CONFERENCE PAPER - This is the version accepted by the conference. Final version uploaded to IEEE Xplore.
doi: 10.1109/AI4G50087.2020.9311085 https://ieeexplore.ieee.org/document/9311085
Citation: J. C. F. Li, D. Zimmerle and P. Young, "Optimizing Networked Rural Electrification Design using Adaptive Multiplier-Accelerated A* Algorithm," *2020 IEEE / ITU International Conference on Artificial Intelligence for Good (AI4G)*, Geneva, Switzerland, 2020, pp. 164-169.

Optimizing Networked Rural Electrification Design using Adaptive 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

Peter Young
Department of Electrical & Computer Engineering
Colorado State University
Fort Collins, CO, USA
peter.young@colostate.edu

***Abstract*— Networked rural electrification can potentially improve energy resources utilization, reduce cost and enhance supply reliability. Identifying optimal connection paths is critical for proper network design. To overcome the inefficiency of applying standard A* path-finding method to complex topography, multiplier-accelerated A* (MAA*) algorithm, which utilizes a modified heuristic, has been developed in previous research. While MAA* can generally reduce computation time by ~90% at the cost of ~10% optimality, the computation burden can still be remarkable for some areas with intricate topological variations. This paper proposes an adaptive version of MAA*. By introducing intermediate nodes in MAA*, the new algorithm significantly simplifies computations in complex regions. This greatly facilitates the analysis and design of optimal network for cost-effective electricity supply to users in remote, difficult-to-reach areas.**

***Index Terms*-- rural electrification; SDG7; A* algorithm; path finding.**

I. Introduction

Enacted by United Nation [1], the seventh Sustainable Development Goal (SDG7) envisions affordable, reliable, sustainable and modern energy will be accessible to all by 2030. As reported by recent studies [2,3], the progress of electrification is remarkable. Population without electricity access decreased from 1.2 billion to 840 million during 2010-2017. On the other hand, the reports also pointed out the problem of rural-urban divide. Among the unserved population, 732 million people are living in rural areas. Electrification of these areas requires extra effort and could be costly due to increased complexity. For example, owing to uncertainties in resource and demand forecasts for small geographical areas, correct planning for renewable energy generation for individual village or small town is challenging. Longer-term, urbanization policy may unexpectedly impact population and hence energy demand [4-6]. In addition, location of the village or town may not be optimal for efficient generation [6-8].

To facilitate electrification for these difficult areas, a networked rural electrification scheme has been proposed [9]. The scheme uses a cost-optimized network to connect villages in a wider area together, and each village is supplied by (i) centralized generation sites with good resources such as strong solar radiation and/or wind for efficient generation, and (ii) supplementary local generation modules that can be added or removed based upon local demand (Figure 1).

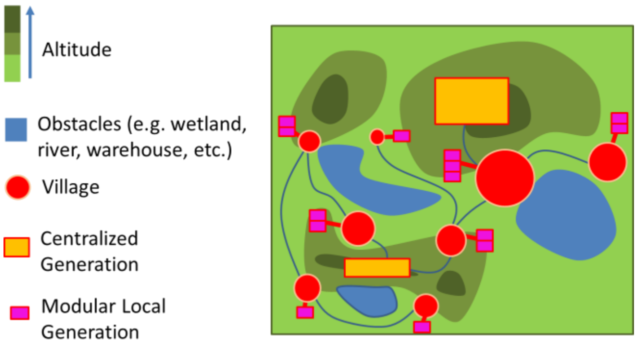

Figure 1 Networked Rural Electrification

Viability of this approach depends on the cost of building the network, and hence depends on correctly identifying an optimal, or near-optimal, connection topology that accounts for the topography of the area. To properly design the network, the two steps are (i) finding (near) optimal paths for connecting any two villages, and also optimal paths connecting each village to the centralized generation sites, and (ii) synthesizing the lowest-cost connection topology based the optimal paths identified. While this approach is conceptually easy to understand, designing such a network is difficult in practice because of high computational complexity. This paper proposes an efficient algorithm that could significantly reduce the complexity and facilitate the design process.

