=10*, it is not worth using larger c to improve estimations in the test cases performed here.

IV. Conclusion

In rural electrification, using distribution networks to interconnect villages can potentially improve the efficiency of energy resource utilization and improve supply reliability. However, the cost of building the distribution network is justified by potential savings. To minimize the cost of such a system, the costs for cable routing between any two villages in the concerned area have to be evaluated. While standard A* algorithm and typical accelerators can be used for these calculations, the computation burden is very large due to complex topography in practical rural areas. Computational costs are also important due to the uncertainty in the map itself, which may require many routings to be calculated with variations in the map inputs.

Based upon testing, a small subset of which is shown here, the proposed multiplier-accelerated A* algorithm compares well to the standard A* algorithm for small multipliers, while saving substantial computational time. Although this cannot be over-generalized, additional experiments have shown that the results shown here are greatly amplified in larger practical maps, indicating that the improvements from the algorithm may be justified by the computational savings.

References

[1] H. Zerriffi, *Rural Electrification – strategies for Distributed Generation*, Springer, 2011

[2] A. Kottayil, “Technologies and Application of Microgrid/Smartgrid to Asia Pacific,” unpublished, presented in Session 3 of the 9th meeting of Asia Solar Energy Forum, Asian Development Bank, Beijing, March 2016.

[3] J. Li, “Managing solar resources under micro-grid settings in remote locations,” unpublished, presented at Session 3 in the 9th meeting of Asia Solar Energy Forum, Asian Development Bank, Beijing, March 2016.

[4] V. Kishore, D. Jagu and E. Nand Gopal, “Technology Choices for Off-Grid Electrification,” in *Rural Electrification Through Decentralized Off-grid Systems in Developing Countries*, S. Bhattacharyya, Ed. Springer, 2013, pp. 39-72.

[5] J. Petersa, M. Sieverta, and M. Tomanc, “Rural electrification through mini-grids: Challenges ahead,” *Energy Policy*, Vol.132, 2019, pp.27–31

[6] S. Russell and P Norvig, “A* search: Minimizing the total estimated solution cost” in *Artificial Intelligence*, 3rd ed., Pearson, 2015, pp. 94-99.

[7] P. Yap, N. Burch, R. Holte, and J, Schaeffer, “Any-Angle Path Planning for Computer Games”, in Proc. 2011 AIIDE, pp.201-conference, 2011.

[8] X. Cui and H. Shi, “A*-based Path-finding in Modern Computer Games”, *International Journal of Computer Science and Network Security*, Vol.11(1), January 2011.

[9] S. Rabin, “A* speed optimizations,” in *Game Programming GEMS*, , Charles River Media, 2000, pp.264-271.

[10] D. Harabor and A. Grastien, “Online Graph Pruning for Pathfinding on Grid Maps”, in *Proc. 2011 AAAI*, pp.1114-1119.