DISSERTATION

NETWORKED RURAL ELECTRIFICATION – OPTIMAL NETWORK DESIGN UNDER COMPLEX TOPOGRAPHY

Submitted by

Jerry Chun-Fung Li

Department of Systems Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2022

Doctoral Committee:

Advisor: Peter Young
Co-Advisor: Daniel Zimmerle

Jim Cale
Jeni Eileen Cross

ABSTRACT

NETWORKED RURAL ELECTRIFICATION – OPTIMAL NETWORK DESIGN UNDER COMPLEX TOPOGRAPHY

The 7th of United Nations' Sustainable Development Goals (SDG7) aims to "ensure access to affordable, reliable, sustainable and modern energy for all" by 2030. While substantial progresses have been made in the last few years, 759 million people in rural areas still have no or limited access to electricity. Due to the distances and geographical complexity of rural areas, providing electricity to this unserved population is very costly. As IEA recently pointed out, rural electrification is increasingly costly. With the current electrification approach, it is expected that 660 million people will remain without electricity access by 2030. In addition, accurate planning for small rural power system is difficult as both demand and energy resource forecasts are highly uncertain. Thus, achieving SDG7 is very challenging.

In this research, a Networked Rural Electrification framework has been proposed. This approach can potentially accelerate SDG7 by reducing system cost, enhancing reliability, and offering installation flexibility for small communities in remote areas. In this framework, villages and generation facilities are connected via an optimal, low voltage network that can be built with inexpensive poles and cables. To make this approach economically feasible, cost for building the network is crucial. A specific difficulty associated with this approach is the anisotropicity of search space for optimal design of the power distribution network, which results from complex topographical variations in these rural areas. Traditional optimization methods are not suitable for designing this network because of computational complexity, accuracy requirement, and practical implementation considerations. To address the issues, new computation methods and tools have been developed. These include (i) Multiplier-accelerated A* (MAA*) and (ii) Adaptive Multiplier-accelerated A* (AMAA*) algorithms, which resolve the computational complexity

problem by significantly reduce computation time while maintaining good optimality, and (iii) Levelized Interpolative Genetic Algorithm (LIGA) which, when used in conjunction with A*, MAA*, or AMAA*, provides viable alternative plans to tackle unexpected route change problem right before or even during project implementation, and (iv) a fuzzy rule-based system for further network topology optimization.

ACKNOWLEDGEMENTS

This research would not have been possible without the support of many people. I would like to thank my advisor Peter Young and co-advisor Daniel Zimmerle for their invaluable guidance and feedback. I especially want to thank Daniel for inspiration, for reading my numerous revisions, and for all unconditional support as I moved along this rewarding journey. I am grateful to committee members, Jim Cale and Jeni Eileen Cross, for their support and taking time to serve on my committee.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 – INTRODUCTION

The 7th of United Nations' Sustainable Development Goals (SDG7) [1] aims to "ensure access to affordable, reliable, sustainable and modern energy for all" by 2030. Globally, the population without electricity access fell from 1.2 billion in 2010 to 840 million in 2017 [2] [3]. However, there is still a substantial rural-urban divide in electrification: 87% of the unserved population live in rural areas and 21% of the global rural population has no or limited access to electricity. Due to the distances and the complexity of grid-attached electrification in rural areas [3], further electrification will require increasingly marginal investments to reach these remote areas. According to the latest report [4], 759 million people in rural areas still have no or limited access to electricity. With the current electrification approach, IEA projects 660 million people will remain without electricity access by 2030 [5]. In this research, we develop new methods for enhancing rural electrification efforts and reducing investment burdens.

1.1 Rural Electrification Today

International organizations and researchers have explored the benefits of electrification for rural areas [6] [7] [8] [9] [10]. In general, there is a clear linkage between availability of electricity and rural development. While richer households may be more ready to pay for electricity and may see more immediate benefits, poorer households have also shown some willingness to prioritize electrification expenditures. However, loads in newly electrified areas remain difficult to predict, complicating system planning.

1.1.1 Current Practices in Rural Electrification

Due to physical and economical barriers, it is usually uneconomical to connect these areas to the main grid. Many international organizations and policy makers believe renewable energy, especially solar

photovoltaic, is a feasible solution for rural electrification [1] [3]. Based on the multi-tier framework proposed in [6], it has been suggested that Solar Home Systems (SHS) can be used to fulfil Tier 1 (min. 3W and 12Wh/day for a household) and Tier 2 (min. 50W and 200Wh/day for a household) [6] [11]. Grid-connections or minigrids are recommended for higher consumption levels [6] [12]. While this framework is widely accepted, research has shown that there are opportunities for significant improvements in economics, reliability, and investment risk management [13] [14] [15] .

1.1.2 Limitations of the Multi-tier Framework

The multi-tier framework considers each village or user independently – i.e. it considers delivery of electricity to each household as an independent activity. While this approach is straightforward, there are two important factors that may hinder the current electrification efforts. These factors are:

1. Location of the village, which affects the available energy resources and hence generation efficiency, and
2. Unique difficulty in system planning for small population spread over large geographical areas, often with complex topographic features.

In a rural context, village locations were historically selected to provide ready access to water, soil, food storage, defense, and similar concerns. These locations are often sub-optimal locations for solar or wind power generation. Developing these energy sources within the village may therefore require higher capital investment and land usage than the investment required in nearby locations with better renewable energy conditions. Furthermore, the acquisition of land for energy generation near agricultural villages may raise tensions with landowners that are counterproductive.

In addition, prior studies have demonstrated the difficulty of estimating loads when planning initial electrification and early load growth [12] [16] [17] [18] [19]. Capacity planning based on statistics is not reliable for villages due to low geographical density and small customer count. Longer term demand

depends heavily on population change and economic growth, and is highly uncertain. Future urbanization plan of the neighborhood areas, changing ecological conservation policies, and many other non-technological factors can also significantly affect the outlook of rural electrification [20] [21]. In particular, villages may grow differentially, with growth occurring in some villages, outstripping installed power generation, while growth may be retarded in other villages, idling power generation. Furthermore, while energy resources statistics for a big area are reliable, data for a small geographical coverage may not be as useful and even may not be available. Due to uncertainties in both demand and supply, it is not uncommon to over-invest or under-invest for rural electrification projects [22] [23]. Such mismatches increase the need to move power between villages.

1.1.3 Illustrative Example: Rural Electrification in Mengtougou, China

Rural electrification in China is at a more advanced stage than most developing countries. In 2015, the country had already declared 100% electrification rate [24]. However, detailed statistics about service level are not publicly available. Based on installed capacity, average level of electrification for Chinese rural, off-grid households is approximately tier 3+ in the UN multi-tier framework (min. 200W and 1kWh/day for a household) [25]. Based on the total rural consumption (excluding productive use) outlined in national statistics [26], another estimate of rural electricity consumption is 2000-3000KWh per household per year. Since this total consumption figure includes some grid-connected rural households which likely have higher energy consumption, average consumption of rural households without grid connection could be lower. However, the 2000-3000KWh/year average consumption is large enough that even off-grid households are likely at tier 3+. Therefore, the two estimations indicate that the service level of the average non-grid-connected rural household in China is approximately tier 3+. Such capacity can support basic services such as lighting, TV, phone charging and some light appliances for daily life, with some support for higher-load appliances such as refrigerators. For developmental purposes, however, this capacity level is not sufficient

for supporting productive use activities, such as energy-intensive farming or raising livestock for commercial purposes.

The Mentougou district of Beijing, China, Figure 1.1, represents a typical scenario for rural electrification, and is a near-standard example of rural electrification in China.

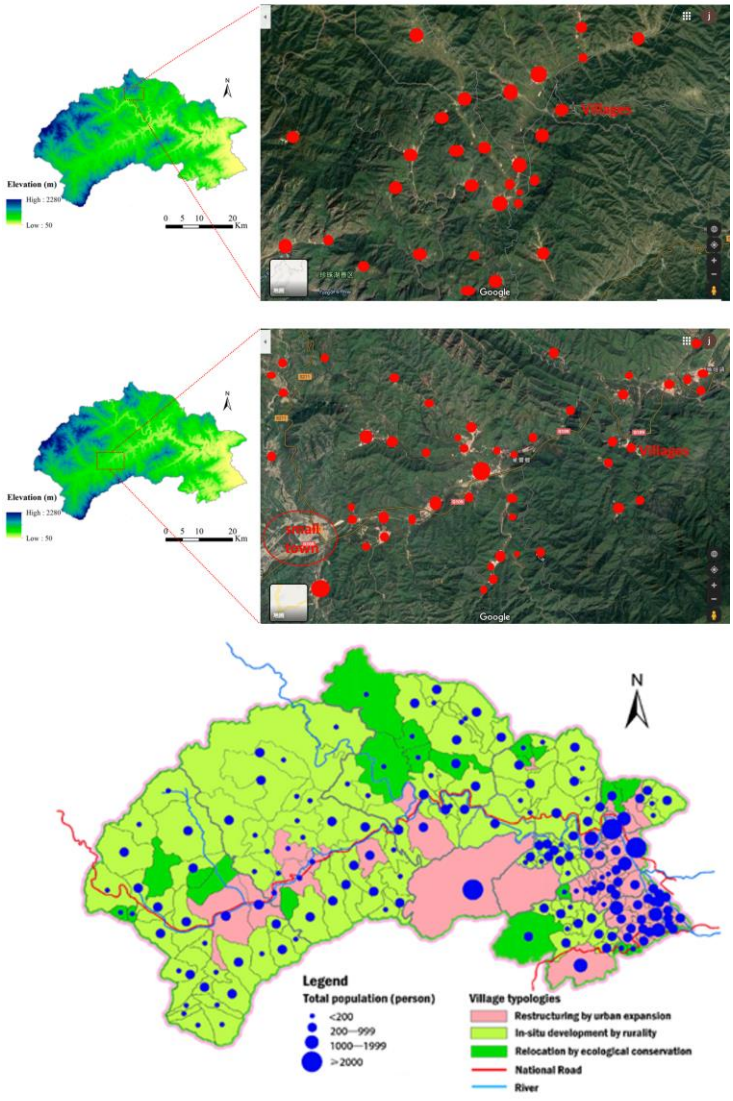


Figure 1. 1 Geographical structure and population distribution of Mentougou district of Beijing (source: [20] [21]).

Villages in the southeastern area have growing populations and have been gradually urbanized. The central grid connects these villages and has been improved to support productive activities. In contrast, most other areas of the district have lower population, and population growth. Further development is becoming more difficult in these areas because of limited electric power. Adding new capacity by increasing SHS penetration or building larger, local minigrids, is typically not economically competitive for productive use activities. Connecting to main grid is also impractical given the remote location and geographical complexity.

1.2 Overview of the Research Proposal

As pointed out in Section 1.1.2, suboptimal location of villages and difficulty in small system planning are barriers for effective rural electrification. To address these issues, this research proposes a flexible and cost-effective method, referred as Networked Rural Electrification, for locally interconnecting villages.

1.2.1 Proposed Framework

Unlike conventional minigrids for tier 3 and above [6], this approach connects villages, potentially including local generation, with each other and with centralized generation. The approach emphasizes low voltage interconnects and near-optimal routing in areas with dispersed loads and substantial topological variations, shown schematically in Figure 1.2.

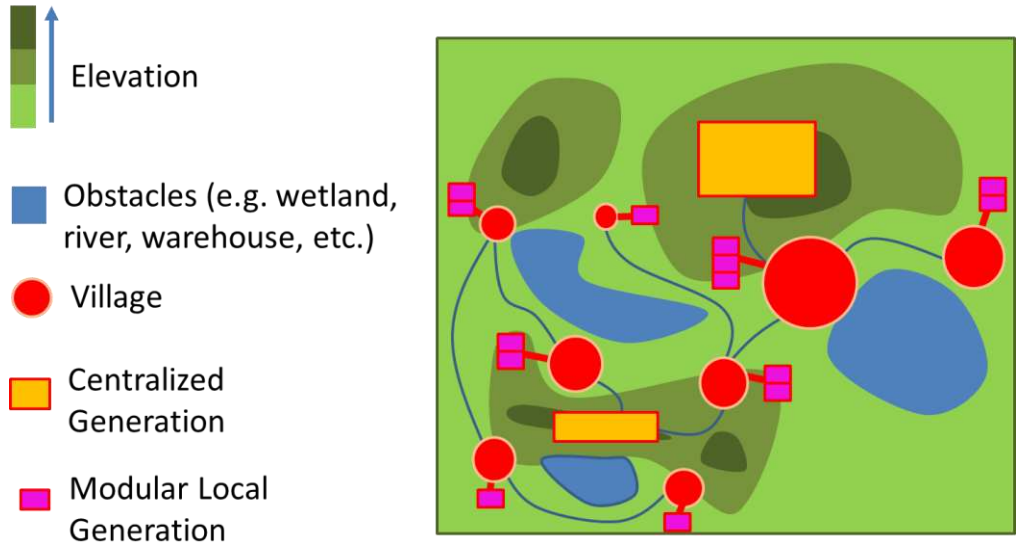


Figure 1. 2 **Schematic view of the proposed networked electrification method. Villages may contain local generation resources, complimented by centralized generation placed in areas with better resource potential and fewer land conflicts. Unelectrified areas in the developing world are most often remote, with challenging topographical constraints, as indicated the elevation and obstacle shading in the figure.**

The networked rural electrification model proposed here is focused on resolving this capacity-economic problem using local generation, primarily from renewable resources. To apply this networked model, the first step is to discretize the map of the region of interest into a set of 2D grid squares to enable computation. Each coordinate pair is then assigned an elevation and an ease-of-access metric (Figure 1.3). It should be pointed out that using an elevation metric instead of a full 3D representation is a practical computation consideration. In this case, for example, we are assuming 10m x 10m resolution (i.e. the map covers 3km x 3km). This resolution supports low voltage power distribution that can be easily implemented in rural environment with poles and cables. For low voltage connections, optimally positioned poles cannot be too far apart. Here we assume a maximum span between poles of $\approx$15 m (diagonal on a 10 x 10 m grid).

In contrast to the accessibility of individual grid squares, elevation usually does not need to be represented by a continuous variable. It is simpler, and computational advantageous, to represent elevation by a few discrete levels (this work uses 3). The number of levels can be increased as needed, and

computational advantages of the proposed algorithm are even more apparent with increased number of levels.

Additionally, grid squares are annotated with locations that have strong renewable resources, and locations where generation can be added to individual villages with proper land and renewable resources. In all cases, we assume excess power could be transmitted to nearby communities when available. Figure 1.4 illustrates this type of schematic interconnection using a portion of the map in Figure 1.3.

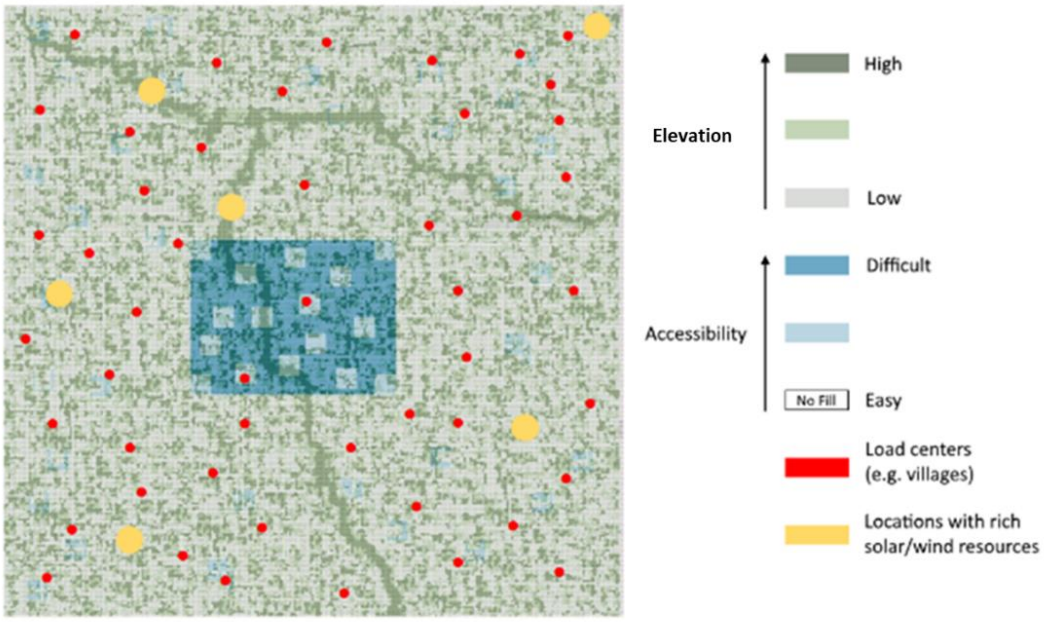


Figure 1. 3 **Discretized map indicating both elevation (greens) and accessibility (blues) metrics. Load centers are indicated coincident with villages in this example, although this is not always the case. Ideal locations for renewable resources are also marked. Size of village and generation locations are shown larger than the operational map for illustration; in practice, each point is assigned to one grid square.**

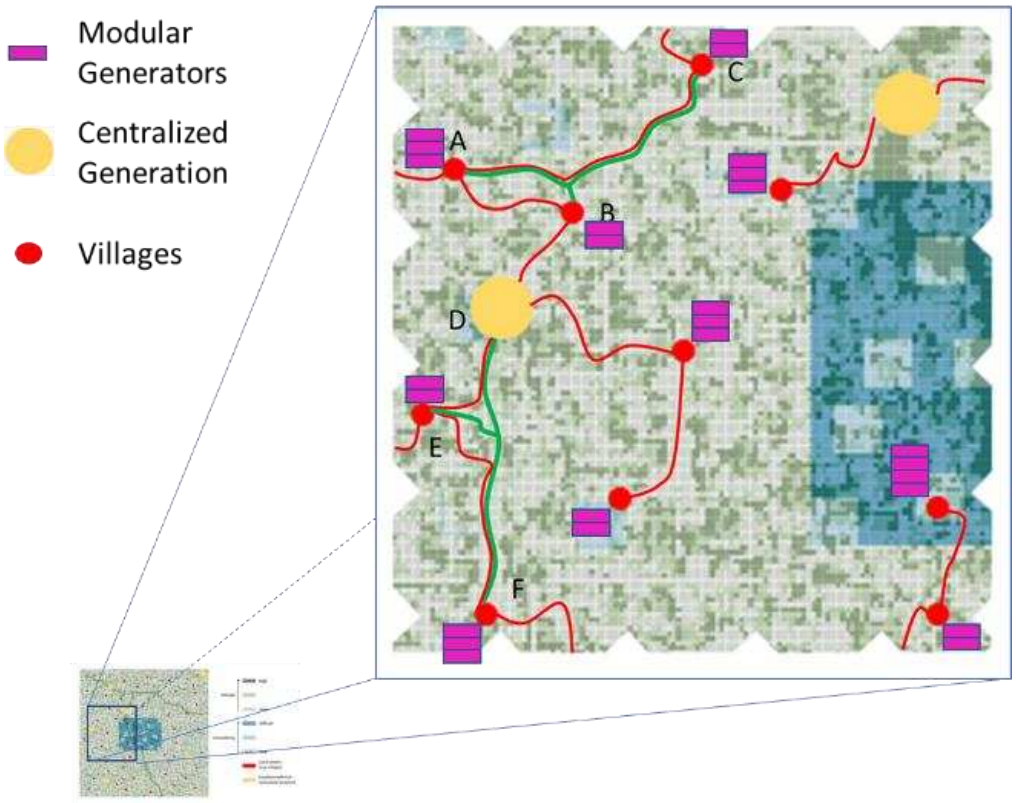


Figure 1. 4 **An example of power supply interconnections, using a subset of the study map shown in Figure 1.3. Rectangular icons indicate the land use and resource potential for local generation near each population center. Red lines reflect notional routes for power connections between villages (red) and centralized generation (yellow).**

The objective of the work presented here is the routing of the power connections (red lines) in Figure 1.4: Given any two points of interest, what is the optimal, or near-optimal, route between the points? Given near-optimal routes between any two locations in Figure 1.3, numerous methods are available to synthesize the minimum cost network [27]. Given N points of interest there are $\frac{1}{2}N(N-1)$ possible interconnections that must be calculated, although this number can possibly be reduced by some trimming of pairs. Therefore, the pair-wise route planning algorithm must be both efficient and require little-to-no human supervision.

After finding the optimal routes, the next step is designing the network design. To minimize cost, it is assumed that the network will not include redundancy, i.e. only $N-1$ wires will be used to connect N points. Simple combinatoric analysis reveals the number of acceptable combinations under this arrangement

(Table 1.1). Acceptable combinations refer to those with all the p points connected. Even under the simple case of connecting 10 points (e.g. 1 centralized generation site and 9 villages) with 9 wires, there are over 220 million acceptable configurations. Hence, determining optimal configuration by exhaustive search is prohibitive, and determining by intuitive design is not reliable.

Table 1. 1 Acceptable configurations

No. of points	No. of wires	Acceptable combinations
3	2	3
4	3	16
5	4	135
6	5	1581
7	6	23604
8	7	427260
9	8	9.08×10^{6}
10	9	2.21×10^{8}
11	10	6.09×10^{9}
12	11	1.86×10^{11}
13	12	6.29×10^{12}
14	13	2.32×10^{14}
15	14	9.25×10^{15}
20	19	2.62×10^{24}
30	29	9.70×10^{42}

This work reviews existing computational methods potentially applicable to our problem, examines their shortcomings, and systematically develop a set of new algorithms suitable for networked rural electrification design.