To perform the two mentioned tasks, standard methods are available. For example, the widely used A* search algorithm appears to be a reasonable solution for finding the paths, and minimum spanning tree (MST) can be used to obtain the minimum cost network. However, as explained in Section II, standard A* algorithm is very inefficient for this application and common acceleration techniques are also not applicable due to significant topological (cost) variations. Since hundreds of combinations may have to be evaluated, a fast method suitable for identifying optimal interconnection paths is highly desirable. This will become even important if there are considerable uncertainties in input map and many Monte Carlo simulations will be required to ensure robustness of the solution. Multiplier-accelerated A* (MAA*) algorithm [9] has hence been developed to improve the computational efficiency of pathfinding. While MAA* can generally reduce computation time by ~90% at the cost of ~10% optimality, the computational burden may still be large for areas with intricate topological variations. This paper proposes an adaptive version of MAA*. By introducing intermediate nodes in MAA*, the new algorithm significantly simplifies computations in complex regions. This greatly facilitates the analysis and design of optimal network for cost-effective electricity supply to users in remote, difficult-to-reach areas.

II. Brief Review of Multiplier-accelerated A* Algorithm and it's Performance

Although A* is a proven algorithm for optimal path finding, computation complexity is problematic when applied to a large search space. Complexity of the algorithm is described by $O(b^{\epsilon d})$ [10], where b is the branching factor, ϵ is the error in heuristic estimate defined as $(h^* - h)/h^*$ (where h^* and h are the actual and estimated cost from the node to final goal), and d is the solution depth – i.e. the path length in this application. Figure 2 illustrates the effect of ϵ and d under a branching factor of, for example, 2.13 with constant step cost.

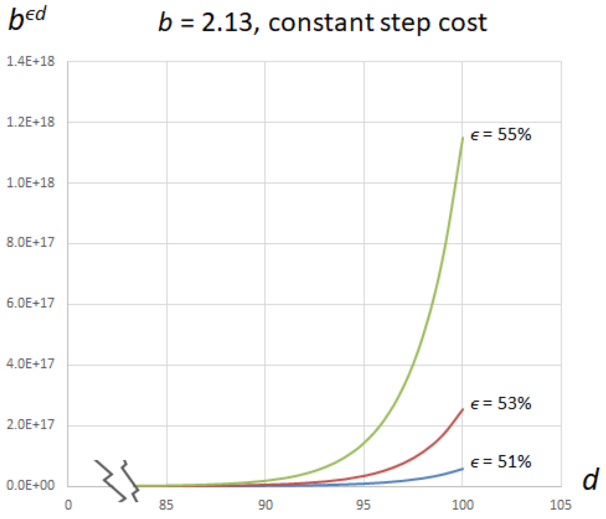

Figure 2 Time complexity vs search depth

As seen, time complexity increases exponentially when the heuristic error ϵ and the solution depth d increase even under constant step cost assumption. To improve computation efficiency, there are two approaches: (i) reducing d by simplifying the search space, and (ii) reducing ϵ by designing better heuristic estimates.

For some applications (e.g. video games), simplifying search space is quite straight-forward. Methods such as Quadtree, NavMesh etc. essentially divide space into simple geometric shapes and use a significantly fewer waypoints to represent these shapes and hence resulting a much smaller d [11, 12]. Standard A* algorithm can then be applied to these waypoints. Many other acceleration techniques [13, 14] are also based on different geometric simplifications. Viability of geometric simplification depends heavily on isotopicity of the search space. For networked rural electrification, these techniques are not applicable because varying topography leads to a high degree of cost anisotropicity. The search space, therefore, cannot be effectively reduced.