1.2.2 Relevant Theories and Method Developments

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:

1. finding (near) optimal paths for connecting any two villages, and also optimal paths connecting each village to the centralized generation sites, and
2. 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. Efficient algorithms that significantly reduce complexity are required. The following provides an overview about the methods developed in this research.

To perform the first task, the widely used A* search algorithm [28] appears to be a reasonable solution for finding the optimal paths. However, as will be illustrated in Section 2.2.2, 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 becomes more important if there are considerable uncertainties in input map and Monte Carlo simulations are required to ensure robustness of the solution. The multiplier-accelerated A* (MAA*) algorithm [29] [30] (Chapter 3) improves 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 large topological variations over short distances, or when path oscillation happens (Section 4.3). Thus, an adaptive version of MAA*, or Adaptive Multiplier-accelerated A* (AMAA*) has been developed [31]. The algorithm significantly simplifies computations in certain complex regions with minimal impact on optimality. This greatly facilitates the analysis and design of optimal network for cost-effective electricity supply to users in remote, difficult-to-reach areas.

Most optimal path finding methods, including canonical A*, MAA* and AMAA*, will only give one optimal or near-optimal path for connecting the two target nodes. Although this is sufficient for initial design, it is not uncommon that contractors are required to change routings immediately before, or even during, implementation. This can be caused by property right disputes, unexpected ground conditions, stakeholder opinions, changes in land policy, etc. Therefore, it is highly desirable to generate a family of near-optimal paths instead of a single path, which can provide alternative routings and avoid significant redesign effort when encountering implementation problems. Levelized Interpolative Genetic Algorithm (LIGA) has been developed to satisfy this requirement [32].

Once all the optimal paths have been identified, basic minimum cost network can be obtained by standard method such as minimum spanning tree (MST) or capacitated minimum spanning tree (CMST) [27]. Further optimization based on the geometry of the routes is also possible. A fuzzy rule-based system is developed for this purpose [33]. Figure 1.5 summarize the design processes and methods developed for the proposed Networked Rural Electrification approach.

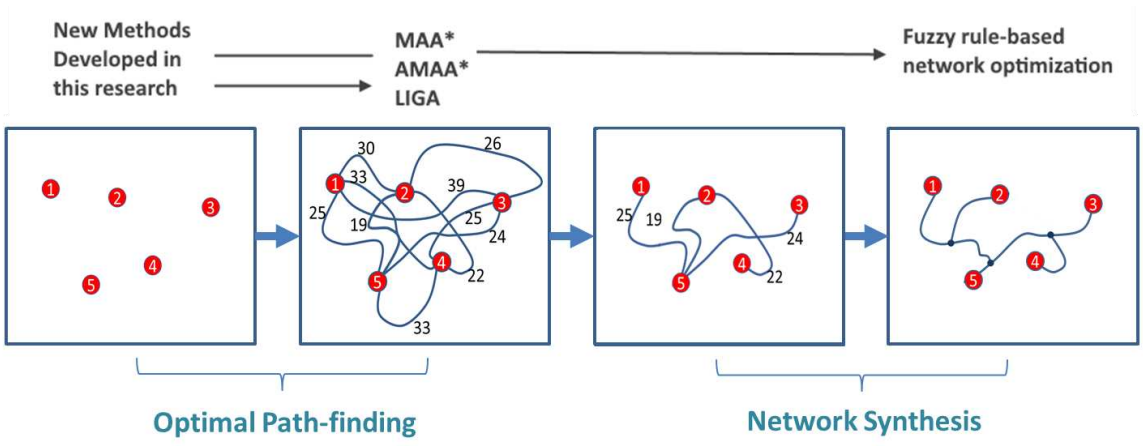


Figure 1. 5 Summary of design processes and methods developed for Networked Rural Electrification

1.3 Dissertation Outline

To make networked rural electrification practical, route planning must meet three objectives: (1) identify nearly cost-optimal paths in reasonable time, (2) account for political and sociological factors, and (3) run effectively on standard computer hardware with minimal user intervention. Regarding (2), a family of near-optimal routes is superior to a single, optimal solution, as it allows for late-stage change in routes after engagement with local communities.

This dissertation responds to these criteria and is organized as follows:

Chapter 2 qualitatively explores the effects of anisotropic search space, which is a result of complex topography in our application, to the standard optimization methods.

Chapter 3 provides more detailed discussions on why canonical A* algorithm is inefficient when applied to the rural routing problem, followed by the introduction of MAA* method. Numerical examples are used to illustrate the performances of these algorithms.

Chapter 4 discusses AMAA*, which is applicable for routing through some extra-difficult regions in the search space. The method effectively deals with a phenomenon we termed as “path oscillation” – a situation where the search process switches frequently between paths inside and outside of a difficult region due to the functioning of the A* algorithm.

Chapter 5 explains the LIGA algorithm and demonstrates how family of optimal paths can be obtained.

Chapter 6 begins with basic optimal network design using MST. A fuzzy rule-based inference system is then developed for further route optimization.

Chapter 7 summarizes the results and reviews the potential contributions of the networked rural electrification framework.

CHAPTER 2 – COMPLEX TOPOGRAPHY AND ANISOTROPICITY OF SEARCH SPACE

In networked rural electrification model, digital – i.e. discretized – map is used to represent the rural areas (Section 3.1.1) and is also the search space of optimal pathfinding. Complex topography of these areas lead to highly anisotropic search spaces and presents major challenges to classical optimization methods. In this chapter, we briefly examine how a characteristic of the search space, here termed 'anisotropicity,' leads to these challenges. Detailed discussions, examples and proposed solutions are provided in later chapters.

2.1 Cost Function and Characteristics of Search Space

For most optimization problems, optimal solutions can be obtained by minimizing / maximizing a specially defined cost function. Definition of the cost function depends on the problem and may vary in form depending upon the optimization method and problem formulation. For networked rural electrification, the cost function is defined by the actual cost for connecting two points on the map.

On a uniform flat plane, which is the simplest search space, cost for connecting any two points can be simply represented by the distance between them and the optimal path (i.e. lowest cost) is always the straight-line connecting the point pair. When simple, walkable / non-walkable obstacles are added to this search space, A* algorithm, together with some common acceleration methods, can effectively identify the optimal path [34] [35] [36] [37]. Under a more realistic rural conditions, however, properties of the search space change significantly over short distances, a property of the search space termed 'anisotropicity.' As will be illustrated in Section 2.2.2, if the search space is highly anisotropic, it will lead to high degree of computational complexity.

Typically, rural areas in need of off-grid electrification are in regions with difficult terrain, including mountains, heavy vegetation, and rivers that are difficult to cross. In these environments, determining the

optimal connection path between two points must account for changes in elevation and accessibility of every point along the connection. For the study problem, the cost of the $(i+1)^{th}$ movement in the A* search is:

$$C_{i+1} = [(z_{i+1} - z_i)^2 + (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{\frac{1}{2}} + w_1|z_{i+1} - z_i| + w_2 a_{i+1} \qquad (2.1)$$

where $x_i, y_i, z_i,$ are the 2D coordinates and elevation of the i^{th}point in the path, a_i is the accessibility of the i^{th} point on Figure 2.3, w_1, w_2 are weighting coefficients.

The first term in Eqn. 2.1 is the length of the movement from step i to step $i+1$ and is directly proportional to cost. The second term is the cost due to change in elevation which drives additional costs for the power distribution systems, and the third term is the cost due to accessibility of the destination grid square. The weighting coefficients are selected to weight elevation change and accessibility constraints similarly to distance, and depend on technical capability, equipment availability, materials used, and labor cost.

For this analysis the discretized grid is 300 grids square (i.e. $u, v \in [1 \dots 300]$ where u and v are discretized indices in the two horizontal directions), unit distance is assumed between rows and columns of grid squares (i.e. $x_{i+1} - x_i = 1, y_{i+1} - y_i = 1$), elevation is discretized to three levels ($z_i \in \{1,2,3\}$), and accessibility is discretized to three levels ($a_i \in \{3,10\}$). Weights are set as $w_1 = 6$ and $w_2 = 3$. The first weighting factor w_1 represents the cost penalty caused by unit change in elevation, and is a scaled cost due to both path length and installation difficulty. The weight depends on labor cost, equipment availability, skill level, etc., and therefore varies from region to region. The weight is usually > 1 , as costs are typically several times larger than the corresponding length-driven cost in practice. The second weighting factor w_2 , which must be scaled to interact with $w_1$and length-driven costs, relates accessibility of a node to cost penalty. The factor is chosen so that, at the first level (i.e. $a_i = 3$), the magnitude of the penalty is comparable to a modest change in elevation. This corresponds to manageable common situations, such as

movable obstacles, slightly unfavorable soil condition, etc. The second level (i.e. $a_i = 10$) is relatively large so the path-finding algorithm will very likely avoid this grid square unless it is absolutely necessary to go through it. Examples are wetland, areas that require special permitting, costly rights-of-way, etc.

In classical optimal pathfinding process, costs of moving in different directions along each point on potential optimal paths are evaluated and compared to a dynamic baseline in order to decide next movement. With this cost model, it is not difficult to visualize that the costs for moving in different directions can be very different. In other words, the search space is highly anisotropic. As illustrated in Section 2.2.2, this property will significantly increase computation efforts and make classical optimization methods inefficient.

2.2 Effect of Anisotropicity on A* Algorithm

To understand how anisotropicity affects A* pathfinding method, the principles of A* algorithm is reviewed, followed by a qualitative examination of the search process in an anisotropic space.

2.2.1 Principle of A* Path Finding

The A* algorithm is a well-recognized path finding method that is provably optimal provided the initial heuristic always underestimates the cost of the path [28]. The algorithm has been widely used for solving practical problems such as highway route planning. However, as will be shown in Section 2.2.2, the classic A* algorithm is computationally intractable when applied to continuously varying terrain, as is required for this problem.

Conceptually, A* algorithm is a step-by-step search procedure that moves along the "most probable lowest cost path". However, this "most probable lowest cost path" can change at each step as more knowledge about actual cost has been revealed while the algorithm is running. In A* algorithm, a total estimated cost (F) of a specific node on the map is used to determine when the "most probable lowest cost path" should change. The total estimated cost of a particular node represents the estimated minimum cost

for moving from the starting node to the final destination via this node. As such, the total estimated cost of a particular node is the sum of (i) the known minimum cost (G) for moving from the starting node to this node, and (ii) the heuristically estimated, unknown cost (H) for moving from this node to the final destination. Note that G is known because the A* algorithm begins computing from the starting node so the best path cost for arriving at that particular point has already been determined. Suppose the search is now moving along a current "most probable lowest cost path", A* will compute estimated total costs for moving to the final destination via the next immediate nodes in all possible movement directions. It will then compare the lowest one (CL) to a previous record, known as OPEN LIST, which containing information about total estimated costs of nodes explored earlier. If no candidate in the OPEN LIST has a total estimated cost lower than CL, the algorithm will continue to move on the current path in the direction of CL. If, however, one or more nodes in OPEN LIST has a total estimated cost lower than CL, the algorithm will switch to the lowest cost node in the OPEN LIST and search on a different "most probable lowest cost path". Note that the OPEN LIST will be continuously updated and fully explored points on the nodes are moved to another record known as CLOSED LIST. The process repeats until the search arrives at the final destination or the algorithm concludes that it is not possible to arrive.

To guarantee optimality, however, H must not over-estimate. That is, H should always be smaller than the actual cost for moving from a particular node to the final destination. Since $F = G + H$, H of earlier nodes on the search paths are gradually and partially replaced by G as the search moves towards the later points. If H is always smaller than the actual cost, higher-than-estimated F will be observed at some nodes along the search paths and hence some candidates in the OPEN LIST will have a lower F compared to all possible movements along the current path. The algorithm will therefore switch to a potentially better search path in order not to miss the real optimal. On the other hand, if H is sometimes over-estimating, some potentially better search paths in the OPEN LIST will be misrepresented as higher cost paths and therefore leading to the possibility of missing the optimal path. Thus, to ensure optimality, H must always under-estimate the actual cost. More rigorous theoretical treatments are found in [38] [39] [40].

To implement A* algorithm, one of the founders of the algorithm, N. Nilsson, has proposed the systematic procedures outlined in Figure 2.1 [40]. The figure is Nilsson's original language. Note that $\hat{f}$ in the procedures and F described above bear the same meaning. Nilsson used the symbol pair f and $\hat{f}$ to better present the theoretical arguments for A* algorithm, which is optional for the purpose of understanding our problem. In this research, procedures in the figure are implemented with Python 3.

(1) Put the start node s on a list called OPEN and compute $\hat{f}(s)$.

(2) If OPEN is empty, exit with failure; otherwise continue.

(3) Remove from OPEN that node whose $\hat{f}$ value is smallest and put it on a list called CLOSED. Call this node n. (Resolve ties for minimal $\hat{f}$ values arbitrarily, but always in favor of any goal node.)

(4) If n is a goal node, exit with the solution path obtained by tracing back through the pointers; otherwise continue.

(5) Expand node n, generating all of its successors. [If there are no successors, go immediately to (2).] For each successor n_i, compute $\hat{f}(n_i)$.

(6) Associate with the successors not already on either OPEN or CLOSED the $\hat{f}$ values just computed. Put these nodes on OPEN and direct pointers from them back to n.

(7) Associate with those successors that were already on OPEN or CLOSED the smaller of the $\hat{f}$ values just computed and their previous $\hat{f}$ values. Put on OPEN those successors on CLOSED whose $\hat{f}$ values were thus lowered, and redirect to n the pointers from all nodes whose $\hat{f}$ values were lowered.

(8) Go to (2).

Figure 2. 1 A* implementation procedures, Sources [40]

2.2.2 Efficiency of A* Algorithm

This section illustrates the performance of A* algorithm on the digitized test map shown in Figure 1.3. Such map can be created from digital terrain model (DTM) using data from satellites or unmanned aerial vehicle (UAV) survey. DTM has a long history and is very mature today. As discussed in [41] [42] [43] [44], the method has evolved from basic to sophisticated and is able to reliably identify objects, make

necessary corrections, support quick processing, etc. Thus, this work assumes a reliable digitized map can be practically obtained. Furthermore, since this work proposes to establish centralized solar generation facilities, DTM is useful for selecting location with strong solar radiation when reliable statistics for small areas are not readily available [45] [46] [47].

An important characteristic that significantly affects the efficiency of A* path finding is anisotropicity of the map. In Section 2.2.1, we have seen how A* algorithm decides to switch search path. Whenever there is a lower cost node available in the OPEN LIST, path switching takes place. The more inaccurate the heuristic estimate (H) is, the more frequently will switching occur. Thus, computational complexity of A* depends on the error of heuristic estimate, which in turn affects the average branching from each node along the solution path. Precise quantification of A* time complexity in an anisotropic search space is difficult. For large space with constant step cost, A* time complexity is described by $O(b^{\epsilon d})$ [38], where b is the branching factor – an average number of downstream explorations of each point on the path; $\epsilon = (h^* - h)/h^*$ is the error in heuristic estimate, where h^* the actual cost, and h the cost estimated by the heuristic from the node to final goal; and d is the solution depth – path length in our case. Even with constant step cost, the complexity of search increases exponentially with d when ϵ is finite (Figure 2.2, for illustration, average b is assumed to be 2.13).

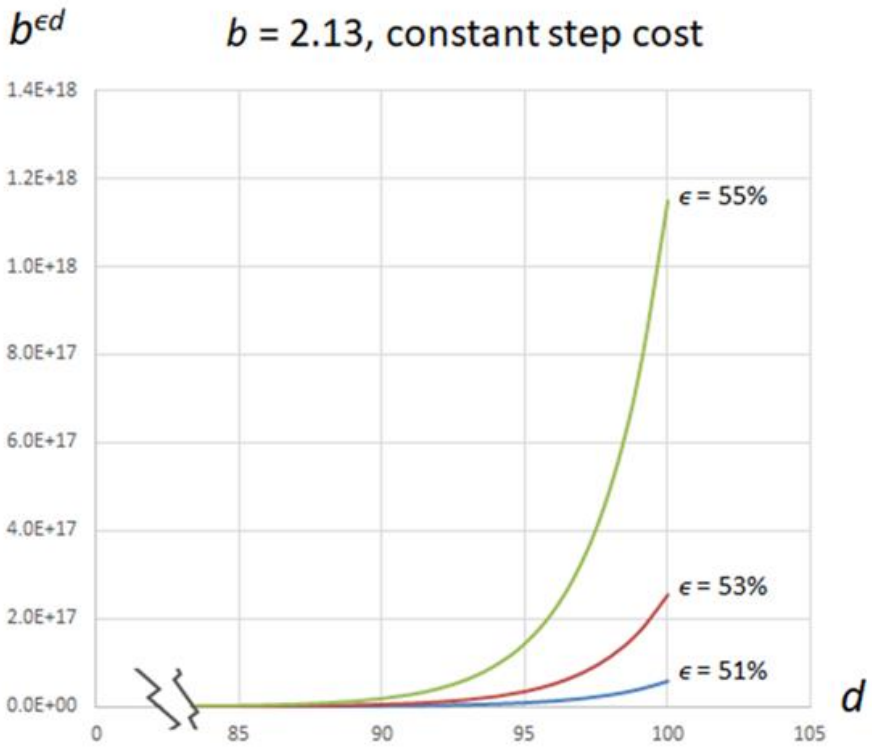


Figure 2. 2 **Time complexity vs search depth for the branching factor in the discretized map (b = 2.13) and assuming a constant step cost for illustrative purposes. A search depth of 50-250 is typical of many pairwise routings for rural electrification problems, while maintaining $\epsilon < 50\%$ is exceptionally challenging for the topology of the problem set.**

Due to significant topological variation and change in accessibility, it is difficult to define a heuristic function with small error over the entire search space while still meeting the A* requirement that the heuristic always underestimates the cost. Computational complexity described by Figure 2.3 assumes constant step cost. In our application, step cost can vary significantly. While researchers [48] [49] have developed more sophisticated methods to analyze the effect of ϵ on A* search, one would intuitively expect time complexity of our problem will grow even more rapidly than the constant step cost situation.

Figure 2.3 is an example of path switching under high degree of anisotropicity. A* search is used to find the optimal connection between (20,20) and (290, 290) in test map (Figure 1.3). The heuristic estimate H is chosen to be the R^2 distance between nodes. The figure is a snapshot of the CLOSED LIST, and the

numbers represent the coordinates of nodes explored. The number 140119.0 in the figure corresponds to (140,119) in the test map. As shown, A* algorithm switches to different "most probable low cost paths" attached to nodes in the sequence of (140,119), (160,100), (64,46), and then move along the same path via (65,47), (66,48), (67,49), (68,50). It then jumps again to the path associated with (103, 119), then yet another path attached to (101, 117). This jumping process continues and the algorithm even jumps back to very early candidate paths (highlighted in blue) in the OPEN LIST many times before moving merely one horizontal step from (140,119) to (141,119).

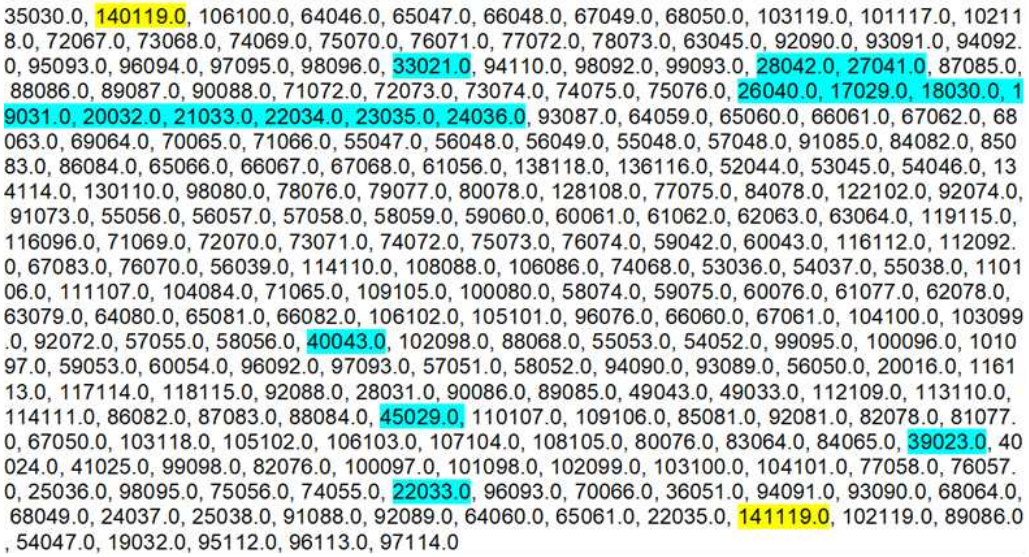


Figure 2. 3 Path switching in A* path finding algorithm

Recall the definition of cost function in this application (Eqn. 2.1), using R^2 distance between nodes as heuristic estimate H is obviously quite inaccurate when anisotropicity is high. Thus, the observation in Figure 2.3 is consistent with the complexity projection $O(b^{\epsilon d})$.

Anisotropicity can lead to a wide range of total estimated cost F at every search depth and hence making it possible for late-stage search to jump back to very early candidate path(s) in the OPEN LIST. Figure 2.4 illustrates the actual switching mechanism: when the search move on Path (i) from A_0 to A, the

current node jumps back to B on Path(ii) based on the path selection criterion of A* algorithm. The search continues for a just few steps along Path (ii) then jumps back to A on Path (i) when the current node attempts to move from C_0 to C. After one movement from A to D, however, the current node jumps to E on Path (iii). As a result, the algorithm will likely explore, and possibly close, every square on the map, producing a complete, or nearly-complete, exploration of the entire map. Due to this type of nearly-completely exploration of the map, the A* search becomes computationally inefficient.

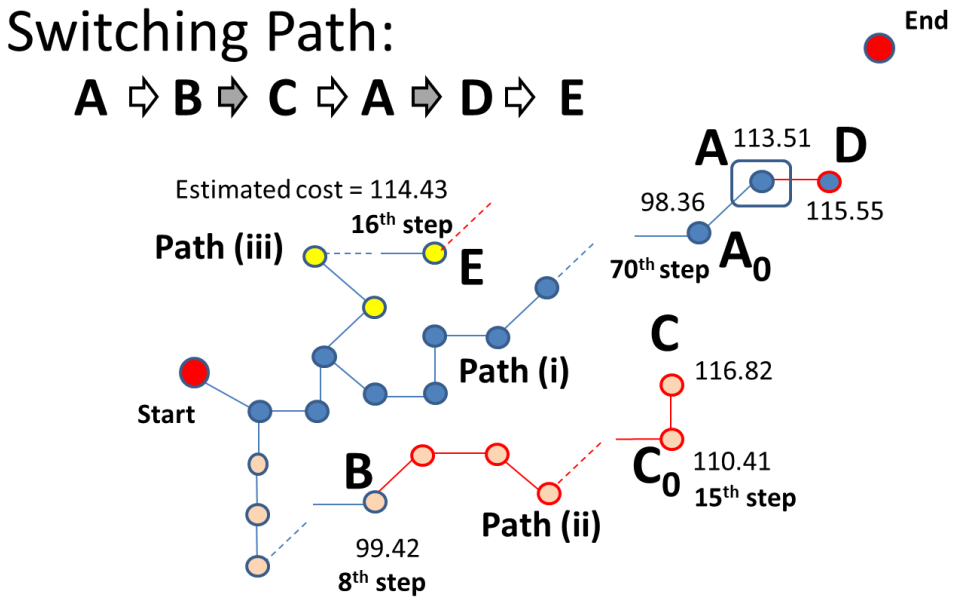


Figure 2. 4 Mechanism of path switching

Three computation examples using a typical i7 machine running Windows 10 are summarized in Figure 2.5 and Table 2.1. It should be noted that computation time is not the best indicator of algorithm complexity, as it depends on program code optimality, computer speed and loading, etc. However, the parameter should be useful for illustrating the effects of anisotropicity and different path lengths. Since $\frac{1}{2}N(N-1)$ optimal paths have to be identified for a rural electrification network with N nodes, canonical A* algorithm is obviously too inefficient.

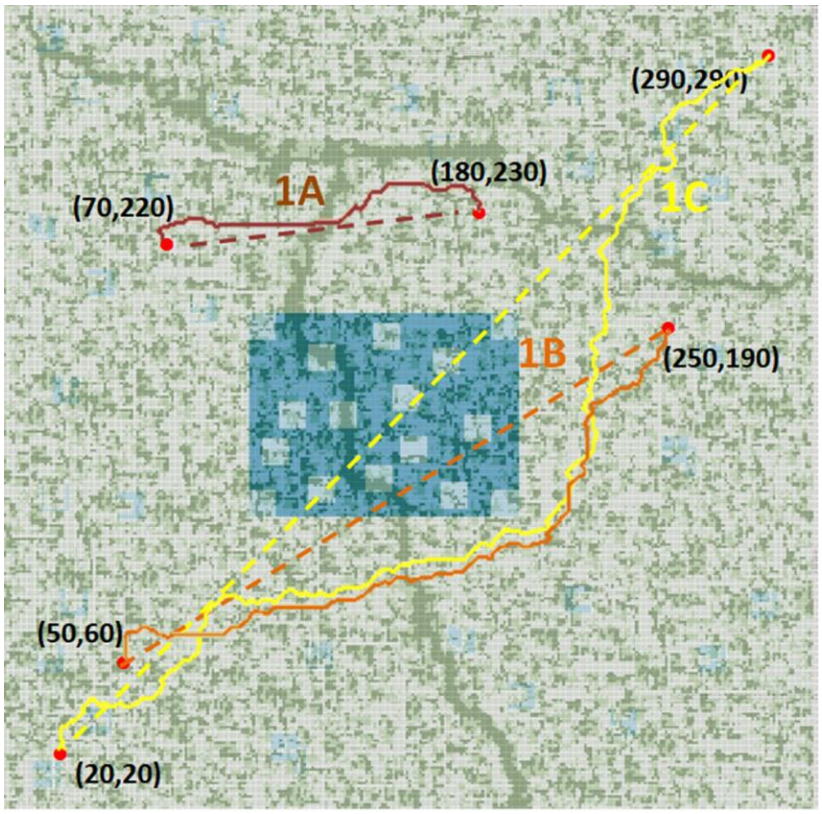


Figure 2. 5 Three Cases for Evaluation. Paths 1A & 1B optimized using A*, 1C routed using MAA*(1). See Section 2.2

Table 2. 1 Performance of A* search algorithm

Test Case	Connection	R^2 distance	Straight-line Cost	Routed Cost	Optimal Path Length, d	Size of CLOSED LIST	Computation Time
1A	(70,220) and (180,230)	110	463	197	120	9,657 (10.7% of search space)	11h
1B	(50,60) and (250,190)	239	2,620	349	262	22,495 (24.3% of search space)	92 h
1C	(20,20) and (290, 290)	382	3,397	Did not converge in time allowed.		Aborted at 50,000 (> 50% of search space)	Aborted at 296 h

Another useful and probably more indicative metric for the evaluating the performance of path finding algorithm is the size of the final CLOSED LIST. Elements in the final CLOSED LIST are fully explored nodes at the point when the optimal path has been identified. In other words, they are the nodes that have

to be evaluated before we can find the optimal path. Thus, for a given node pair, a more efficient algorithm will produce a smaller final CLOSED LIST.

It is of interest to show the effect of anisotropic variation with our test map: if we modify the search space by retaining the variation in accessibility but discarding elevation in Figure 2.5 (i.e. assuming the world is flat), computation time for Path 1A drops by 98% to 12 mins, while the CLOSED LIST size reduces by 87% to 1,231. As the distance between points increases, the impact of solution depth, d, becomes more significant according to $O(b^{\epsilon d})$. For a flat-world assumption on Path 1B, computation time reduces by 87% to 12h while the CLOSED LIST size drops by 55% to 10,213. Given that 10-100 interconnections must be computed to design the power network, the A* computational effort – which produces a globally optimum solution – is incompatible with a typical network design.

Using numerical examples to summarize and visualize the effect of anisotropicity could be helpful for develop a deeper awareness of the problem. Because of anisotropicity, A* algorithm's computation burden for two apparently similar paths can be very different due to differences in ϵ. Effect of this error is illustrated in Table 2.2. As usual, the heuristic estimate used is the R^2 distance between the nodes. Note that the average time required for making one effective move on the optimal path (T/d) grows exponentially as ϵ increases (Figure 2.6). This trend closely resembles the $O(b^{\epsilon d})$ prediction.

Table 2. 2 Effect of ϵ

Path		Average Error ϵ	Size of closed list	Optimal Path Length (d)	Optimal Path Cost	Computation time (T), sec	Time per move (T/d), sec
Short	(20,20)-(50,60)	67.9%	191	54	88	23	0.43
	(180,180)-(180,230)	80.9%	1454	53	517	1168	22.0
Medium	(140,65)-(240,30)	73.2%	1010	104	186	688	6.61
	(240,30)-(275,120)	76.9%	2016	105	175	2492	14.2
Long	(120,280)-(275,120)	78.7%	2556	247	363	4652	18.8
	(50,60)-(250,190)	82.9%	3822	270	384	10054	37.2

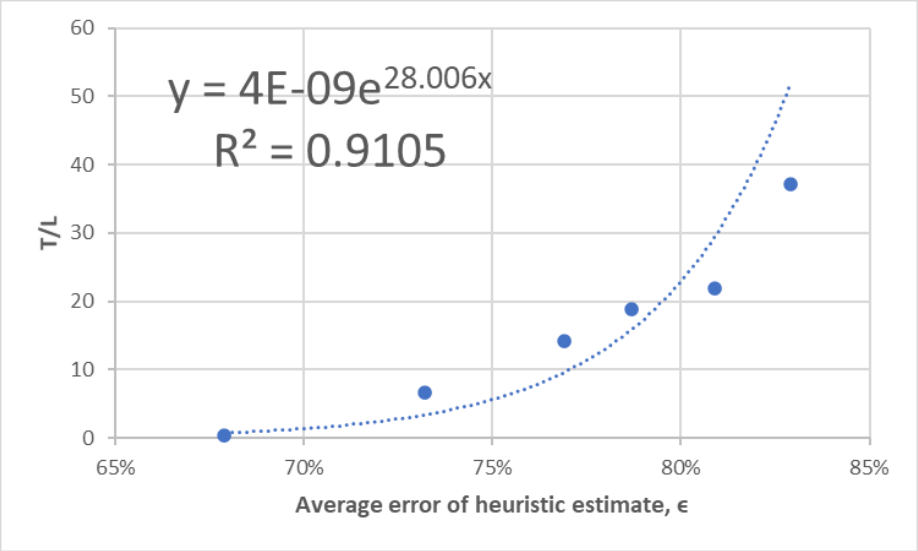


Figure 2. 6 Effect of error in heuristic estimate, ϵ

2.2.3 Common Acceleration Methods and Why They Are Not Applicable

As will be explained in Chapter 3, our proposed methods will significantly accelerate the search and reduce the size of final CLOSED LIST when compared to the standard A* algorithm. Before discussing these algorithms, however, it should be pointed out that there are many common acceleration techniques for improving the efficiency of A* path finding in large search space. For example, in many applications such as computer games, A* path finding can be very fast when the search space can be simplified using techniques such as hierarchical path-finding A* (HPA*) or a navigation mesh (NavMesh) [34] [36] [37]. However, specific conditions must be met to apply these techniques. For example, NavMesh divides a clearly defined "walkable" space into convex polygons, and replaces each polygon (or the edge of the polygon) with a single waypoint to simplify the search space; an example is shown in Figure 2.7.

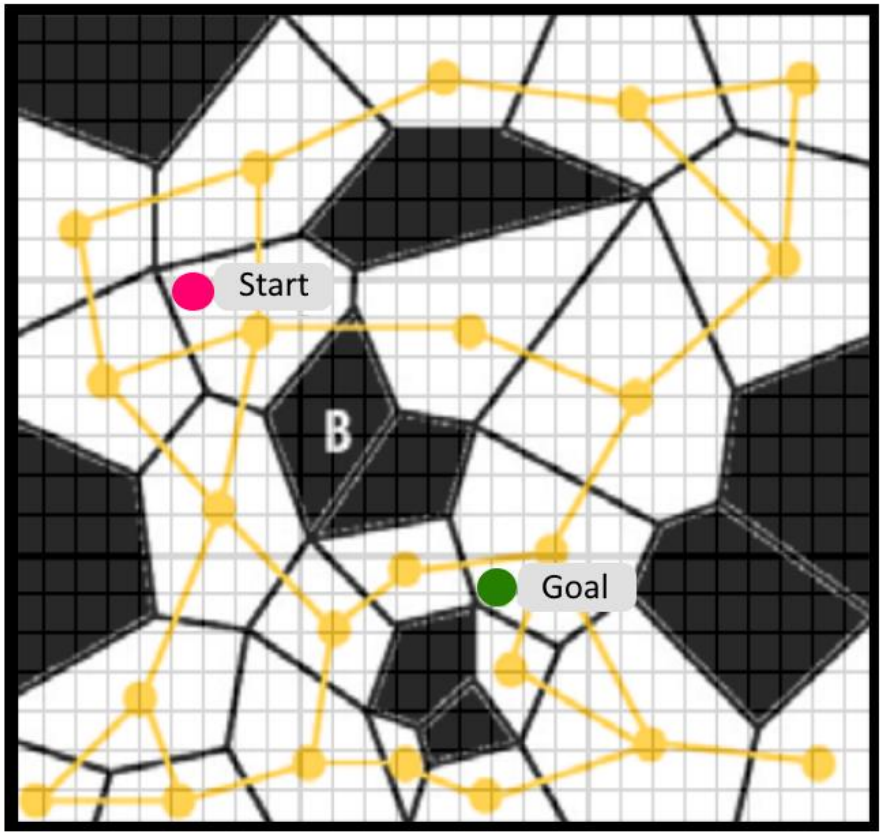


Figure 2. 7 **The NavMesh method simplifies a large search space (background grid), by dividing the space into a set of convex polygons, which common transit costs, and then replacing each with a single waypoint. (Image taken from reference [36]).**

Under the networked rural electrification framework, however, these techniques are not applicable because (i) all nodes in the space are essentially "walkable", albeit at different costs, and (ii) the search space is anisotropic due to varying elevations and accessibility in all directions. NavMesh, or similar simplification methods, reduce complexity by combining many grid squares together – i.e. by assuming isotropic areas will be substantially larger than individual grid squares. In rural electrification, this condition is not met. Indeed, the choice of grid square size is largely determined by the heterogeneity of the area, and by definition, only rarely can adjacent grid squares be combined into a single isotropic subregion.

Other researchers have attempted to reduce computational complexity by developing methods to reduce the error of heuristic estimate [50] [51] [52], typically by exploiting one or more problem-specific constraints. Although the efficiency of A* path finding may have been improved by defining a sophisticated estimate H, the efforts for computing H will inevitably increase. Thus, depending on the properties of the search space, this approach may or may not lead to meaningful improvement. As a result, most of these methods are only applicable under specific conditions, and the author is unaware of any that can be applied to highly anisotropic conditions.

2.2.4 Complex Topography Challenge for Optimal Pathfinding: Summary

To summarize, A* path finding is inefficient for rural electrification network design because complex topography of rural environment makes the search space highly anisotropic. Under this situation, it is difficult to define a computationally-simple heuristic estimate H that maintains a small error for the entire search space while never over-estimating costs. Computation complexity increases exponentially with the heuristic estimate error ϵ according to $O(b^{\epsilon d})$. Common acceleration techniques are not applicable to this situation because those methods are effective only for search spaces that are either less anisotropic or exclude the majority of possible branches (e.g. 'stay on roads' or 'do not drive through walls'). Alternatively, computational efficiency could possibly be improved by defining a more accurate heuristic. However, this approach is dependent on specific properties of the search space and is not generally attainable. In addition, the increasing complexity for computing such a heuristic may offset the advantage of improving search efficiency.

2.3 Effect of Anisotropicity on Other Optimization Methods

Besides A* algorithm, this research has also developed methods based on classical genetic algorithm and fuzzy logic. They are, namely, levelized interpolative genetic algorithm (Chapter 5) and modified fuzzy

inference (Chapter 6). Both methods are hierarchical in nature, i.e. a final solution is obtained by comparing the refined version of several coarse solutions.

Anisotropicity of the search space also has significant impacts on hierarchical approach. For a fairly isotropic space, correspondence between coarse solution and its refinement is usually strong. That is, if coarse solution A has a lower cost than coarse solution B, the refined cost of solution A will likely (though not absolutely) also have lower than that of solution B. With increasing anisotropicity, however, such correspondence will be weakened. For example, if simple R^2 distance between two points is chosen as a coarse representation of optimal path cost, correspondence between real optimal cost and this coarse cost may be reasonably high for a fairly isotropic space, but will be very low for our application. In Chapters 5 and 6, we will further illustrate this limitation and so develop appropriate measures to properly identify optimal solutions.

CHAPTER 3 – NEAR-OPTIMAL ROUTING USING MULTIPLIER-ACCELERATED A* ALGORITHM [1]

3.1 Introduction

The 7th of United Nations' Sustainable Development Goals (SDG7) [1] aims to *"ensure access to affordable, reliable, sustainable and modern energy for all"* by 2030. Globally, the population without electricity access fell from 1.2 billion in 2010 to 840 million in 2017, which was a remarkable achievement [2] [3]. In the last few years, however, the progress has become considerably slower. According to the latest SDG7 tracking report, 759 million people are still being left without electricity supply [4]. Based on the current practices, International Energy Agency (IEA) estimates that 660 million people, or nearly 10% of the global population, will remain without electricity access by 2030 [5]. In addition, there is still a substantial rural-urban divide in electrification: 84% of the unserved population (640 million people) live in rural areas and 20% of the global rural population has limited-to-no access to electricity. Due to the distances and the geographical complexity of these isolated locations, further electrification of remaining remote areas will require substantially increased marginal investments. Better approaches to resolve issues in capacity planning, affordability, and reliability are needed [5].

Research has illustrated a clear linkage between availability of electricity and rural development [6] [7] [8] [9] [10]. While richer households may be more ready to pay for electricity and may see more immediate benefits, poorer households have also shown some willingness to prioritize electrification expenditures. However, loads in newly electrified areas remain difficult to predict, complicating system planning.

[1] This chapter was published in *Energy and AI* as "Effective rural electrification via optimal network: Optimal path-finding in highly anisotropic search space using Multiplier-accelerated A* Algorithm" [30].

Due to cost, physical distance and challenging topography, it is typically uneconomic to connect the remaining unelectrified areas to the main grid. As a result, many international organizations and policy makers believe renewable energy, especially solar photovoltaic, is a more economically attractive solution for rural electrification [1] [3]. Rural electrification is currently structured into a multi-tier framework proposed [6]. Tier 1 (min. 3W and 12Wh/day for a household) and tier 2 (min. 50W and 200Wh/day for a household) can be addressed with "solar home systems (SHS)" – typically a small PV panel and battery installed on an individual home [6] [11]. Tiers 3 & 4 require more electric supply similar to grid-attached systems (AC systems at 120V and above, with multi-amp connections). These can only be implemented by grid-connections or by minigrids providing similar power levels. Additionally, these systems are required for higher consumption activities, such as motor loads or small business activities [6] [12]. Research has also shown that there are opportunities for significant improvements in economics, reliability, and investment risk management [13] [14] [15].

In a rural context, villages locations were historically selected to provide ready access to water, soil, food storage, defense, and similar concerns. These locations are often sub-optimal locations for solar or wind power generation. Developing these energy sources within the village may therefore require higher capital investment and land usage than would be required if generation were placed in nearby locations with better renewable energy conditions. Furthermore, the acquisition of land for energy generation near agricultural villages may raise counterproductive tensions with landowners.

Load forecasting is also a major issue for rural electrification: Studies often indicate difficulty estimating loads during planning, and early load growth after electrification often falls short of expectations or occurs in location other than those forecast [12] [16] [17] [18] [19]. Generalized population or load statistics are not reliable for villages due to small geographical coverage and customer count. Longer term demand depends heavily on population change and economic growth, and is highly uncertain. Future urbanization plan of the neighborhood areas, changing ecological conservation policies, and many other non-technological factors can also significantly affect the outlook of rural electrification [20] [21]. In

particular, villages may grow differentially, with growth occurring in some villages, outstripping installed power generation, while growth may be retarded in other villages, idling power generation. This increases the need to move power between villages.

In general, the solution to many of these issues is to implement in minigrid context the generation siting load balancing implemented in national-scale grids: Interconnecting multiple load centers – i.e. villages or businesses – with multiple, ideally sited, generation locations. Minigrids in remote rural areas are small but connections can be extremely complicated due to intricate topographies, driving a need to automate the planning of interconnections and generation. This study addresses the interconnection planning, by proposing a flexible and cost-effective method to route interconnecting power lines between villages. Unlike conventional minigrids for tier 3 and above [6], this approach connects villages (some of which will have co-located generation), with each other and with centralized generation. The approach emphasizes low voltage interconnects and near-optimal routing in areas with dispersed loads and substantial topographical variation, shown schematically in Figure 3.1. Appendix 2A provides a more specific example, based upon part of Mentougou district of Beijing, China. The authors have worked in similar situations in sub-Saharan Africa (esp. Rwanda) with similar topographical challenges.