Reducing error in the heuristic estimate is even more challenging since ϵ is highly dependent on the properties of the search space. Using a simple heuristic estimate to closely track the actual cost throughout the entire space is not usually achievable, and the high degree of anisotropicity in this application will lead to even larger ϵ. Researchers have developed more sophisticate heuristic estimates in order to reduce complexity [15-17], but most of these methods are only applicable under some specific conditions. For networked rural electrification, search space may take any physical form and these methods are not directly relevant.

Multiplier-accelerated A* (MAA*) Algorithm was developed to tackle this problem. Essentially, MAA* is an algebraic, instead of geometric, search space reduction method that selectively ignores, instead of reduces, error in heuristic estimate. The principle is illustrated in Figure 3.

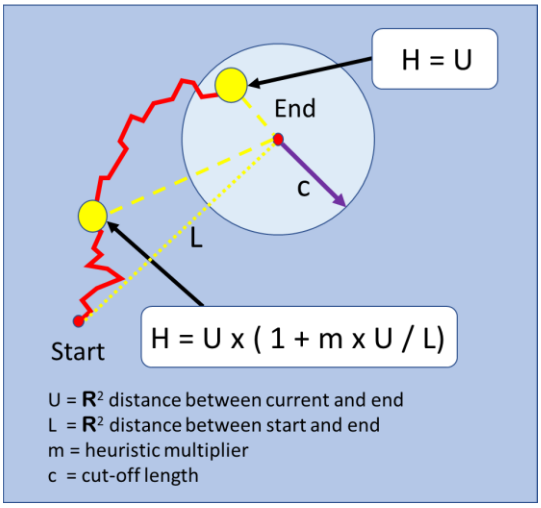

Figure 3 Multiplier-accelerated A* algorithm

The heuristic cost (H) in MAA* is a modulated $\mathbf{R}^2$ Euclidean distance that decreases as the current search point becomes closer to the end point.

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

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

In a canonical A* algorithm, when ϵ is large, the search jumps back to earlier paths in the open list and restarts frequently due to wrong cost estimation. If degree of anisotropicity is high, the search may even return to early search nodes from late stage search nodes because the range of estimated cost of the early nodes can vary widely under these topologies. However, restarting from very early nodes will unlikely lead to a better path due, again, to large ϵ. Therefore, the rationale of Equation (1) is to exaggerate the heuristic cost for nodes far away from the destination, and hence reduce the chance of jumping back to very early nodes during the later stages of the search. Effectively, MAA* reduces the search space by ignoring some error-driven search incentive when those incentives would be unlikely to lead to better paths. However, it is important to realize that MAA*, unlike A*, cannot guarantee optimality since the modulated heuristic cost H may over-estimate actual cost and hence may violate admissibility criterion of A*[18]. To trade-off between optimality and computation complexity, user can choose different values of m and c, which, respectively, define the level of exaggeration of the heuristic cost of distant nodes and the region in which to resume, if preferred, to standard A* algorithm.

Figure 3 and Table 1 illustrate the performance of MAA* algorithm. Computation time and optimality are used as metrics. Although computation time is not an authoritative measure – it will vary with coding quality and machine loading – the results can still clearly demonstrate the advantage of the new algorithm.

Calculations are based on a 30 x 30 demonstrative map, with a cost function composed of (i) path length, (ii) incremental costs due to changing elevation in routing, and (iii) accessibility of the locations along the path. Similar results are observed with a larger and more realistic 300 x 300 map (Figure 4). Details are provided in [9].

When c is fixed at 10 and m is small (0.1, 0.2 or 0.5), the path identified by MAA* is very close to (or exactly the same as) the global minimum found by standard A* algorithm. Computation time has reduced but remains in the same order of magnitude.

When larger m is used (1 or 2), location the path identified deviates remarkably from the global minimum but the cost is not very different. On the other hand, computation time has been reduced very significantly. In other words, it is a near-optimal path that requires much less time to identify.

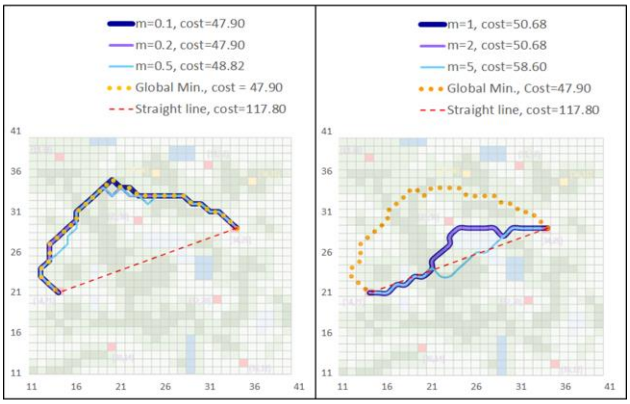

Figure 3 Optimal paths connecting (14,21) and (34,29)

For practical purposes, small cost over-estimation should be acceptable since maps created from surveys or aerial (satellite or unmanned aerial vehicle) photography will also have intrinsic tolerances at comparable level.

Table 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%

In the above example, MAA* has significantly accelerated (near) optimal path finding under anisotropic search space, thereby accelerating networked rural electrification routing.

III. LIMITATION OF MULTIPLIER-ACCELERATED A* ALGORITHM

Generally, MAA* remains effective even for large, anisotropic map. However, the search space reduction strategy used by the algorithm may not work well for some very complex topologies.

In Figure 4, there are two target connections A and B on a 300x300 grid. Connection A is a typical connection and MAA* can efficiently identify the (near) optimal path for connecting the villages. Connection B, however, is very different. Both start and end nodes, although not far away, are lying within a region of low accessibility (i.e. the blue rectangle). In addition to topographical variation, many locations in this region are difficult to access due to natural or human-related reasons. High cost nodes within the region can lead to drastic path cost increase in each step of movement inside the region; i.e. ϵ is large.

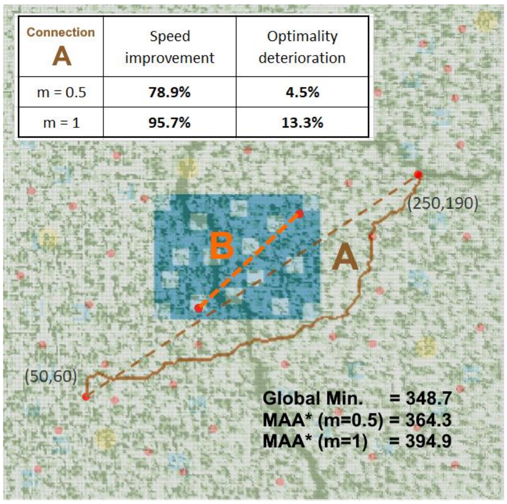

Figure 4 Performance of MAA* on a 300 x 300 map

If human intuition is used for routing, one would likely exit the region as quickly as possible to skirt the difficult area. This is consistent with results using MAA* (Figure 5, for *m*=5 and 10). Unfortunately, the MAA* computation time for this case is extremely long even for relatively large *m*.