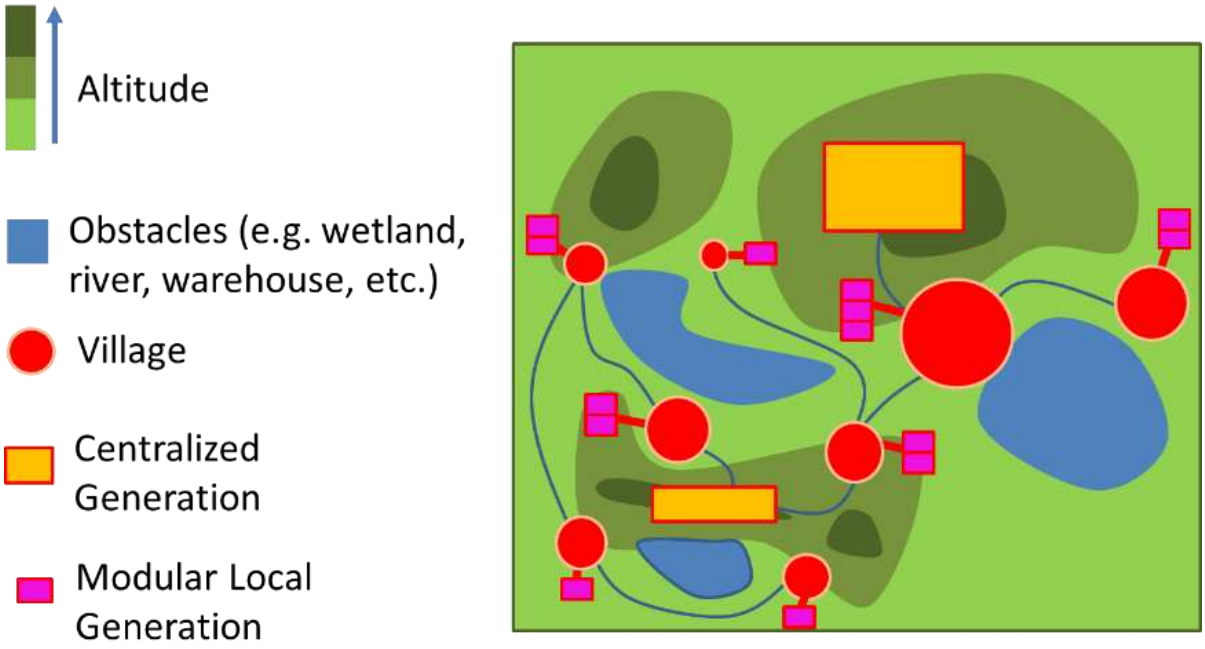


Figure 3. 1 **Schematic view of the proposed networked electrification method. Villages may contain local generation resources, complimented by centralized generation placed in areas with better resource potential and fewer land conflicts. Unelectrified areas in the developing world are most often remote, with challenging topographical constraints, as indicated the elevation and obstacle shading in the figure.**

3.1.1 Formulation of the Networked Model

The networked electrification model can be posed as a minimizing the cost of capacity (capital costs) constrained by (a) required interconnection of multiple load and generation locations, and (b) avoiding keep-out areas where power cables cannot be installed. To pose the problem, the first step discretized the region of interest into a set of 2D grid squares, translating an analog, continuous problem space into a discrete, semi-digital computational problem. Each grid square becomes a coordinate pair assigned an elevation and an ease-of-access metric (Figure 3.2). All examples in this paper will use a 10m x 10m grid square covering a 3km x 3km map area. Both grid size and total map size can be scaled for the region under consideration; key to this discussion is that a map of *at least* 300 x 300 cells is typically required in practical applications. The map utilized here is indicative of the Chinese scenario discussed in the appendix, and is

sufficient to plan low voltage power distribution using poles and cables (see Figure 3A.1, Appendix 2A); the grid spacing aligns with the pole spacing for low voltage connections: The ≈15 m maximum span between poles is approximately the diagonal distance of a 10m x10m grid.

The map structure proposed here can be created from digital terrain model (DTM) using data from satellites or unmanned aerial vehicle (UAV) surveys [41]. DTM is a mature today technology and numerous applications are found in many different sectors [42] [43] [44]. The technology is capable of identifying objects and elevations with minimal human intervention [53] [54] [55]. Furthermore, since this work proposes to establish centralized solar generation facilities, DTM is useful for selecting location with strong solar radiation when reliable statistics for small areas are not readily available [45] [46] [47] .

The DTM is then translated into accessibility and elevation metrics. For routing, accessibility is represented as a continuous variable assigned to each individual grid square. In addition to the DTM, accessibility uses data from ground surveys and consultation with the local population. While elevation data is essentially continuous (typically discretized into 1-5 m steps), we have found that elevation usually does not need to be represented by a continuous variable. It is computationally simpler to represent elevation by a few discrete levels (this work uses 3). Elevation levels can be increased as needed; the computational advantage of the proposed algorithm become even more apparent as number of levels increases.

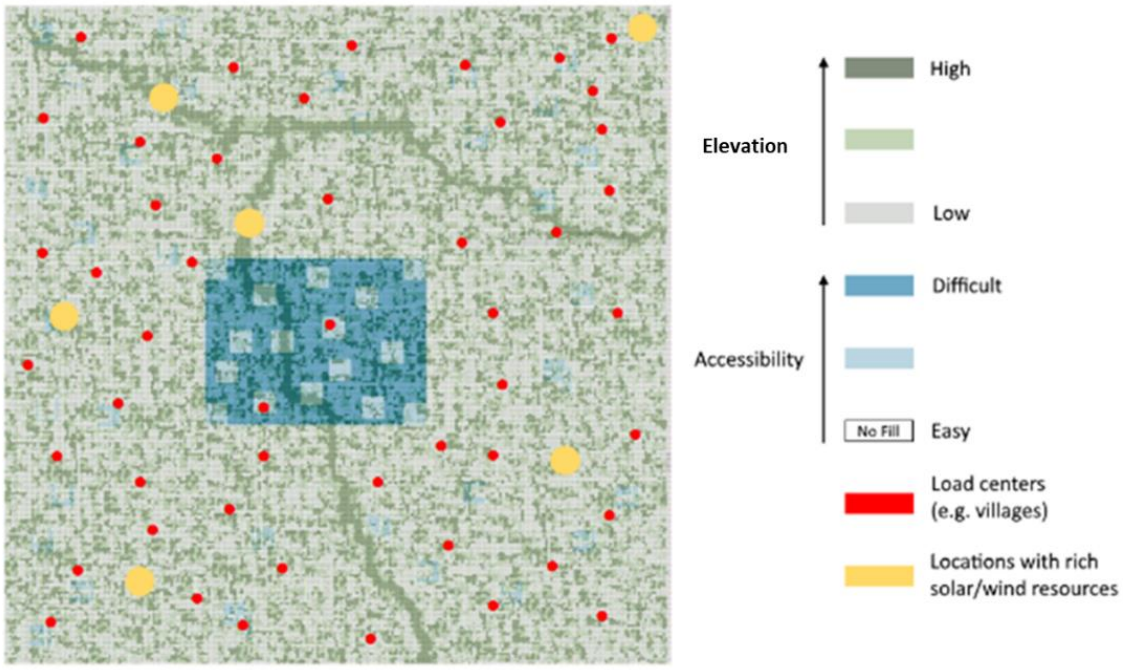


Figure 3. 2 **Discretized map indicating both elevation (greens) and accessibility (blues) metrics. Load centers are indicated coincident with villages in this example, although this is not always the case. Ideal locations for renewable resources are also marked. Size of village and generation locations are shown larger than the operational map for illustration; in practice, each point is assigned to one grid square.**

As with national-scale grids, we assume power can be transmitted between villages and from generation locations – an electrical sizing problem that must be considered in the later stages of planning. Therefore, individual villages can be evaluated to identify suitable generation locations – proper land and renewable resources – and compared to nearby locations that may have better renewable energy resources. Figure 3.3 overlays the resources an interconnection on a portion of the map shown in Figure 3.2.

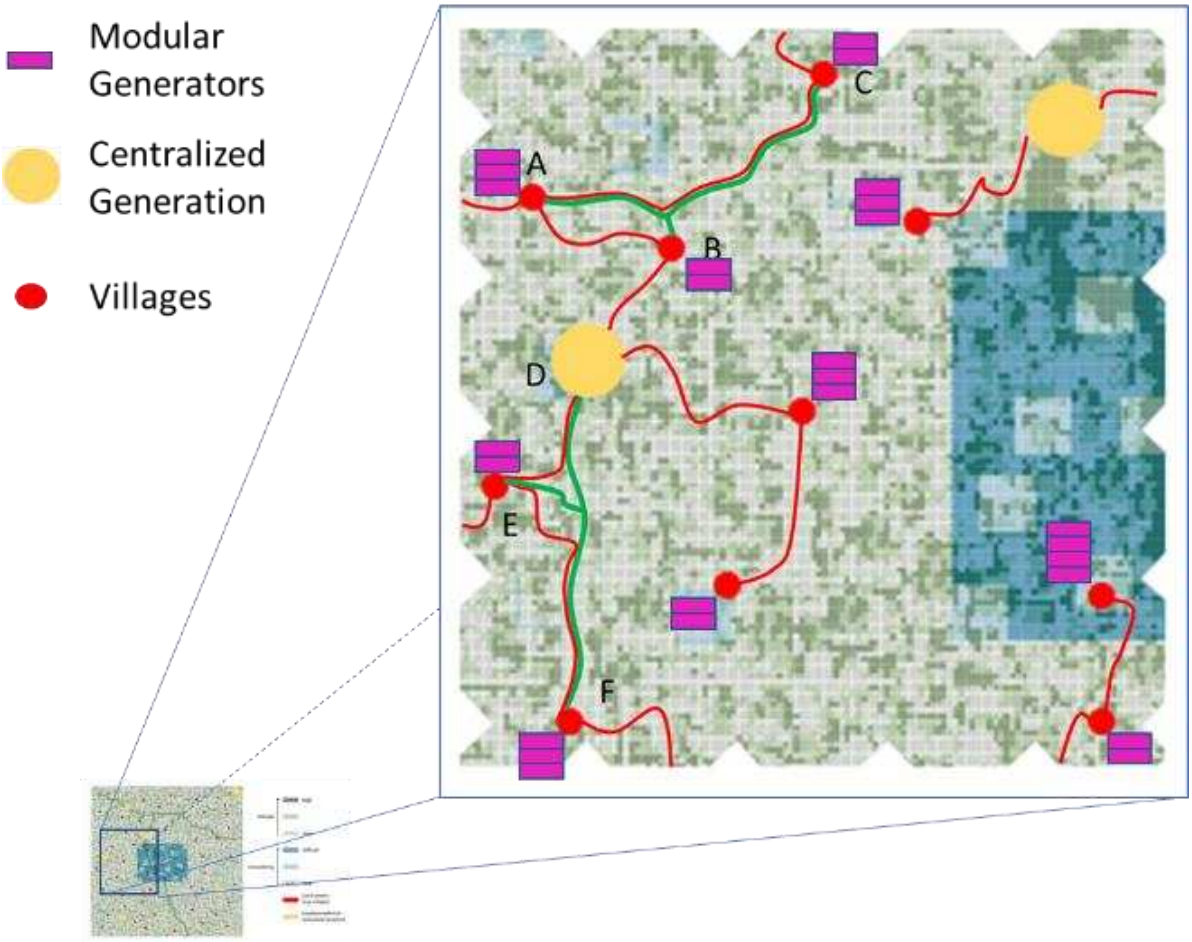


Figure 3. 3 **An example of power supply interconnections, using a subset of the study map in Figure 3.2. Rectangular icons indicate the land use and resource potential for local generation near each population center. Red lines reflect notional routes for power connections between villages (red) and centralized generation (yellow).**

This work focusses on a subset of the complete design of the minigrid: Optimized and computationally affordable routing of the power connections (red lines) in Figure 3.3. Specifically: For any two points of interest, what is the optimal, or near-optimal, route between the points? Given near-optimal routes between any two locations in Figure 3.2, numerous methods are available to synthesize the minimum cost network [27]. With N points of interest there are $\frac{1}{2}N(N-1)$ possible interconnections that must be calculated, although this number can possibly be reduced by some trimming of pairs. Therefore, the pair-wise route planning algorithm must be both efficient and require little-to-no human supervision.

The optimal connection path between two points must account for changes in elevation and accessibility along the connection. For the study problem, the cost of the $(i+1)^{th}$ movement in the path is:

$$C_{i+1} = [(z_{i+1} - z_i)^2 + (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{\frac{1}{2}} + w_1|z_{i+1} - z_i| + w_2 a_{i+1} \qquad (3.1)$$

where $x_i, y_i, z_i,$ are the 2D coordinates and elevation of the i^{th}point in the path, a_i is the accessibility of the i^{th} point on Figure 3.2, w_1, w_2 are weighting coefficients.

The first term (the square root) in Eqn. 3.1 is the length from step i to step $i+1$ and is directly proportional to cost – i.e. the length of the power cable required to span the distance. The second term captures additional costs driven by a change in elevation (additional pole stays, increased transportation costs, labor, etc.), and the third term captures the cost driven by the accessibility of the destination grid square (additional engineering requirements, infrastructure relocation costs, landowner compensation, road crossings, etc.). Of these, accessibility requires the most judgement. For example, sociopolitical factors are exceptionally difficult to capture and represent in an 'accessibility' metric, yet may have a substantial impact on the route: Will landowners welcome or protest electrical systems on their land? Are there areas where electrical systems are would impact cultural or historical properties or impact land values? Ultimately, the project developer needs to evaluate these factors to estimate the cost and advantages of routing through any specific area.

Finally, the weighting coefficients (w_1, w_2) are selected to weight elevation change and accessibility constraints to be similar in magnitude as the distance term.

Including all factors above, route planning must meet three objectives: (1) nearly cost-optimal, (2) account for political and sociological factors, and (3) run effectively on standard computer hardware with minimal user intervention. The second objective is often ignored when constructing algorithmic solutions, yet is critically important in practical applications. Frequently, at late stages of planning – or even during implementation – previously unknown sociopolitical factors will come to light, making planned routing

difficult or impossible. Therefore, a family of near-optimal routes is often superior to a single, optimal solution, as it allows for late-stage change in routes after engagement with local communities. However, efficiently generating family of routes is another research problem and will be discussed in another publication.

It is important to note that the accessibility metric may be the result of more or less extensive pre-analysis of the on-ground situation where the network is proposed. The detail in this analysis may range from superficial (e.g. examining satellite or aerial photos) to complex, including on-ground surveys, review of local maps, and discussion with landowners or government officials. After this type of analysis, the algorithm requires the analyst to assign an accessibility cost to each cell, and to select a weighting factor to scale these appropriately with other costs. This inherently requires engineering, political and economic judgements prior to engaging the algorithm for routing; most project developers cultivate, and often compete on, understanding these issues.

Although using an optimal network to connect villages together for better energy planning and utilization does not appear to be a complex concept, obtaining this network is not as straightforward as it first appears. In most planning systems [56] [57], it is assumed that limited number of paths for connecting two users already exist or can be determined intuitively by examining GIS information. Therefore, the search space for optimizing the connection network is small, and a classical approach such as A* can be easily applied. Indeed, A* algorithm is a well-recognized pathfinding method that is provably optimal provided the initial heuristic always underestimates the cost of the path [28].

However, the assumption of limited number of connection paths and the appropriateness of intuitive path determination should be carefully examined. Unlike road routing, power lines do not need to follow existing paths or other physical features. This opens a plethora of possible paths, and, as demonstrated in Section 2.2, small variations in path shape can lead to significant change in cost. As a result, selecting intuitive paths – which tends to follow existing features – may lead to unsuitably sub-optimal results.

Selecting near-optimal, rather than intuitive, paths require exploring more of the possible routing space, resulting in a much larger search space. The problem for such exploration, as will be shown in Section 2, is that the classic A* algorithm will become computationally intractable when applied to the continuously varying cost variables required for this problem. To resolve this issue, we have developed a multiplier-accelerated A*, "MAA*", algorithm that significantly improves the speed of standard A* algorithm at a cost of some optimality; see discussion in [29]. With this AI-based algorithm, rural energy network designer can trade-off between computation speed and optimality for different design problems or at different design stages. This novelty is important for quick but reasonably accurate site and configuration evaluations, which is not possible with classical methods.

The remainder of the paper is organized as follows: Section 2 describes why canonical application of A* to the rural routing problem is intractable, discusses the MAA* algorithm, and examines computation examples. Section 3 summaries simulation results of MAA* algorithm and, based on these results, applies minimum spanning tree techniques to synthesize example networks. Section 4 discusses limitations of MAA*. Conclusive comments are provided in Section 5.

3.2 Inefficiency of A* algorithm in a highly-anisotropic search space and the development of multiplier-accelerated A* algorithm

A* algorithm is proven and widely used for optimal pathfinding, and commonly utilized in applications such as highway route planning. Although A*can be directly applied to the discretized map (Figure 3.2), performance suffers because common acceleration techniques are not applicable to our problem. For example, in many applications such as computer games, A* path-finding is fast, even for large maps, when the search space can be simplified using techniques such as hierarchical path-finding A* (HPA*) or a navigation mesh (NavMesh) [34] [36] [37]. However, specific conditions must be met to apply these techniques. For example, NavMesh divides a clearly defined "walkable" space into convex polygons, and replaces each polygon (or the edge of the polygon) with a single waypoint to simplify the search space; an example is shown in Figure 3.4.

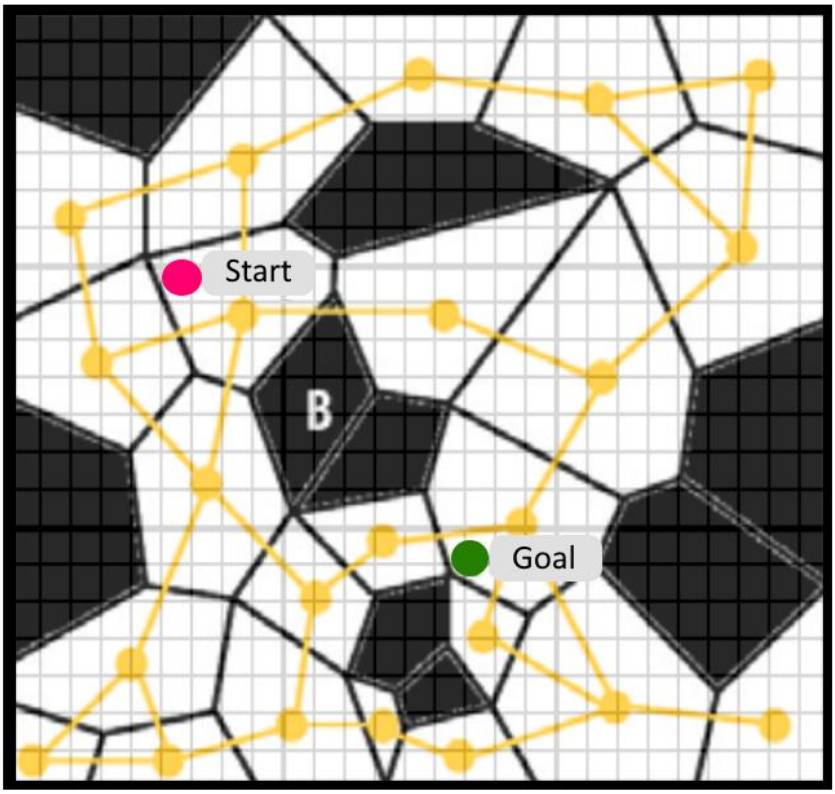


Figure 3. 4 **The NavMesh method simplifies a large search space (background grid), by dividing the space into a set of convex polygons, which common transit costs, and then replacing each with a single waypoint. (Image taken from reference [36]).**

Under the networked rural electrification framework, however, these techniques are not applicable because (i) all points in the space are essentially "walkable", albeit at different costs, and (ii) the search space is anisotropic due to varying elevations and accessibility in all directions. NavMesh, or similar simplification methods, reduce complexity by combining many grid squares together – i.e. by assuming isotropic areas will be substantially larger than individual grid squares. In rural electrification, this condition is not met. Indeed, the choice of grid square size is largely determined by the heterogeneity of the area, and by definition, only rarely do adjacent grid squares be combined into a single isotropic subregion.

Other researchers have attempted to reduce computational complexity by developing methods to reduce the error of heuristic estimate [51] [50] [52], typically by exploiting one or more problem-specific constraints. As a result, most of these methods are only applicable under specific conditions, and the authors are unaware of any that can be generally applied to highly anisotropic conditions.

To address these issues, the multiplier-accelerated A* (MAA*) algorithm works directly from the discretized map, without the need to reduce the map complexity. In Li, et al. [29] we numerically demonstrated the efficiency improvement of MAA* against standard A* path-finding method; for the rural electrification problem, MAA* reduces computation time by ~90% while sacrificing optimality by 10% or less. Section 2.1 briefly reviews the mechanism that drives inefficiency using canonical A* in this particular application, and Section 2.2 reviews MAA*.

3.2.1 Efficiency problem of A* algorithm

For subsequent discussion, it is important to understand the underlying computational challenge with the A* algorithm, applied to this problem. Since search space for this problem is large (e.g. 90,000 points for Figure 3.2) and not practically reducible into simpler geometry due to topological variation and change in accessibility across the whole map (i.e. highly anisotropic), common A* acceleration techniques cannot work efficiently. The computational complexity of A* depends on the error of heuristic estimate, which in turn affects the average branching from each node along the solution path. Precise quantification of A* time complexity in a anisotropic search space is difficult. However, a first-order approach is possible: For large space with constant step cost, A* time complexity is described by $O(b^{\epsilon d})$ [38], where b is the branching factor – an average number of downstream explorations of each point on the path; $\epsilon = (h^* - h)/h^*$ is the error in heuristic estimate, where h^* the actual cost, and h is the cost estimated by the heuristic, both from the node to final goal; and d is the solution depth – path length in our case. Even with constant step

cost, the complexity of search increases exponentially with d when ϵ is finite (Figure 3.5, for illustration, average b is assumed to be 2.13).

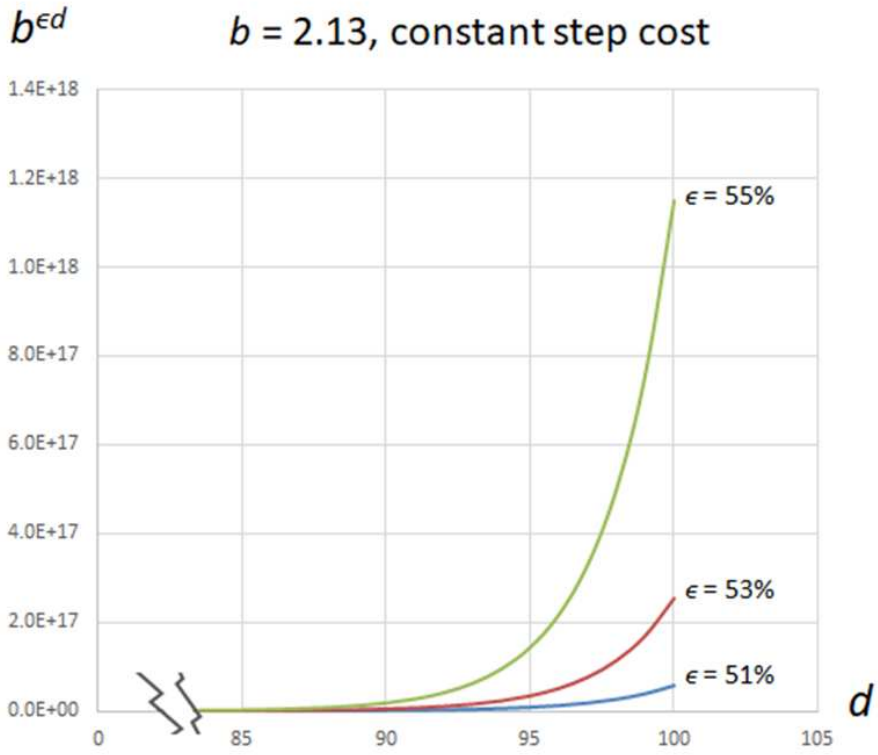


Figure 3. 5 **Time complexity vs search depth for the branching factor in the discretized map (b = 2.13) and assuming a constant step cost for illustrative purposes. A search depth of 50-250 is typical of many pairwise routings for rural electrification problems. In practice, this is exceptionally challenging – nearly impossible – given the anisotropic map characteristics.**

For A* algorithm, the heuristic function h must always under-estimate actual cost in order to ensure optimality. In this case, the heuristic error is always positive. It is common to use straight-line distance as h so optimality is guaranteed. Practically, it is difficult to define a heuristic function with small error over this highly anisotropic search space, while still meeting the A* requirement that the heuristic always underestimate actual cost. If a chosen h may sometimes over-estimate the actual cost, absolute value of the heuristic error should be used to estimate computational complexity.

Computational complexity described by Figure 3.5 assumes constant step cost. In our application, step cost can vary significantly. While researchers [48] [49] have developed more sophisticated methods to analyze the effect of ϵ on A* search, one would intuitively expect time complexity of our problem will grow even more rapidly than the constant step cost situation.

As practical examples, Figure 3.6 and Table 3.1 shows the results of connecting three point pairs using the standard A* search algorithm. Computations are performed on an i5 machine, running Microsoft™ Windows 10™ with standard settings. The program is written in Python (Version 3.6). While absolute runtimes depend on program code optimality and computer speed, times are comparable between these test cases. For this analysis the discretized grid is 300 grids square (i.e. $i \in [1 \ldots 300]$), unit distance is assumed between rows and columns of grid squares (i.e. $x_{i+1} - x_i = 1, y_{i+1} - y_i = 1$), elevation is discretized to three levels ($z_i \in \{1,2,3\}$), and accessibility is discretized to three levels ($a_i \in \{0,3,10\}$). Weights are set as $w_1 = 6$ and $w_2 = 3$. The first weighting factor scales the cost penalty caused by unit change in elevation to path length. It depends on labor cost, equipment availability, skill level, etc., and therefore varies from region to region. However, it will usually be a few times of the corresponding length cost in practice. The second weighting factor w_2, scales accessibility to path length. The factor is chosen so that, at the first level (i.e. $a_i = 3$), magnitude of the penalty is comparable to a modest change in elevation. This corresponds to manageable common situations such as movable obstacles, slightly unfavorable soil condition, etc. The second level (i.e. $a_i = 10$) scales accessibility such that the path-finding algorithm will avoid this node unless it is absolutely necessary to go through it. Examples are wetland, area that requires special permit, costly right-of-way, etc. See Appendix 2B for additional illustrations of A* performance.

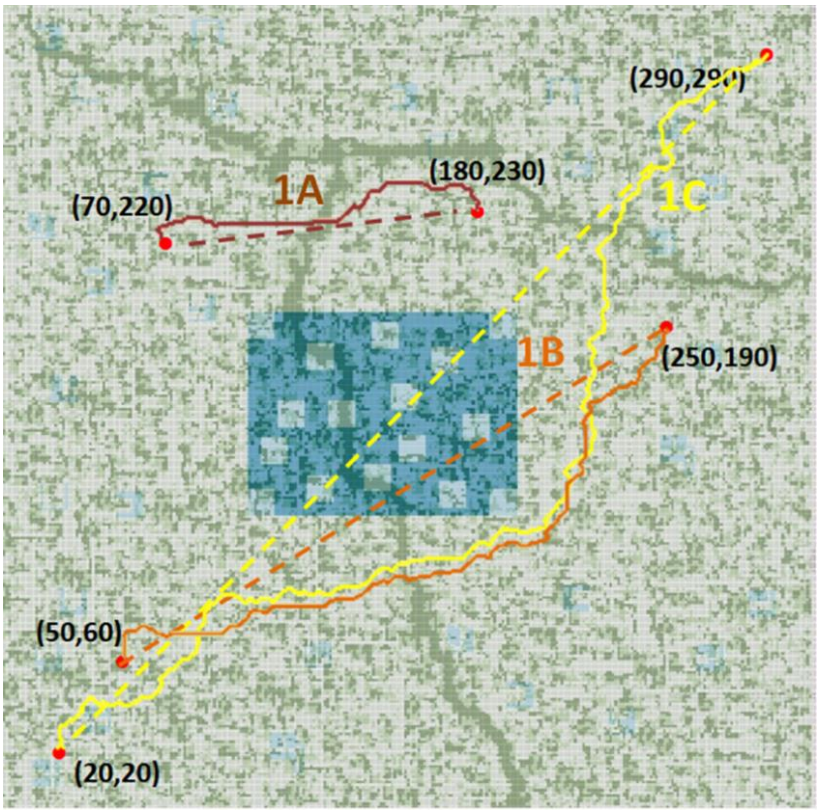


Figure 3. 6 **Three node pairs used to evaluate performance. Paths 1A & 1B were routed using canonical A*. Path 1C was routed using MAA*(1). Elevation and accessibility are the same as Figure 3.2.**

Table 3. 1 **Performance of A* search algorithm for the test cases in Figure 3.6**

Test Case	Connection	R^2 distance	Straight-line Cost	Routed Cost	Optimal Path Length, d	Size of CLOSED LIST	Computatiol Time
1A	(70,220) and (180,230)	110	463	197	120	9,657 (10.7% of search space)	11h
1B	(50,60) and (250,190)	239	2,620	349	262	22,495 (24.3% of search space)	92 h
1C	(20,20) and (290, 290)	382	3,397	Did not converge in time allowed.		Aborted at 50,000 (> 50% of search space)	Aborted at 296 h

* Closed list size is the count of all nodes visited by the algorithm; 90,000 nodes exist in the entire map.

Consistent with the $O(b^{\epsilon d})$ complexity, the solution depth, d (i.e. the optimal path length), drives an exponential increase in computational time. Additionally, the anisotropic variation increases ϵ, amplifying the exponential time complexity. Anisotropicity forces the search to visit large portions of the map as d

increase, as illustrated by the size of the closed list (measure of memory complexity). These results are consistent with other applications of A* to large complex search spaces [38].

To illustrate the impact of anisotropic variation, retaining the variation in accessibility but discarding elevation in Figure 3.6 (i.e. assuming the world is flat), computation time for path 1A drops by 98% to 12 mins, while the closed list size reduces by 87% to 1,231. As the distance between points increases, the impact of solution depth, d, becomes more significant. For a flat-world assumption on path 1B, computation time reduces by 87% to 12h while the closed list size drops by 55% to 10,213. Given that 10-100 interconnections must be computed to design the power network, the A* computational effort – which produces a globally optimum solution – is incompatible with a typical network design.

3.2.2 Multiplier-accelerated A* algorithm

Qualitatively, the underlying issue with the canonical A* algorithm in this example is that A* will continuously trace back and revisit very early candidates in the open list to search for a more optimal path, even when near the destination node. The algorithm therefore devolves to a near-complete search of the map between the start and target nodes. In practice, however, there is a low probability that revisiting an early node will substantively change path cost, as the new path is also subject to the same large heuristic error between start and target nodes. The multiplier-accelerated A* (MAA*) algorithm exploits this low probability to reduce inefficient revisits of early candidate nodes. The principle is illustrated in Figure 3.7 and a small-scale demonstration was provided in [29]. A more realistic, full-scale application is examined in this work.

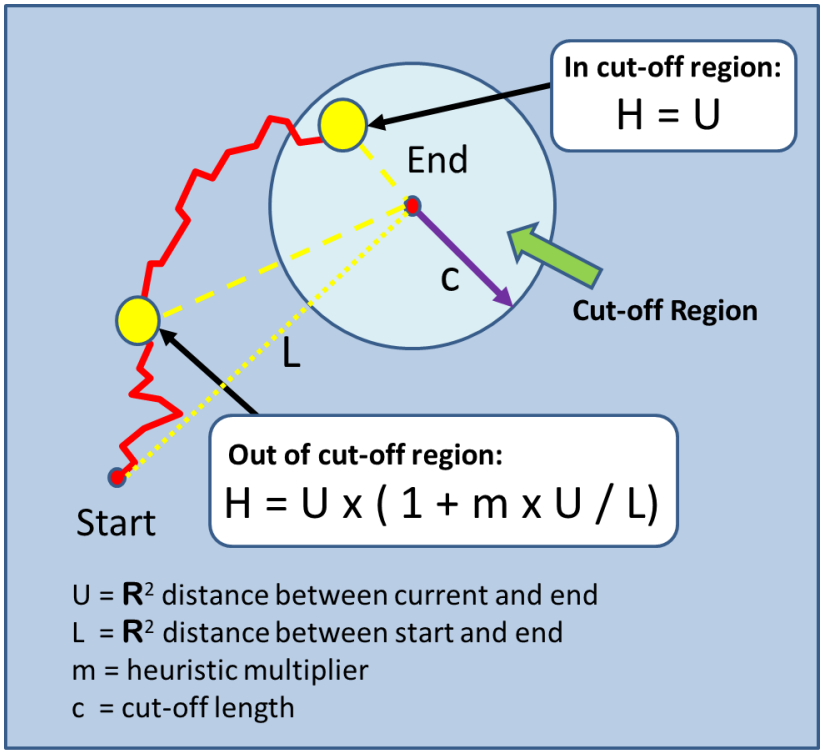


Figure 3. 7 Multiplier-accelerated A* Algorithm (source: [29])

The standard A* algorithm for this type of problem typically utilizes a Euclidean or Manhattan distance as the heuristic estimator. When the map is highly anisotropic, similar estimated total costs exist at virtually all search depths. Therefore, the A* algorithm will frequently revisit open paths uncovered early in the search to ensure optimality. To address this problem, MAA* incorporates this consideration by inflating the traditional heuristic U (i.e. the Euclidian distance between the current search position and the end point) by $\frac{m \times U}{L}$, where m is a chosen multiplier, and L is the Euclidean distance between the start and end points (see Figure 3.7). At the early stage of the search, U and L are close and $H \approx U \times (1 + m)$. Inflation is gradually reduced as the current point moves closer to the end point and U is now small compared to L. When the current point enters into the circular c, `centered on the ending point with radius c, the algorithm reverts to the traditional heuristic. Frequency of revisiting prior path decisions (and increasing the closed list size) is reduced by increasing the estimated cost of those decisions. However, since the method cannot

guarantee that the heuristic cost is always less than actual cost, the path identified by MAA* may be suboptimal [39].

To demonstrate the performance of MAA*, we consider the connection between (50,60) and (250,190) in Figure 3.6. Results are summarized in Tables 3.2 and 3.3.

Table 3. 2 Computation Results using MAA* for path 1B in Figure 3.2

Cut-off (c)	Multiplier (m)	Time	Closed List	Path Length	Cost
10	0.5	19.5 h	9053	262	364
	1	3.9 h	4193	268	395
	2.5	0.3 h	1087	253	533
	5	0.04 h	557	245	654
50	0.5	23.3 h	8799	273	363
	1	7.9 h	5081	282	407
	5	0.7 h	1650	239	660
100	0.5	29.8h	11101	273	371
	1	7.3h	5635	271	408
	5	1.5h	2978	244	626
Global Min	0	92.3 h	22495	262	349

Table 3. 3 Comparison among different m values @ c = 10, connection (50,60)-(250,190)

	Connection: (50,60) - (250,190) @ c = 10 Direct connection cost = 2620.03				
	Computed Optimal Cost	Computation Time (sec)	Normalized Computation Time	% Over-estimate w.r.t. optimal	% improvement w.r.t. direct connection
Standard A*	349	332201	100.0%	0.0%	86.7%
MAA*					
m = 0.5	364	70223	21.1%	4.5%	86.1%
m = 1	395	14186	4.3%	13.3%	84.9%
m = 2.5	533	1085	0.3%	52.7%	79.7%
m = 5	654	144	0.0%	87.7%	75.0%

The results are consistent with the observations in [29] where a much smaller and less realistic 30 x 30 map is used. Parameters m and c impact the algorithm's performance:

1. *Multiplier*: The multiplier, m, amplifies the heuristic of early searches in the open list to reduce frequent revisits. Thus, the frequency of switching back to early paths will reduce when m is large. This decreases computation time, but will also ignore potential candidates for an optimal path. This is reflected in Tables 3.2 and 3.3 – a large m substantially reduces computation time while substantially increasing cost (suboptimal routing), while small is small increases computation time and reduces cost. Note $m = 0$ reduces MAA* to canonical A*.
2. *Cut-off region:* The search reverts to canonical A* when it enters the cut-off region. In other words, true optimal search is conducted inside the cutoff region. Keeping a larger cut-off region will significantly increase computation burden and there is little practical advantage of doing so. The only reason to keep a small cut-off region is to avoid truncation error in computation, an issue only when m is very large. Thus, c can even be chosen as zero in normal cases.

For convenience, the following notation will be used: MAA*(p, q) would mean MAA* algorithm with $m = p$ and $c = q$. If q is omitted, $c = 0$.

Figure 3.8 shows the physical shapes of the obtained paths.

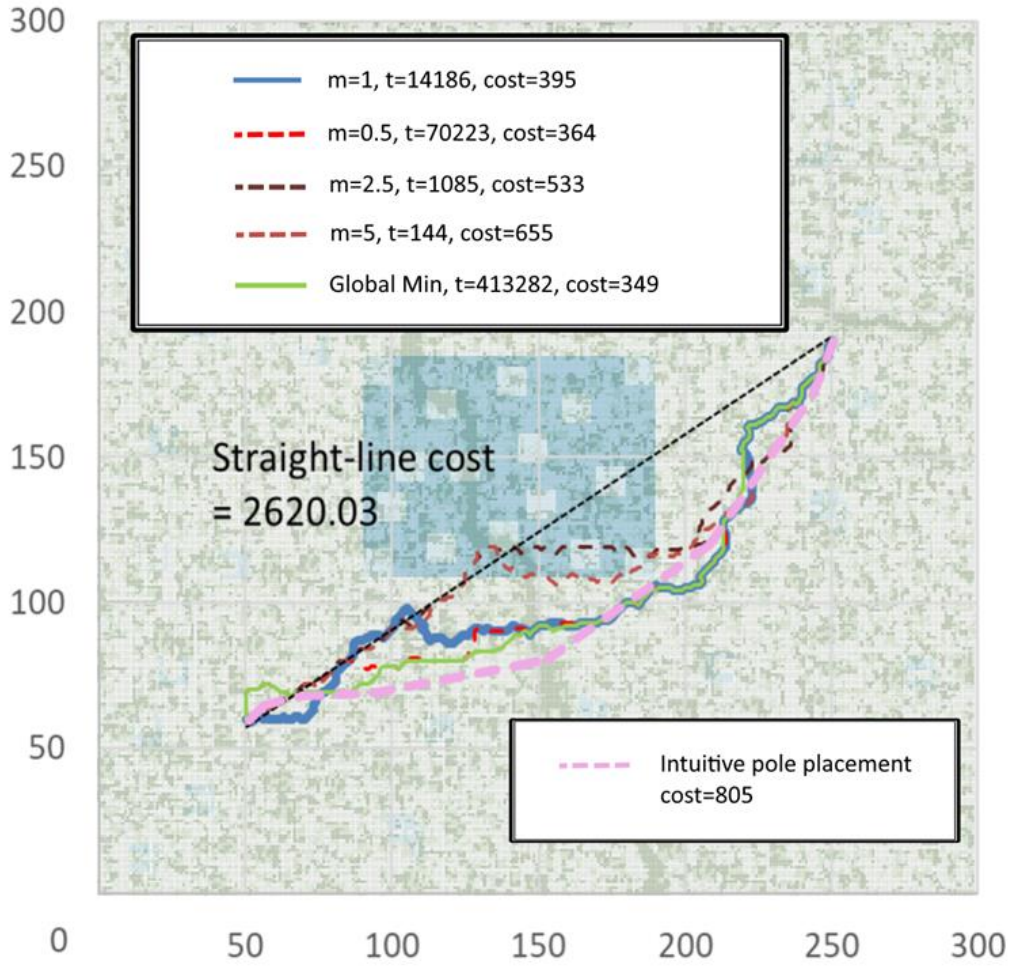


Figure 3. 8 **Actual path shapes: A* and MAA* @ c=10, for connection (50,60)-(250,190). Intuitive pole placement method is also included for comparison. A change from m=1 to m=5 reduces computational time by two orders of magnitude, but increases cost by 66%. Due to high anisotropicity, small displacement from optimal path can lead to significant change in path cost: Intuitive pole placement, which appears similar to the global optimal has a cost 2.3 x the global optimum.**

The figure also illustrates why a reasonably detailed map is essential for this problem. Unlike high voltage transmission where tall towers are used and cable routing is less affected by ground surface conditions, rural electrification uses low voltage and builds simple structures close to ground level. If one uses a high voltage methodology to connect (50,60) and (250, 190) by intuitively choosing a few "proper" intermediate points connected by straight line segments (the pink dotted-line in Figure 3.8), the cost of this "simpler" path (cost=805) is double that of the MAA*(1) path cost. An example of a longer connection is provided in Appendix 2C.

3.3 Designing optimal network for rural electrification

As explained in Section 1, designing optimal network consists of two steps: (a) identify all the optimal paths connecting any two villages in the area of interest, and (b) synthesize the minimum cost network based on the optimal paths identified.

3.3.1 Results of optimal path finding based on MAA* algorithm

As an illustrative example of complete network design, (near-) optimal connections among the 12 villages on Figure 3.9 are computed using MAA* algorithm with $m \in \{1,2,5\}$. With 12 villages, 66 routes are to be evaluated. Table 3D.1 in Appendix 2D summarizes the computation results. For comparison purpose, the straight-line costs and the R^2 distance are included in the table.

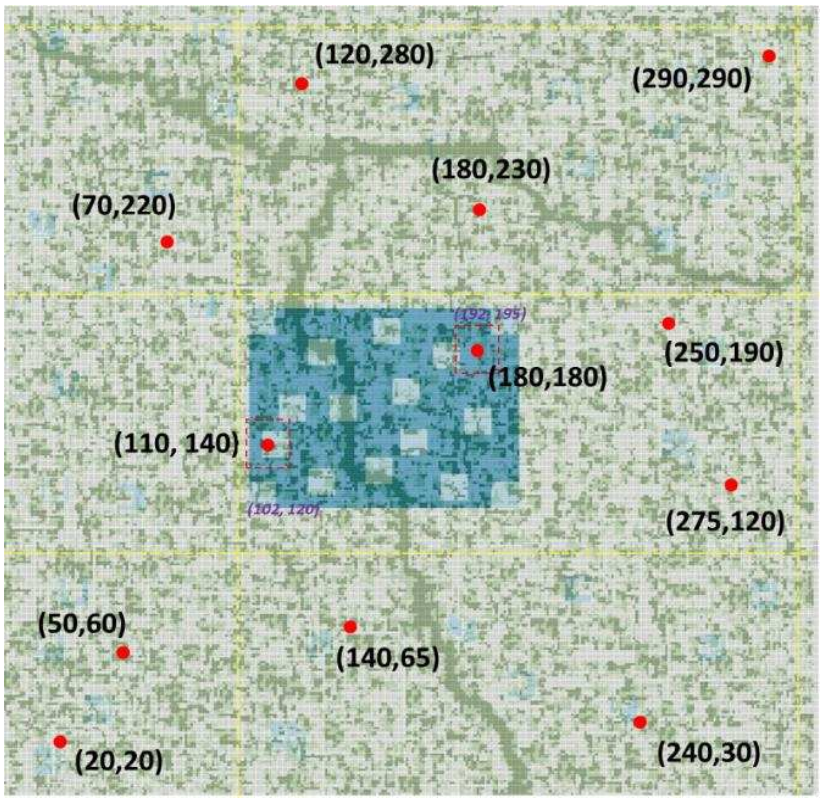


Figure 3. 9 Illustrative map

For most routes, MAA*(2) runs substantially faster than MAA*(1) while sacrificing modest optimality. Although this property cannot be guaranteed due to the randomness associated with path selection under different m, 45%, or 30 out of the total 66 routes computed with MAA*(2) are lying within 10% optimality impact compared to MAA*(1) in this example. In terms of speed, 55%, or 36 out of the 66 routes offers more than 75% computation time reduction compared to MAA*(1) (see Table 3D.1). If 20% optimality impact is allowable (e.g. during initial screening), 54 routes (i.e. 82%) computed using MAA*(2) fall within this limit. For computational efforts, 59 routes (i.e. 89%) offers more than 50% computation time reduction.

Certainly, the above comparison can only provide a rough impression about the accuracy and speed trade-off between MAA*(1) and MAA*(2). The MAA*(2) route with the best-preserved optimality is not necessarily also having great speed improvement. Different topographies can lead to very different results. Further analysis and guideline for selecting m values are provided in Appendix 2D.

In a collective sense, Table 3.4 compares the total cost and computation time for all 66 paths in this example. MAA*(2) requires 1/3 the computational time, while increasing cost by 12%. Thus, depending on complexity of the problem and accuracy required, network designer can make reasonable trade-off by selecting appropriate m value.