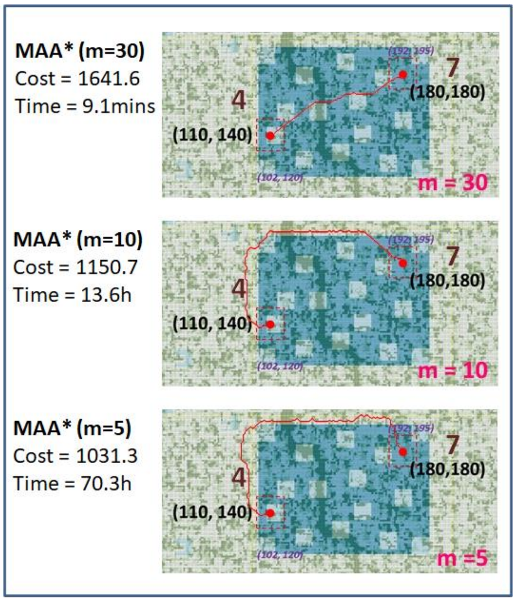

Figure 5 Searching costly zone

As in all cases, large ϵ will lead to exponentially increasing computation burden as per $O(b^{\epsilon d})$. However, this specific problem can be qualitatively understood as a high frequency switching, or path oscillation, between search paths inside and outside of the difficult region. High cost paths within the region have made the long paths outside of the region comparatively inexpensive. As the search within the region begins, it soon encounters move so expensive that the algorithm infers that some paths outside of the region could be possibly be less expensive. The algorithm then commences to explore these outside paths. The outside search remains favorable for many steps because the step cost increment, in general, will be less than the step cost increment within the region. However, after many steps, the outside exploration accumulates sufficient path cost that it becomes more expensive than the previous search path within the region. The algorithm then jumps back into the region to continue the search. However, since the single step cost in the region is large, cost of the inside search path will soon become more expensive than outside paths stored in the open list. The algorithm will again jump to the outside paths and continue for quite some more steps, and then jump back to inside paths. The process repeats inefficiently in this mode. In fact, using MAA* (m=5) to find the optimal path for Connection A only creates a closed list (i.e. nodes being completely explored) of size 557, but the corresponding list size for Connection B is 22,555. This means that only 0.6% of the search space needs to be explored in case of Connection A, but 25% in case of Connection B despites it appears to be much shorter.

Thus, even with MAA*, computation time is too long to be practical unless *m* is very large. However, when *m* is large, optimality of the solution is questionable. As shown in Figure 5, MAA* can complete the computation in about 9 minutes when m=30, but the solution is obviously far from optimal. When smaller *m* is used, optimality is improved but computation time is too long (Table 2).

Table 2 MAA* computation time for Connection B (on i5 machine with typical configuration)

Method	Optimal Cost Obtained	Computation Time	Size of Closed List	% of search space explored
MAA* (m=1)	932.4	664h	71518	79.5%
MAA* (m=2)	941.9	255h	45212	50.2%
MAA* (m=3)	975.5	160h	34733	38.6%
MAA* (m=4)	976.5	92.9h	26044	28.9%
MAA* (m=5)	1031.3	70.3h	22555	25.1%
MAA* (m=10)	1150.7	13.6h	9258	10.3%
MAA* (m=30)	1641.6	0.15h	973	1.1%

IV. ADAPTIVE MULTIPLIER-ACCELERATED A* ALOGRITHM

To tackle difficult regions, the adaptive multiplier-accelerated A* (AMAA*) algorithm has been developed.

As illustrated in section III, the problem originates from frequent switching between inside and outside search paths, which is fundamentally rooted at the large ϵ in the difficult region. Thus, instead of performing search solely based on cost comparison, AMAA* uses a directed search approach based on a projection of ϵ. This is done by adaptively defining and inserting intermediate nodes (P1 and P2 in Figure 6) according to the specific structure of the search space, and then applying MAA* algorithm to the newly constructed line segments.

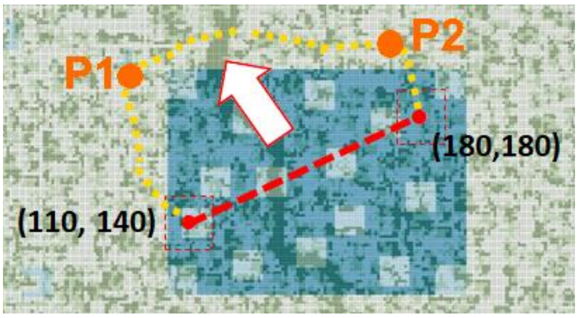

Figure 6 Principle of AMAA*

Intermediate node(s) are defined based upon (i) location desirability, and (ii) minimization of ϵ.