Table 3. 4 Computation Results using MAA* at m=1 and m=2

Algorithm	Computation Time (h)	Total Path Cost
MAA*(1)	883	27022
MAA*(2)	293	30301
MAA*(2)/MAA*(1)	33%	112%

3.3.2 Network synthesis using minimum spanning tree

Given near-optimal routes between all nodes, the standard minimum spanning tree (MST) algorithm may be applied to select routes for an optimal network. In practice, the network designer must also include capacity considerations in order to optimize capital cost while providing suitable voltage and frequency regulation. These factors can be included using a capacitated minimum spanning tree (CMST) [13], and the cost of additional computation time. Since this paper primarily aims to illustrate MAA* routing, we assume MST will provides a sufficiently representative network design. Figure 3.10 depicts the resulting network. Using moderate computational power, MAA* provides decision makers quick, low-cost, insight on whether electrification of an area or region can be done more effectively by networking villages rather than pursing separate electrical systems for each village.

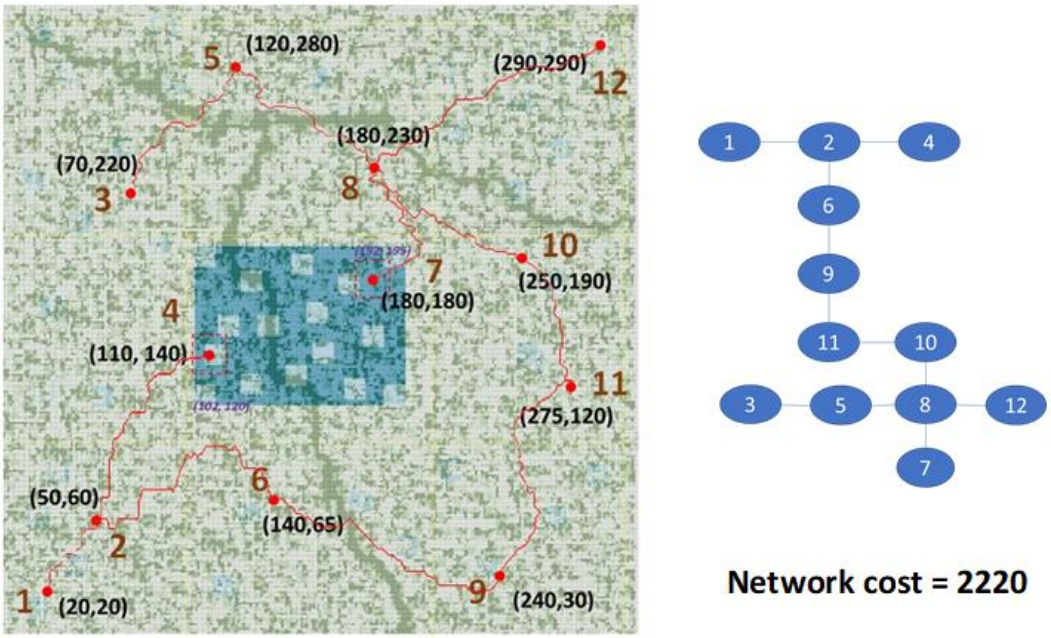


Figure 3. 10 Optimal network for rural electrification, obtained using minimum spanning tree (MST) method

3.4 Limitations

While this work has demonstrated a method for feasibly routing networked rural electrification, several limitations must be discussed.

First, the selection of locations, and resulting point-pairs (villages and generation locations), constrains the cost that can be calculated by *any* routing algorithm followed by *any* spanning tree algorithm: Lines may only branch at the defined points. In practice, judiciously inserting additional branch points may further reduce cost. For example, in Figure 3.11, connection 8-10 and 8-7 run nearly parallel to each other for some distance. Inserting an additional branch point between these three villages would likely be lower cost to the village-to-village connections. Choosing the appropriate position to combine the paths is another optimization problem by itself; ongoing research by the authors applies fuzzy logic to the identified network to further simplify the topology and reduce costs.

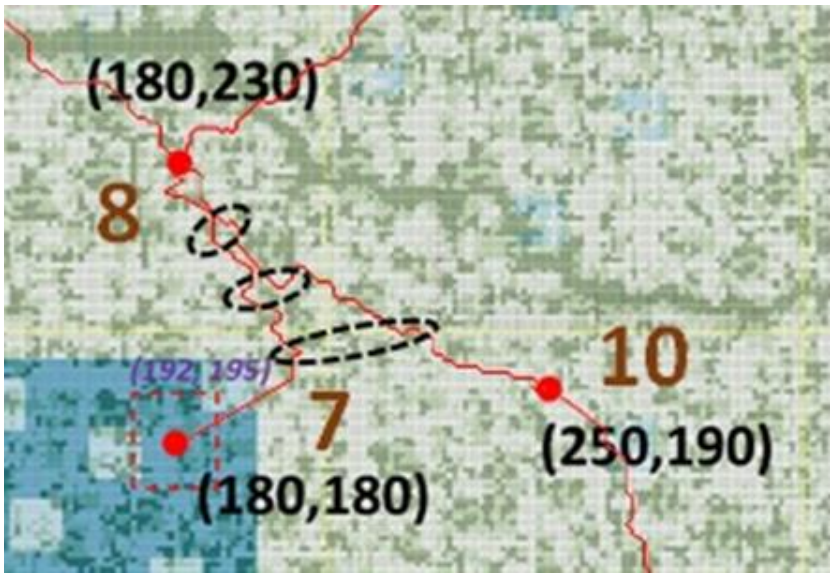


Figure 3. 11 Combining routes for further optimization

Second, the MAA*, like A*, does not address objective (2) introduced in Section 3.1: These algorithms provide a single routing solution, while practical applications would benefit from a family of solutions with similar, near-optimal, costs. A different algorithmic approach (under development by the author team) is necessary to meet this objective.

Finally, while MMA* algorithm can offer significant speed improvement over standard A* search in most cases, highly complex topographies that result in large heuristic error still challenge MAA*. An example is routing inside the low-accessibility, high cost, region centered in Figure 3.9, and enlarged in Figure 3.12. In this type of case, MAA*'s modified heuristic will not prevent the algorithm from exploring a large portion of the map. Computing the optimal connection between villages 4 & 7, requires 70 h with MAA*(5), and 664 h with MAA*(1). Based on the optimal path obtained by MAA*(1), the average heuristic error is 95.9%, and it takes 4.2 h to make one effective movement along the path. An adaptive multiplier-accelerated A* (AMMA*) algorithm has been proposed for tackling such topographies [31].

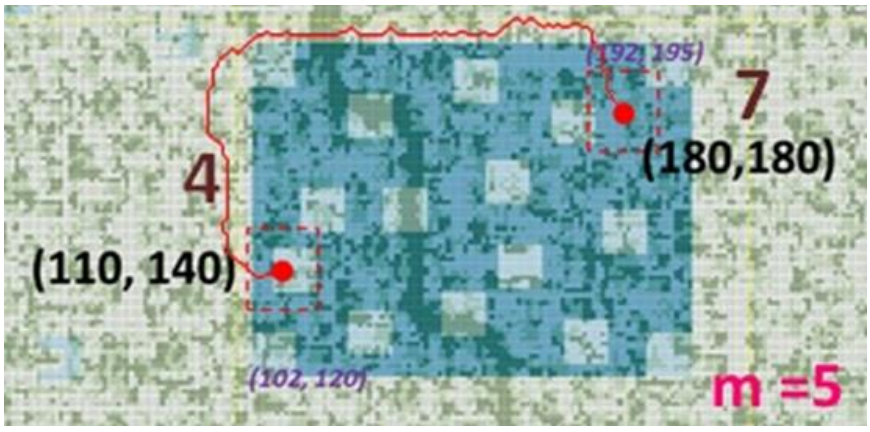


Figure 3. 12 Search under very complex topography

3.5 Conclusions

Networked rural electrification provides the possibility to supply grid-compatible electricity to remote villages at a cost-effective manner and is beneficial for achieving SDG 7. However, since these systems are

small, up-front engineering costs represent a substantial fraction of the overall system cost. Detailed surveys to place lines are therefore uneconomical. Further, we have shown that (fast) intuitive pole placement – i.e. engineering intuition – does not lead to cost-effective line routings. Therefore, a computational solution requiring minimal human intervention is needed.

The anisotropic nature of the village environment makes the standard method for path finding – canonical A* -- computationally unattractive; supercomputers would be required even for regions as small as that used here for examples. For practical project regions, the computational costs would be unacceptable.

The proposed multiplier-accelerated A* (MAA*) algorithm in this work significantly reduces the computation burden while maintaining the necessary spatial resolution and producing near-optimal line routings. A further advantage of this algorithm is the ability to trade-off between computation speed and optimality. During early stages of a proposed project, inputs must be accumulated at low cost, may change rapidly, and more possible designs must be assessed to determine if the project is worth pursuing. At later stages, when it is more cost-effective to accumulate additional information, a more computationally intensive approach is desirable to optimize and finalize the estimated project cost. This type of adjustment is impossible for classical methods that produce a single, optimal solution. As demonstrated, the near optimal solutions can be obtained at the resolution of 10m x 10m over a 3km x 3km map. With this resolution, the obtained paths are sufficiently optimal and also physically implementable. This would mean that they can be cost-effectively and accurately built with simple pole-and-cable structure over the entire site, and would be less distributing to the residents because accessibility of each 10m x 10m square has been accounted during optimization.

Finally, the same algorithm could be applied to other, similar, routing design problems, i.e. problems where many possible routes must be considered, and costs and constraints vary continuously and rapidly. Examples include routing gas pipelines or buried communication lines where other infrastructure, natural features like rivers, or soil composition have a large impact on capital costs. In these examples, the grid

resolution and map size would be scaled to match the problem's routing constraints and the available topographical data.

CHAPTER 4 – ADAPTIVE MULTIPLIER-ACCELERATED A* ALOGRITHM [2]

In Section 3.4, some of the limitations of MAA* optimization method have been discussed. For some highly complex regions, even MAA* cannot search effectively due to a phenomenon we termed as "path oscillation". This chapter discusses the mechanism behind this phenomenon and the AMAA* method we developed to resolve the problem.

4.1 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 [58] [22] [23]. In addition, location of the village or town may not be optimal for efficient generation [23] [59] [12].

To facilitate electrification for these difficult areas, a networked rural electrification scheme has been proposed [29]. 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

[2] This chapter was published in the *proceedings of 2020 IEEE / ITU International Conference on Artificial Intelligence for Good (AI4G)* as "Optimizing Networked Rural Electrification Design Using Adaptive Multiplier-Accelerated A* Algorithm" [31].

and/or wind for efficient generation, and (ii) supplementary local generation modules that can be added or removed based upon local demand (Figure 4.1).

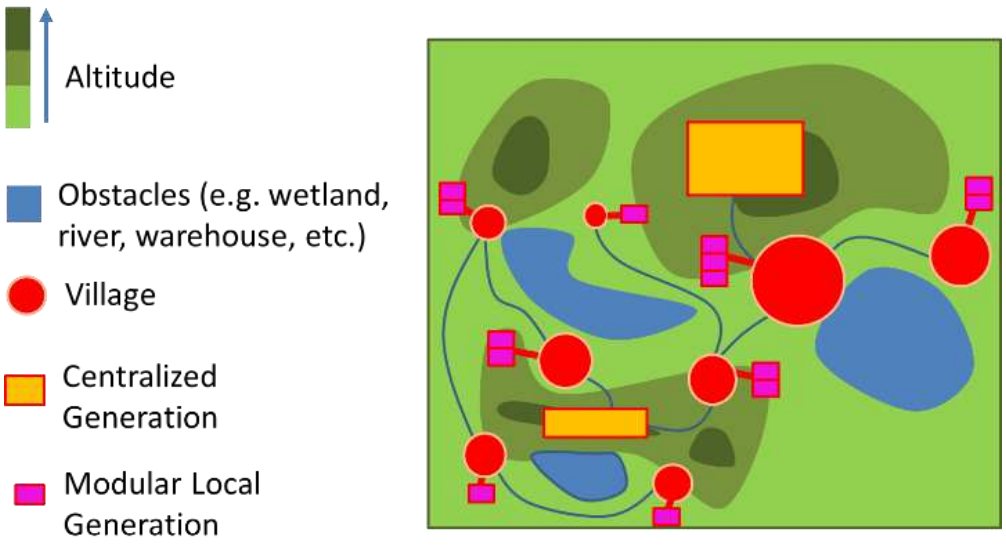


Figure 4. 1 Networked Rural Electrification

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 [29] 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.

4.2 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 $O(b^{\epsilon d})$ [38], 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 4.2 illustrates the effect of ϵ and d under a branching factor of, for example, 2.13 with constant step cost.

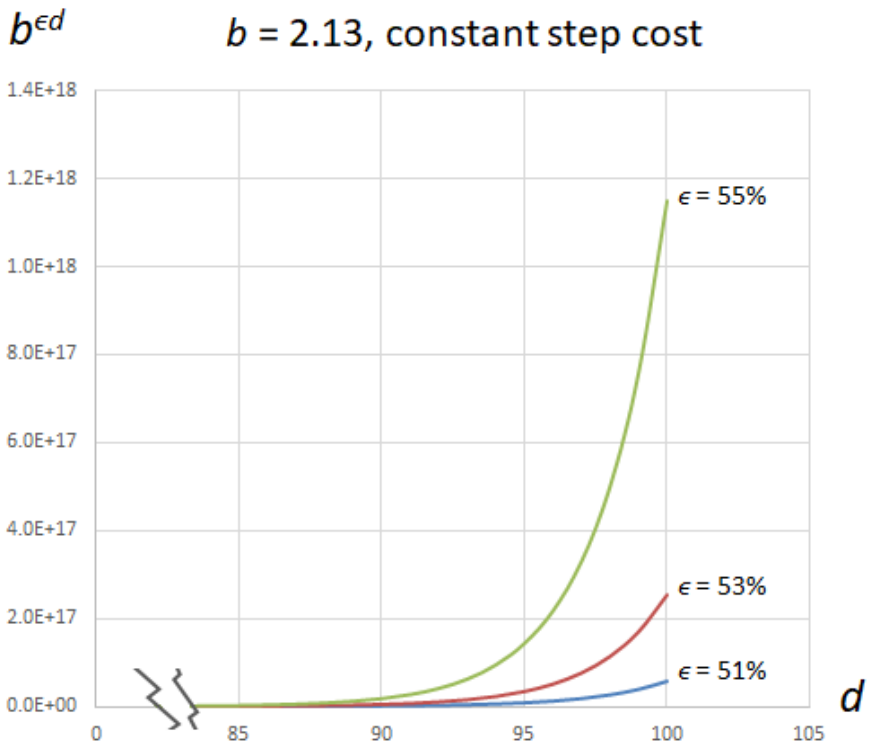


Figure 4. 2 Time complexity vs search depth

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 [36] [60]. Standard A* algorithm can then be applied to these waypoints. Many other acceleration techniques [35] [61] 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 [51] [50] [52], 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 4.3.

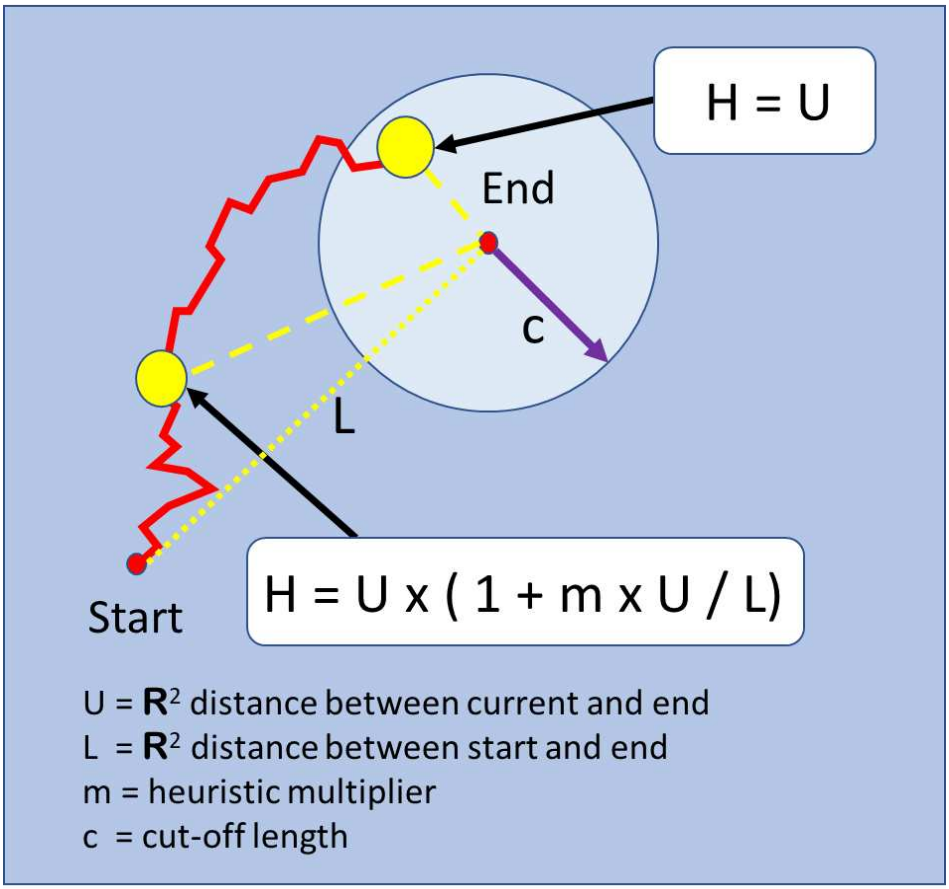


Figure 4. 3 Multiplier-accelerated A* algorithm

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

$$H = U \times (1 + m \times U / L) \tag{4.1}$$

where U is the R^2 distance between current and end points

L is the 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 (4.1) 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* [39]. 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 4.4 and Table 4.1 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 4.5). Details are provided in [29].

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.

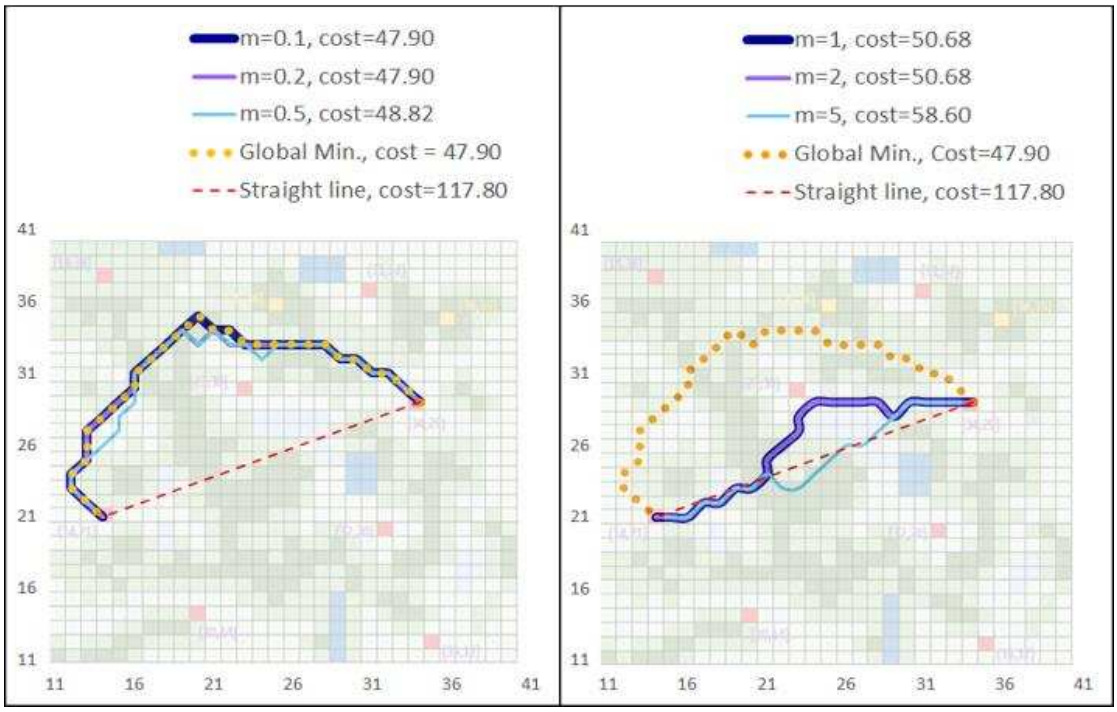


Figure 4. 4 Optimal paths connecting (14,21) and (34,29)

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 4. 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.

4.3 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 4.5, there are two target connections A and B on a 300 x 300 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.