If one starts at (110, 140), an intermediate node on its slight right is more desired than a node on slight left because the former is closer to the destination (180, 180). Similarly, a node slightly above is preferred over a node slightly below. For minimal ϵ, direct calculation is possible but computational quite heavy. Therefore, it is indirectly measured by examining the coarse cost of movement in a number of directions. Scores are assigned to potential candidates based on the two criteria. The method is illustrated in Figure 7.

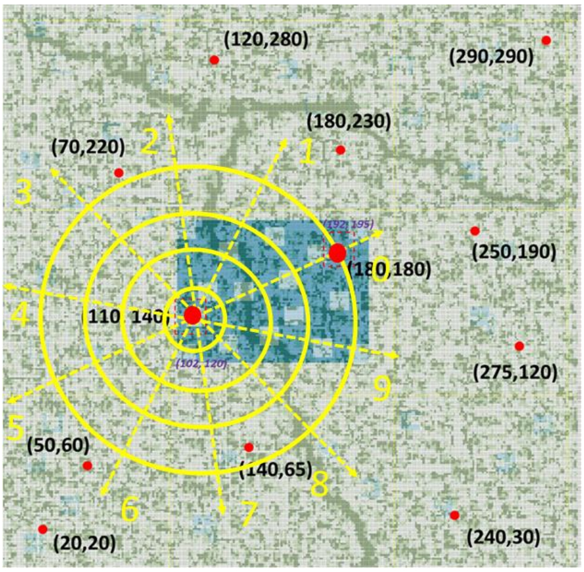

Figure 7 Screening for intermediate node

(i) Place k evenly distributed concentric circles on the start node such that the final circle passes through the end node;

(ii) Draw a line passing through the start and end nodes. Label the intersection closest the end node as "0";

(iii) Draw r-1 lines passing through the start node so that the concentric circles are evenly divided. Label each radii sequentially.

This creates $2kr$ points formed by the intersections of the lines and circles. These are potential candidates for intermediate nodes. For convenient programming, evaluation is done as follow.

(iv) For the $2r$ nodes on each circle, calculate the $\mathbf{R}^2$ distance to the end node, compared to the $\mathbf{R}^2$ distance between start node and end node to create a normalized score, L. A negative change indicates movement towards the destination and will be assigned with a low L. In this scheme, a low score indicates high desirability. Normalization is necessary because measured parameters will be different for different circles;

(v) For the $2r$ nodes on each circle, costs for moving from start node to these nodes are calculated and normalized by corresponding $\mathbf{R}^2$ distance, and denoted as D. Thus, by examining the change in D along the k nodes on one of the $2r$ arrows, one can coarsely deduce the difficulty for moving in that direction and hence estimate whether ϵ is increasing or decreasing;

(vi) The final score for each point is obtained by multiplying L and D.

Due to length limitation, computational details are not discussed in this paper. In addition, alternative methods for creating additional intersections that enhance optimality are also skipped. However, Tables 3 and 4 illustrate how the method works in practice.

With k=10 and r=5, there will be 100 potential candidates. The analyses are tabulated in 10 tables based on the k circles, and summarized in one additional table. Table 3 is the second of the 10 tables, and serves as an illustration of the format and information contained.

Table 3 Analysis of circle 2

Output: Circle 2							
Direction	X	Y	Distance to end	Change in distance wrt start	L Score	D score	Final Score
0	124	148	64.50	-20.00	0.70	1.60	1.12
1	116	154	69.08	-14.30	0.78	1.10	0.86
2	106	155	78.11	-3.10	0.93	1.30	1.21
3	98	150	87.32	8.30	1.36	1.00	1.36
4	93	141	95.34	18.30	1.91	1.00	1.91
5	96	132	96.75	20.00	2.00	1.10	2.20
6	103	125	94.63	17.40	1.86	1.60	2.98
7	113	124	87.32	8.30	1.36	2.20	2.99
8	121	129	77.99	-3.30	0.93	1.60	1.49
9	126	138	68.41	-15.10	0.77	1.70	1.31

By combining the 10 tables, a summary of final score for all 100 candidates is obtained (Table 4).

Table 4 Summary of final score

Direction	Circle 1	2	3	4	5	6	7	8	9	10
0	1.12	1.12	1.61	1.68	1.96	2.24	2.31	2.52	2.66	2.59
1	0.92	0.86	1.09	1.56	1.82	2.00	2.27	2.54	2.49	2.38
2	0.93	1.21	2.14	2.91	3.14	3.26	2.88	2.81	2.91	2.65
3	2.64	1.36	1.38	1.55	1.56	1.44	1.45	1.46	1.47	1.48
4	5.54	1.91	1.89	1.88	1.88	1.88	2.07	1.88	2.07	1.88
5	5.20	2.20	2.20	2.40	2.20	2.40	2.40	2.40	2.40	2.20
6	6.37	2.98	3.35	3.20	2.98	2.99	3.01	2.99	2.82	2.63
7	4.80	2.99	3.67	3.41	3.29	3.36	3.23	3.11	3.28	2.96
8	2.35	1.49	1.97	2.28	2.21	2.02	1.92	1.84	1.94	1.67
9	1.52	1.31	1.92	2.57	2.61	2.80	3.16	3.44	3.65	3.82

Inspecting all the directions, it has been found that the scores in direction 3 decrease quickly and remain stable after decreasing, even though the initial score at circle 1 is high. This indicates that moving along direction 3 is (i) not expensive (*D* score) and (ii) not moving too far away from the destination (*L* score). In contrast, early scores for direction 0, 1 and 2 are low, but increase markedly when moving toward the outer circles; these directions are expensive and should be avoided. It is to be noted that direction 8 is also a possible choice, but further illustration is skipped due to space constraints.

It is desirable to move to low score area as soon as possible. Since direction 3 attains a low final score at circle 2, the intersection is chosen as an intermediate node. Coordinates of this node, according to Table 4, are (98, 150). The method can also be applied to the end node (180,180), obtaining the intermediate point (162,196). MAA* algorithm is then applied to (110,140)-(98,150), (98,150)-(162,196), and (180,180)-(162,196) to find the final path (Figure 8)

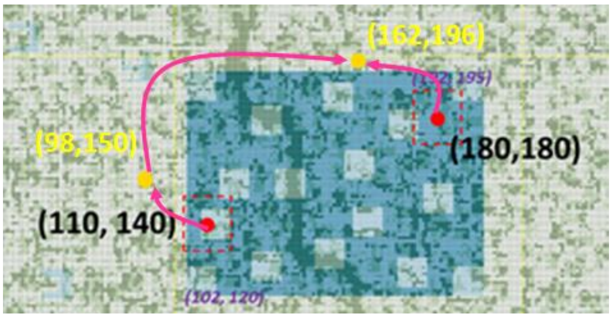

Figure 8 Optimal path finding by AMAA*

Results obtained by using AMAA* algorithm are summarized in Table 5.

Table 5 Computation results, AMAA*

From		(110,140)	(98,150)	(180,180)	TOTAL	
To		(98,150)	(162,196)	(162,196)		
MAA*(m=1)	Cost	207.9	231.9	495.1	934.8	AMAA* (m=1)
	Time (sec)	48.3	5511.5	630.5	6190.3	
MAA*(m=2)	Cost	207.9	243.4	495.2	946.4	AMAA* (m=2)
	Time (sec)	35.0	1144.6	505.5	1685.0	
MAA*(m=5)	Cost	207.9	264.8	501.7	974.3	AMAA* (m=5)
	Time (sec)	24.0	105.5	289.2	418.7	

Compared to Table 2, speed improvement is achieved by AMAA*. When *m*=1, MAA* and AMAA* are delivering the same optimality (932.43 vs 934.83), but the former took 664h to compute while the later only took 1.7h. With larger *m*, the improvement is even more significant. For example, AMAA* provides both better optimality and computation efficiency when *m*=5. Table 6 summarizes the differences.