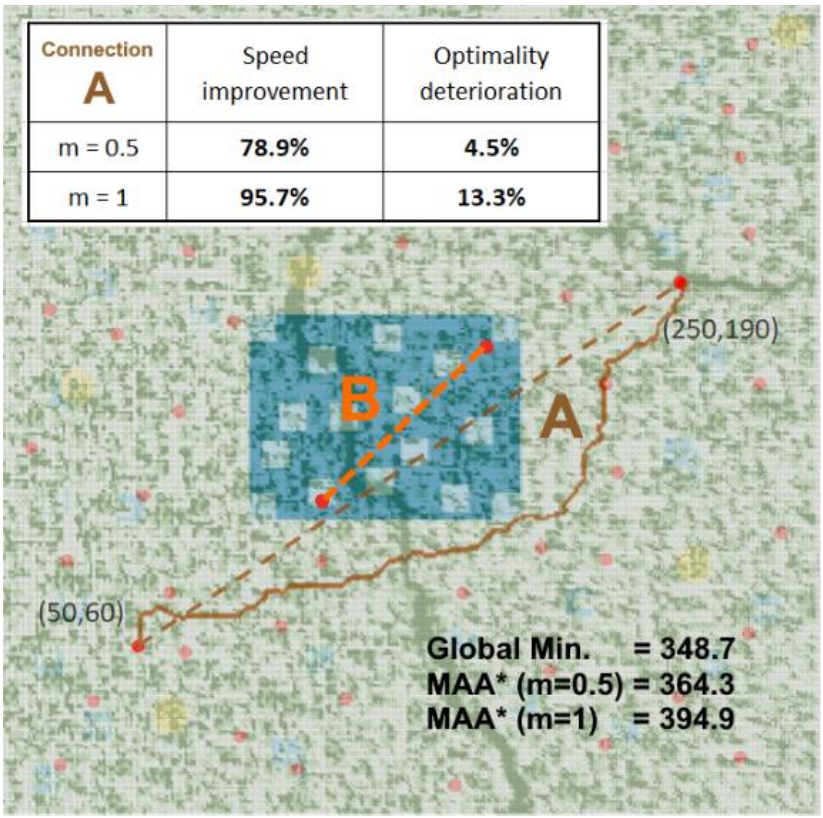


Figure 4. 5 Performance of MAA* on a 300 x 300 map

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 4.6, for *m*=5 and 10). Unfortunately, the MAA* computation time for this case is extremely long even for relatively large *m*.

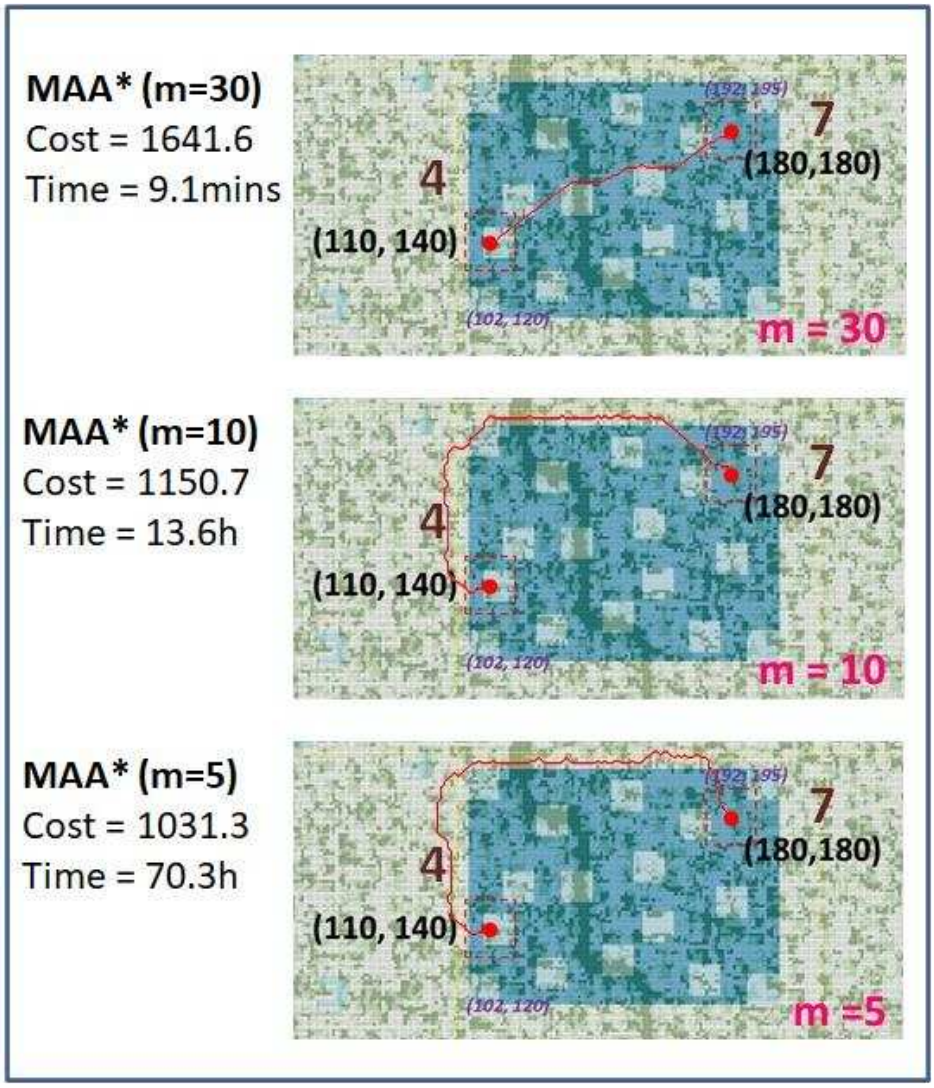


Figure 4. 6 Searching costly zone

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 4.6, 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 4.2).

Table 4. 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%

4.4 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 4.7) according to the specific structure of the search space, and then applying MAA* algorithm to the newly constructed line segments.

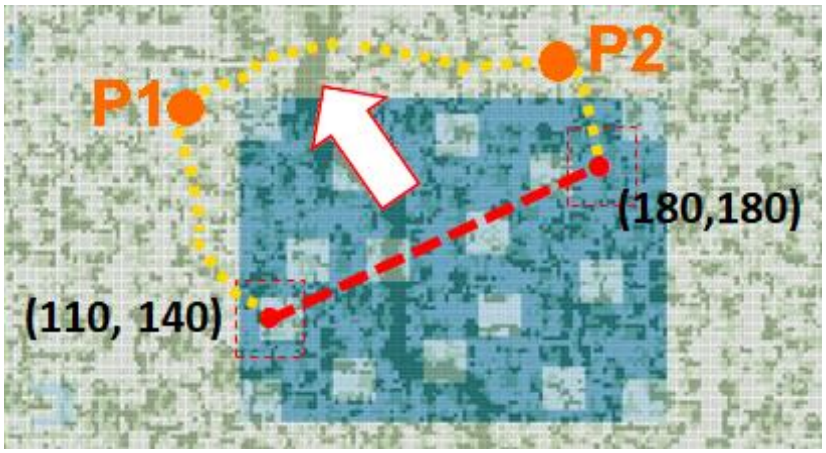


Figure 4. 7 Principle of AMAA*

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 4.8.

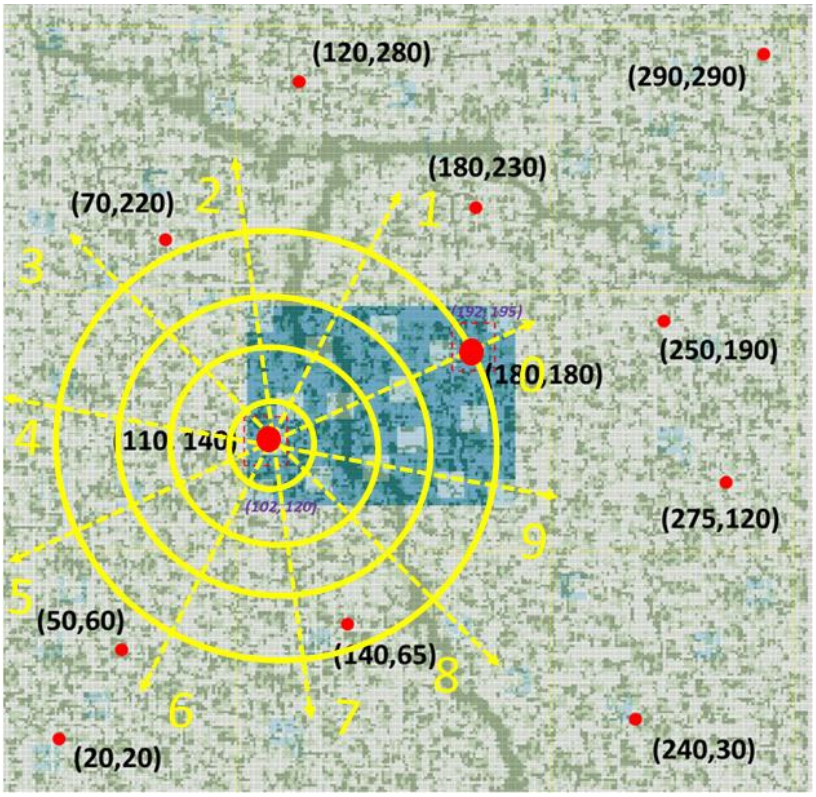


Figure 4. 8 Screening for intermediate node

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.

iv. 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.

v. For the $2r$ nodes on each circle, calculate the R^2 distance to the end node, compared to the 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;

vi. For the $2r$ nodes on each circle, costs for moving from start node to these nodes are calculated and normalized by corresponding 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;

vii. 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 4.3 and 4.4 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 4.3 is the second of the 10 tables, and serves as an illustration of the format and information contained.

Table 4. 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 4.4).

Table 4. 4 Summary of final score

		Circle									
		1	2	3	4	5	6	7	8	9	10
Direction	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 4.4, 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 4.9)

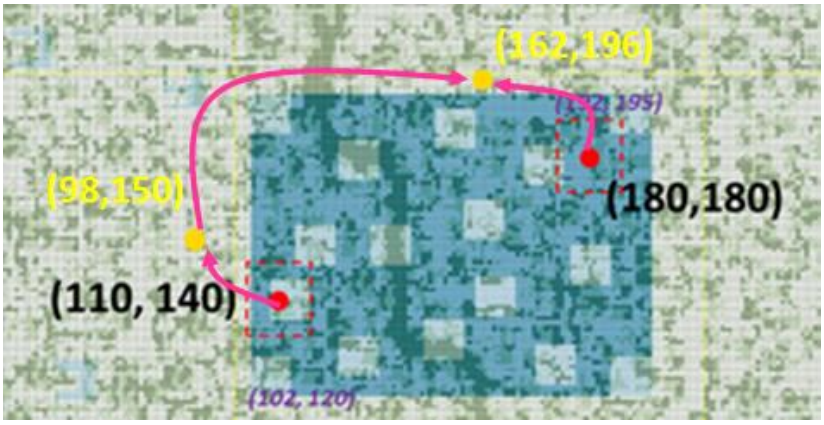


Figure 4. 9 Optimal path finding by AMAA*

Results obtained by using AMAA* algorithm are summarized in Table 4.5.

Table 4. 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 4.2, 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 4.6 summarizes the differences.

Table 4. 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.

4.5 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.

CHAPTER 5 – FLEXIBLE NETWORK DESIGN USING LEVELIZED INTERPOLATIVE GENETIC ALOGRITHM [3]

5.1 Introduction

Ensuring availability of modern energy to everyone by 2030 is an aggressive goal stated in the seventh Sustainable Development Goals (SDG7) of the United Nation [1]. While encouraging progress has been reported, it is also well-recognized that providing energy to the remaining unserved population is increasingly challenging [4] [5]. Very often, this populations lives in small farming clusters (i.e. villages) spreading in remote, difficult-to-reach areas. Current electrification practices tend to focus on photovoltaic (PV) generation, as it can be readily constructed in most areas and requires minimal expertise to maintain. However, accurate capacity planning to optimize system economics is challenging for these small systems due to uncertainties in both demand and supply [22] [23]. Very often, this will lead to prohibitively high cost for electrifying these remote areas. On the energy supply side, while the solar radiation statistics aggregated over large areas is reliable, radiation over specific, small, target area is less available and typically less reliable. Furthermore, locations of these villages were historically determined based on soil and water resources, safety, and similar factors, and may not be optimal for PV generation. This would further increase cost of electrification. On the demand side, population changes in rural areas are difficult to predict in mid-term time frames (5-20 years) complicating load growth prediction and capacity planning. In addition, even short-term demand forecasts are unreliable due to small population. As a result, over-investment and under-investment are common for these small systems.

Since up-front planning is necessarily incomplete, an alternative solution is to implement rural electrification using a combined local and networked supply approach to produce better generation

[3] This chapter has been submitted to *Energy and AI* for publication and is currently under review [32].

performance and utilization, and to accommodate population shifts and differential load growth within a rural area (Figure 1) [29] [30].

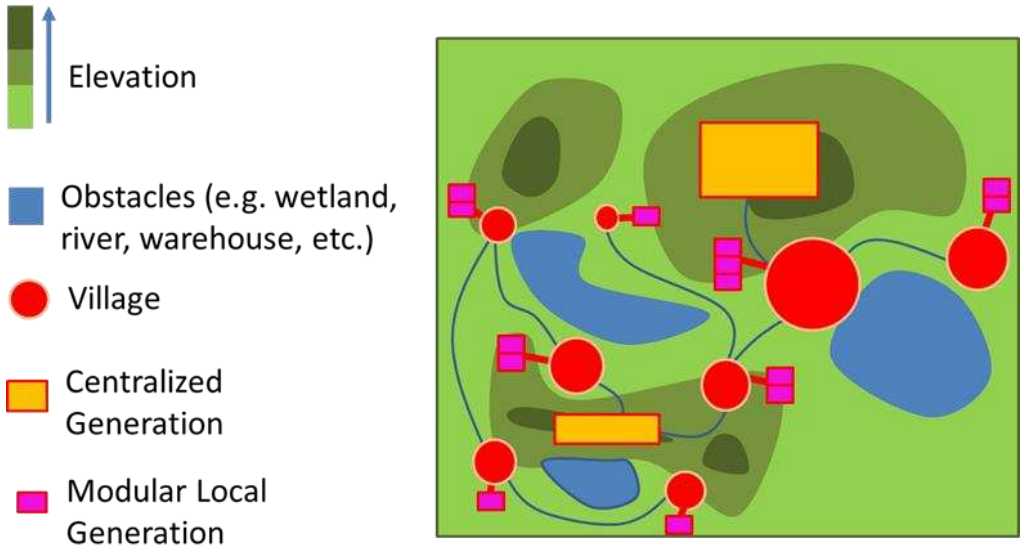


Figure 5. 1 The networked rural electrification framework

In this framework, while each village may be equipped with some generation, a group of villages in a wider area are connected via a distribution network. To achieve cost optimality, this system is likely not a simple radial system, which is common in rural distribution systems [12] [8], but would use a networked system where the villages are supplied mainly by one or more centralized, larger, PV facilities sited for generation efficiency, while the per-village systems provide local generation to the network and backup or emergency generation [29] [30]. Viability of this framework depends on whether the improvement in generation efficiency and reliability can compensate the cost of building the distribution network.

Unelectrified village clusters are generally in areas that are distant from population centers and/or have rugged terrain. For this work, we focus on areas with rugged terrain where relatively small changes in distribution line routing may have substantial impacts on the construction and maintenance costs. Examples include Mentougou district of Beijing, China or the tropical highlands of western Rwanda, where the authors have worked. In these areas, villages which are near each other horizontally may be separated by

large changes in elevation, such as mountain ridge lines or river canyons. These regions also have 'keep out' areas, such as nature reserves or government holdings that are unsuitable for distribution lines. We refer to these rapid variations in the cost of construction as 'anisotropicity.' Therefore, simplistic route planning, connecting 'nearest villages by direct routes,' as would be used on a larger scale for transmission lines, will lead to a significantly sub-optimal distribution network in highly anisotropic environments.

To improve viability, the distribution network must be implemented at the lowest possible cost, given available information. Therefore, the network design is fundamentally a routing, or route-finding, problem, consisting of (i) finding the cost for the (near) optimal path connecting any two nodes, where a node may be either a village or a centralized generation facility, as in Figure 1; and (ii) identify the lowest possible network cost based on the path costs computed in (i). Classical A* pathfinding [28] can, in theory, be applied to accomplish task (i), and minimum spanning tree (MST) or a derivative thereof used for task (ii). However, as shown in [29] [30], classical A* is extremely inefficient for this problem due to anisotropicity associated with the complex topography of these remote rural areas. Additionally, commonly utilized A* acceleration methods are not applicable for a highly anisotropic search space. Therefore, the authors developed a multiplier-accelerated A* (MAA*) and adaptive multiplier-accelerated A* (AMAA*) algorithms to effectively accelerate pathfinding in this anisotropic space [29] [30] [31]. These algorithms produce near-optimal solutions with significantly reduced computational effort. Compared to classical A*, MAA* generally reduces computation by 90% while impacting optimality by less than 10%; the trade-off between optimality and computation time can be defined by the user. AMAA* extends MAA* for complex situations where paths originate in, or must maneuver around, a relatively large area with high routing costs. Used in conjunction with MST, Figure 2 is an example of an optimal network based on these algorithms.

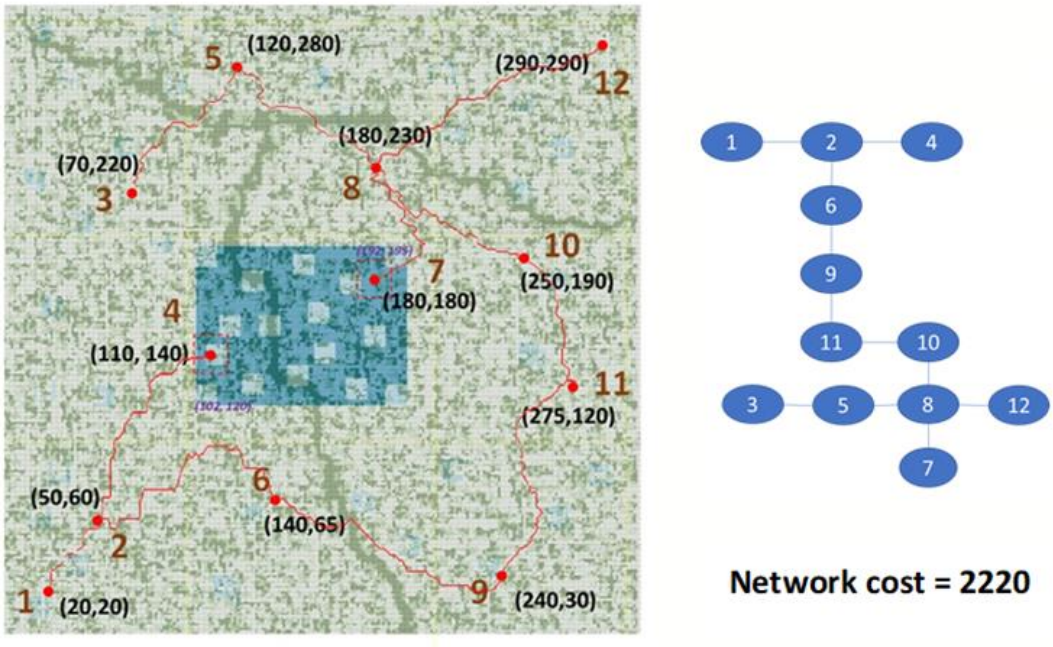


Figure 5. 2 Typical optimal network

Using MAA* / AMAA* method to design an optimal network, including centralized generation locations, total cost can then be compared to the traditional per-village approach to rural electrification. If the computed cost of the networked approach is considerably lower, networked rural electrification is a lower-cost solution for the particular topography and village distribution.

However, the near-optimal network designed by these methods is not necessarily robust to real-world conditions – namely the inevitably incomplete or inaccurate information available when the optimization was completed. For example, to reduce costs, the initial map is typically obtained from satellite photos and digital elevation models. During digitization, it is likely that some land conditions may not be properly categorized or some land features may not be visible. Even when physically accurate, there could also be unexpected situations such as ownership disputes, changing land-use policies, etc. that may change the cost of routing. Therefore, a method that produces multiple, geographically distinctive network topologies with similar low costs allows field planning teams to adjust routing during the detailed planning phase of the project. A*-derivative algorithms, by design, find only one routing solution to any problem. A superior

algorithm – from the point of view of real-world application – would produce multiple routes, with similar cost, allowing routing adjustments as needed.

It is important to note that this same problem of imperfect information is common in other infrastructure routing situations. For example, a gas pipeline through several eastern USA states has been strongly opposed over a short section of its route in West Virginia [62] due to environmental concerns which were not anticipated by the pipeline developers, exacerbated by the lack of pre-planned, viable, alternatives at similar cost.

This paper discusses a hybrid path-finding method combining genetic algorithms – which naturally identifies families of near-optimal solutions – coupled with the previously-discussed A* derivatives to provide multi-path solution to optimal path-finding problem. This approach results in multiple near-optimal networks that can be used to validate a potential networked rural electrification approach, while retaining flexibility during implementation. This paper begins with a more precise definition and discussion of multi-path solutions in Section 2, followed by a description of algorithm in Sections 3-5, followed by discussion and conclusions in Section 6.

5.2 The Necessity of Multi-path Solution

The MAA* and AMAA* algorithms, as is the case with most routing algorithms, operate on a discretized map (search space) where critical parameters impacting routing cost are assigned to each discrete element of the map. In the classic road-routing case, the discrete elements are road segments; in the case of distribution line routing, the discrete elements are small map areas (10m by 10m squares in this work), since the distribution line may theoretically be routed through any square within the map, since roads and trails in rural areas are typically informal, and are of little use for the heavy equipment needed to construct distribution systems. This type of discretization is commonly utilized for similar problems [54] [55].

A key challenge with designing rural electrification systems is the up-front engineering costs for design – primarily siting and sizing. In particular, an on-ground survey of a large area is often prohibitively expensive for the engineering team, and too technically complex for local residents. Therefore, the discretized maps are typically constructed from externally available data, including atlases, ownership records, and, increasingly, satellite photography and digital elevation models. These data sources and analysis methods, while relatively robust, often miss subtle complexities in the local environment, ranging from hidden obstacles to ownership disputes.