Table 6 Comparing AMAA* to MAA*

Method	Optimal Cost Obtained	Computation Time	Size of Closed List	% of search space explored
MAA* (m=1)	932.4	664h	71518	79.5%
AMAA* (m=1)	934.8	1.72h	4496	5.0%
MAA* (m=2)	941.9	255h	45212	50.2%
AMAA* (m=2)	946.4	0.47h	2663	3.0%
MAA* (m=5)	1031.3	70.3h	22555	25.1%
AMAA* (m=5)	974.3	0.12h	1324	1.5%

As qualitatively discussed in section III, frequent switching between paths inside and outside of the high cost region causes significant computational load. This section quantitatively demonstrated this excess load, and provided an effective method to avoid it. With AMAA*, it is practically feasible to solve optimal path finding problems under very complex topography. However, it should be emphasized that AMAA*, like any other heuristic search methods, cannot guarantee optimality.

V. Conclusion

Achieving SDG7 by providing affordable, reliable, sustainable, and modern energy to all in 2030 is challenging. Networked rural electrification can potentially accelerate the process by reducing system cost, enhancing reliability, and offering installation flexibility. However, designing the required optimal network under some complex topographies could be computationally prohibitive. In this paper, it has been demonstrated that AMAA* algorithm can resolve the computation issue for some complex situations, and hence facilitate networked rural electrification. Since AMMA* focuses solely on optimal path finding, further works of developing an integrated framework for complete network synthesis is desirable.

References

[1] United Nation, "Transforming our world: the 2030 Agenda for Sustainable Development", United Nations, A/RES/70/1, 2015
[2] IEA, IRENA, UNSD, WB, WHO, "Tracking SDG 7: The Energy Progress Report 2019", Washington DC., 2019
[3] United Nations, "The Sustainable Development Goals Report 2019", United Nations, 2019
[4] H. Zerriffi, *Rural Electrification – strategies for Distributed Generation*, Springer, 2011
[5] 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.
[6] 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.
[7] 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.
[8] J. Petersa, M. Sieverta, and M. Tomanc, "Rural electrification through mini-grids: Challenges ahead," *Energy Policy*, Vol.132, 2019, pp.27–31
[9] J. Li and D. Zimmerle, "Designing Optimal Network for Rural Electrification using Multiplier-accelerated A* Algorithm," *2019 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, Macao, Macao, 2019, pp. 1-5
[10] S. J. Russell and P. Norvig, "Section 3.5: Informed (Heuristic) Search Strategies & Section 3.6: Heuristic Functions", in Artificial Intelligence: A Modern Approach, 3rd Ed., Pearson, 2015, pp. 92-108
[11] 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.
[12] S. Rabin, "A* speed optimizations," in *Game Programming GEMS*, , Charles River Media, 2000, pp.264-271.
[13] P. Yap, N. Burch, R. Holte, and J, Schaeffer, "Any-Angle Path Planning for Computer Games", in Proc. 2011 AIIDE, pp.201-conference, 2011.
[14] D. Harabor and A. Grastien, "Online Graph Pruning for Pathfinding on Grid Maps", in *Proc. 2011 AAAI*, pp.1114-1119.
[15] R.E. Korf, M. Reid and S. Edelkamp, "The Time Complexity of iterative-deepening-A* ", Artificial Intelligence, Vol. 129, 2001, pp. 199-218
[16] A. Felner, R.E. Korf and S. Hanan, "Additive Pattern Database Heuristics", Journal of Artificial Intelligence Research, Vol. 22, 2004, pp. 270-318
[17] G. Sharon, A. Felner and N. Sturtevant, "Exponential Deepening A* for Real-Time Agent-Centered Search", Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14), 2014, pp. 871–877
[18] P. Hart, N. Nilsson and B. Raphael, "A Formal Basis for Heuristic Determination of Minimum Cost Paths", IEEE Transactions of Systems Science and Cybernetics, Vol. 4, 1968, pp. 100-107