Following the up-front engineering and financing, a more detailed plan is completed by sending an expert survey team to verify distribution routes defined by the *preferred near-optimal network* previously developed. Frequently, survey teams will uncover complexities missing from the original mapping such as property rights or land-use disputes, ground/soil conditions, pressure groups or changing government policy. Without an alternative routing, redesigning the project under time pressure can consume significant resources, increase costs substantially, and will likely produce a compromise solution that is farther from the optimal design.

Two types of rapid re-routing are possible: (a) an *equivalent* near-optimal routing connects the same nodes together via different paths, and (b) an *alternative* near-optimal routing connects all nodes, but in different pairs with different paths. Together, these three elements define a multi-path solution: (a) a preferred optimal path, (b) a family of *equivalent* near-optimal networks (topologically equal to, geospatially unequal to, the preferred network), and (c) a family of *alternative* near-optimal networks (both topologically and geospatially unequal to the preferred network). These families are schematically illustrated in Figure 3. In general, substituting an equivalent network is less intrusive than alternative networks, and is therefore preferred.

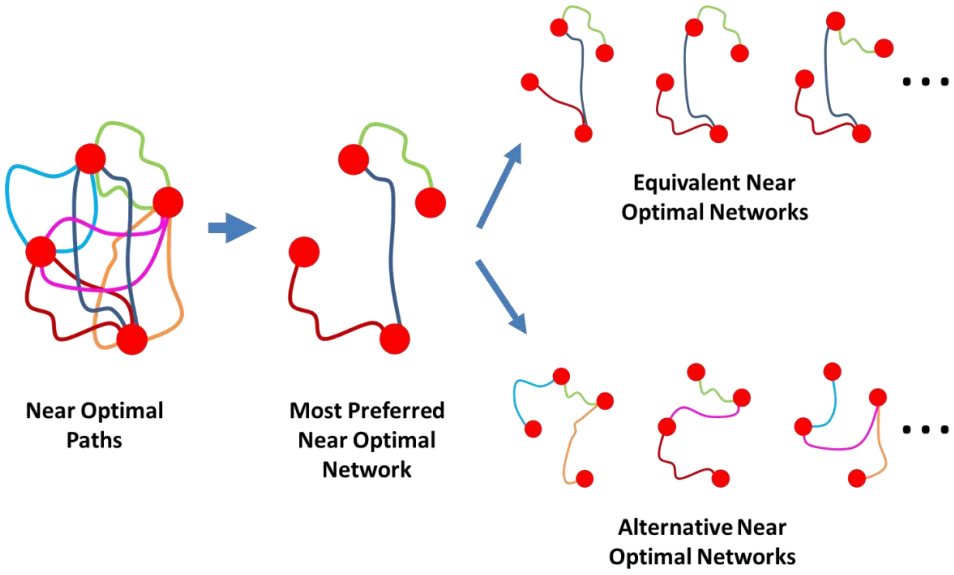


Figure 5. 3 Advantages of multi-path solution

5.3 Using Genetic Algorithm for Path Finding: Why and How?

Genetic algorithm (GA) can be used as an alternative approach to find the optimal path connecting node pairs on our test map. However, for reasons that will become apparent below, GA will not generally deliver solutions with comparable optimality to A*, MAA* or AMAA*. Figure 4 from reference [30] shows that the solution computed by GA (labelled as "intuitive pole placement") on a typical test map can be twice as costly as the A* solution. Further explanations are provided in Section 3.2.

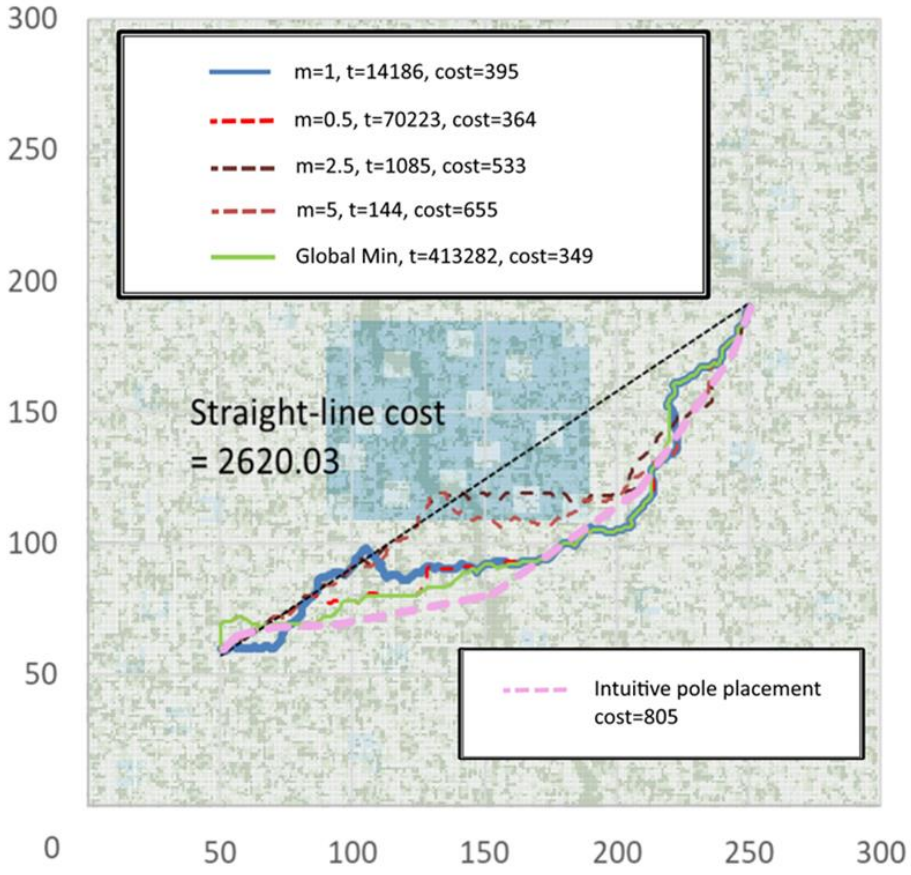


Figure 5. 4 **Comparing computed optimal path costs based on genetic algorithm and A* (and MAA*)**

The reason for examining GA is because it naturally finds multiple solutions with similar costs. This section briefly explains how GA can be applied to path finding problems. Based on this insight, multi-path solution is discussed in Section 4.

5.3.1 Application of Genetic Algorithms for Path Finding

In this section we use a simplistic application of a genetic algorithm (GA) to introduce the pathfinding problem. Genetic algorithms [63] [64] are widely used for optimization problems. The method of applying GA varies with the problem under investigation. As in our previous works [29] [30] we define the cost of the $(i+1)^{th}$ movement in a path as:

$$C_{i+1} = [(z_{i+1} - z_i)^2 + (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{\frac{1}{2}} + w_1|z_{i+1} - z_i| + w_2 a_{i+1} \quad (5.1)$$

where $x_i, y_i, z_i,$ are the 2D coordinates and elevation of the i^{th}point in the path, a_i is the accessibility of the i^{th} point on Figure 4, and w_1, w_2 are weighting coefficients. The objective of the pathfinding algorithm is to minimize $\sum_{i=0}^{P} C_{i+1}$, where P is the number of moves through discrete squares of the maps in a path. A basic, intuitive, algorithm is (Figure 5):

(a) Map a straight-line connecting the two nodes;

(b) Calculate the initial path cost base on the straight line path;

(c) Generate n random nodes around the straight-line path and connect to create a possible path;

(d) Compute the path cost for the new path;

(e) Repeat the process to generate a starting population of new paths;

(f) Sort the generated paths according to descending path cost;

(g) Select the first s paths.

(h) Apply a genetic algorithm to find an optimal path by progressive mutation of the initial population of s paths. For this problem, genetic cross-over is defined as swapping the selected node between paths and mutation is defined as a random movement of one of the nodes in a path.

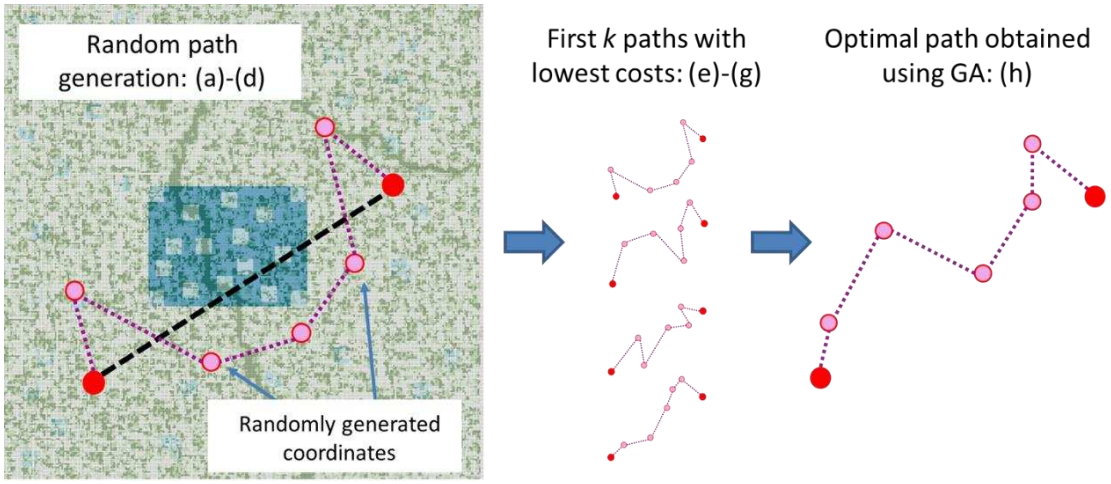


Figure 5. 5 Using genetic algorithm to obtain optimal path in this application

5.3.2 Performance of the Simplistic Application in Highly Anisotropic Conditions

Unlike A* algorithm where the obtained optimal path can essentially take any physical shape, optimal path obtained from traditional GA is simply constructed by $(n + 1)$ straight-line segments. Therefore, a GA operating without assistance is unlikely to provide the same optimality as an A*-derivative algorithm, due to suboptimal routing along the straight path segments in the highly anisotropic cost environment. However, when the R^2 distance between the start and end nodes is large, GA can be conveniently used to identify an approximate location of the optimal path as a series of straight-line segments. Each straight line can then be routed using A* (Figure 6), which is computationally tractable since the line segments are relatively short and search area will be small unless extreme anisotropicity is encountered [31].

For this application, the choice of n drives both the cost and complexity of the algorithm. Larger n allows a more complete exploration of the map and reduces the size of each A* routing, but also leads to many offspring that are high-cost zigzag paths (due to mutations) and generally unfit. Experimentally, a small n of 3 or 4 nodes generates paths with costs lower than the initial straight-line, while $n > 7$ generates many paths with higher cost than initial choices from step (g).

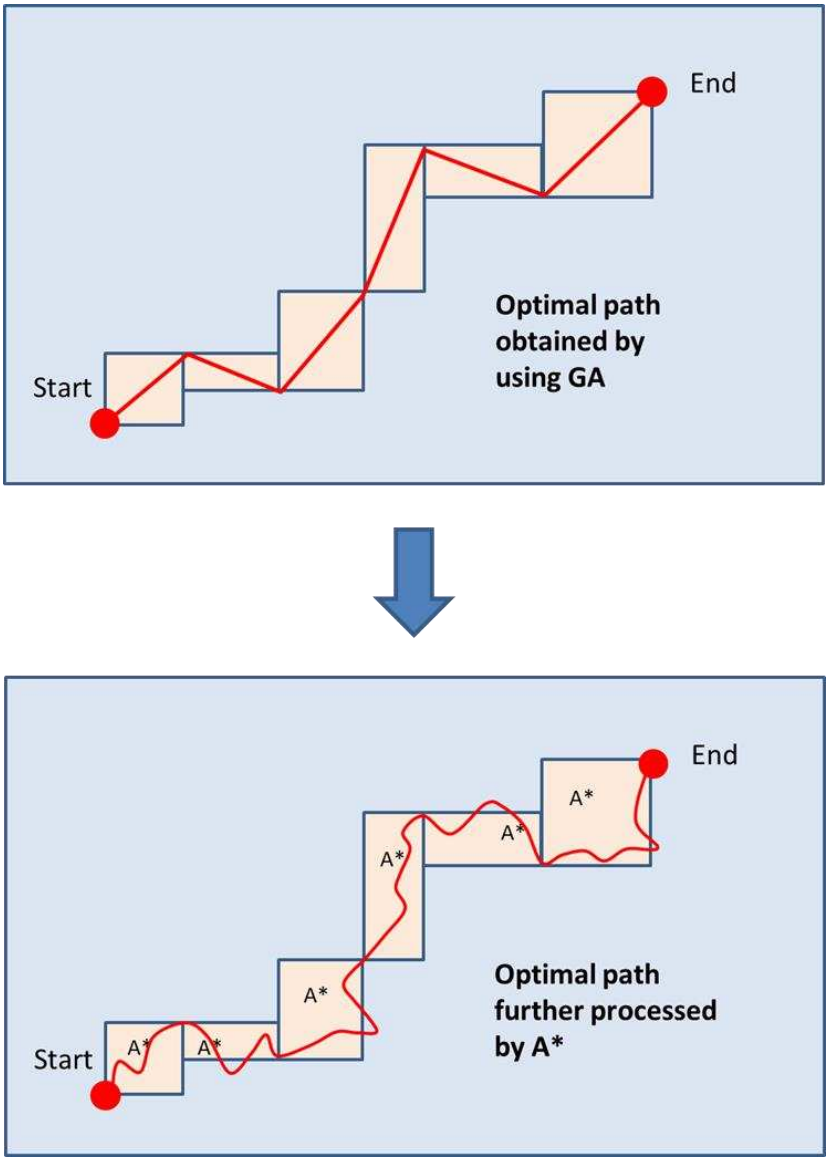


Figure 5. 6 **GA+A* hybridization scheme. In the upper panel, an approximate path, consisting of a small number of straight-line segments, is developed using a genetic algorithm (see text). In the lower panel, detailed routing is completed using A*, constrained to the indicated box regions, which reduces the cost of A* routing in the highly anisotropic environment.**

As an example, Figure 7 compares costs of an example hybrid routing – the globally optimal path between the two nodes – with that from the simplified GA example, and the hybridized GA+A* algorithms. For this example (one of many test cases performed), the GA result has evolved 30 generations.

As expected, A* has a much better performance than GA, and the hybrid algorithm lies in between. It should be remarked that the GA in the example has executed for 30 generations, and result of GA is generally not unique. As an additional comparison, cost of the path computed using MAA*(1) is 395 (Figure 4). [Note: the number m in MAA*(m) is the acceleration coefficient for speeding up A* algorithm. In general, computation speed is faster when m is larger, but resulting paths are farther from optimal. For most practical applications, m=1 reduce computation time by $\approx$90% while optimality is reduced by $\approx$10% level. See [29] [30] for details.]

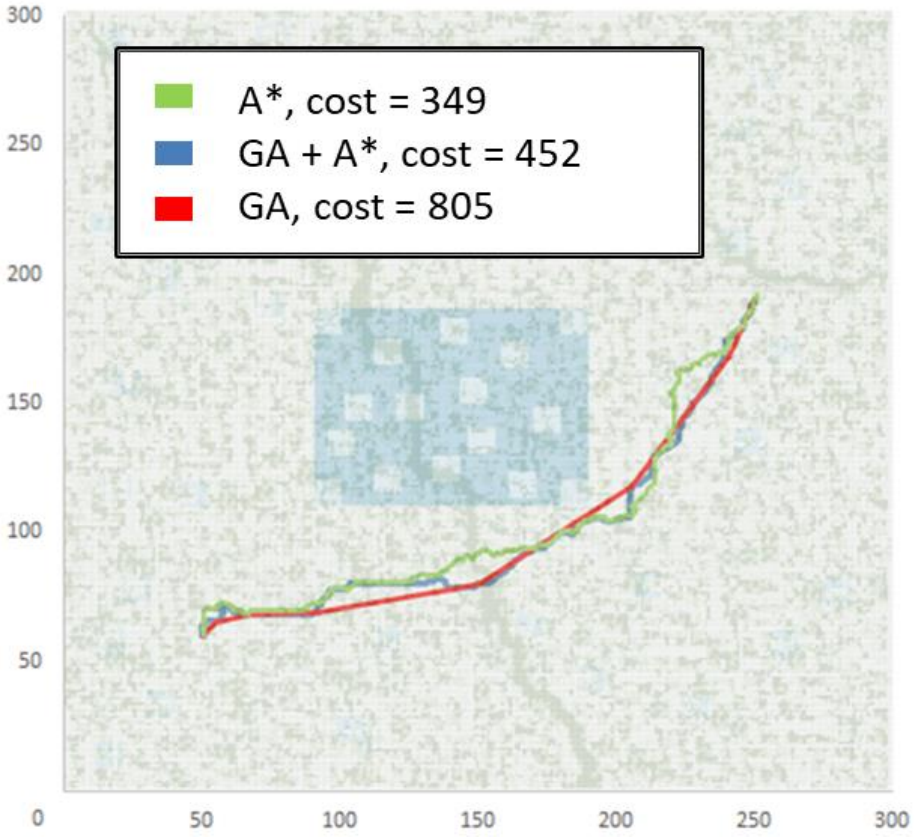


Figure 5. 7 **Resulting path cost from A* (global optimum) compared to genetic algorithm (GA) and the hybrid algorithm with initial genetic routing followed by A* detailed routing (GA+A*). The GA route cost is 805, but the hybrid application of A* to detailed routing drops this cost by 45% to 452. As expected, while neither method achieves the cost of the globally optimal (43% of GA), the GA+A* produces near-optimal results (56% of GA). Note that the GA algorithm readily bypasses the high-cost area between the two endpoints; the A*-derivative algorithms explore large sections of the total map to reach the same conclusion.**

From these experiments, we conclude that the GA solution, followed by local improvement using A* (or MAA* / AMAA*), produces acceptably low-cost solutions. This is primary premise behind the Levelized Interpolative Genetic Algorithm (LIGA) described in this work: The LIGA algorithm coarsely outlines near-optimal routes connecting two nodes using GA, and then completes local routing using an A*-derivative algorithm (A*, MAA*, or AMAA*) to refine the coarse routes. The idea is shown in Figure 8.

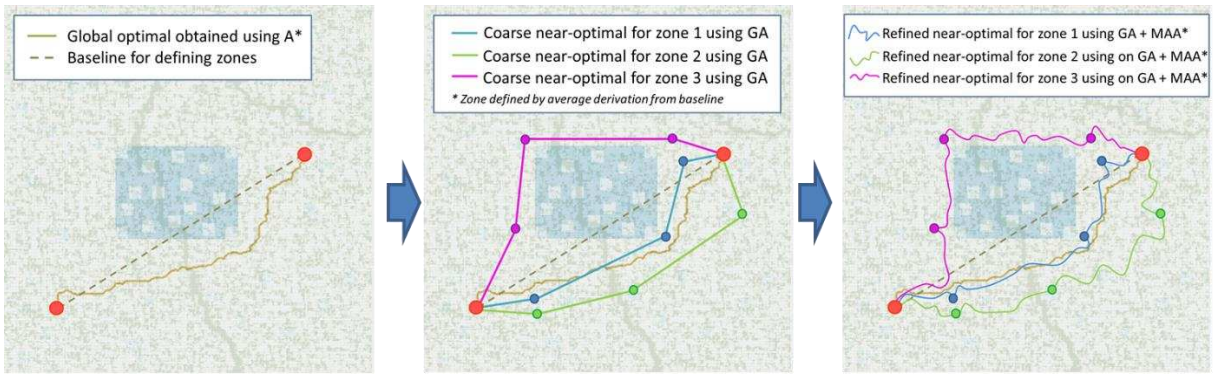


Figure 5. 8 **Generating near-optimal path in different zones by using GA and MAA*. Routing around the high-cost region in the center of the map is challenging for the A*-derivative algorithms, but readily accomplished by the LIGA algorithm.**

It should be noted that while Figure 8 illustrates the primary concept, a further refinement (i.e. interpolation) has to be introduced for increasing degrees of freedom during the optimization process in order to achieve better optimality. Details are discussed in Section 4.

5.4 Levelized Interpolative Genetic Algorithm (LIGA)

In most GA applications, retaining a wide variety of genotype is encouraged [63] [64]. However, in a subset of applications where there is more than one group of dominant near-optimal solutions, crossing-over between groups will unlikely produce better results (although it is not strictly impossible) [65] [66]. Additionally, crossing-over between these groups significantly increases computation burden. From our

work, applying GA for path finding under high anisotropicity exhibits this issue. The initial selection of paths will quickly generate more than one group of near-optimal solutions. For example, paths that 'go right' or 'go left' around a compact obstacle may similar costs, yet be geographically distant. In these cases, non-selective crossing-over produces many suboptimal routes – in the example, routes that cross the obstacle – increasing computational load and reducing the probability of improvement between generations.

If the genotypes can be grouped by proximity – as defined using their relative distance from the straight line connecting the start and end nodes in our case – each group may separately converge to a few dominant, near-optimal, solutions quickly (Group 1-4 in Figure 9). In this application, while crossing-over between genotypes in Group 2 and Group 3 may produce a viable new solution via connecting through the grey dotted line in the figure, crossing-over between genotypes in Group 1 and Group 4, for example, will unlikely lead to any meaningful improvement.

Obviously, proximity, or any other single metric, may not be ideal for grouping genotypes. For example, crossing-over between Groups 2 and 3 is more likely to generate viable paths than Groups 1 and 4 (gray dotted line). While more sophisticated group schemes could be used, most also increase the computation burden, or are hard to configure given the information available when the optimization is most needed – i.e. in early project planning.

Additionally, the second routing step, using MAA* or AMAA*, also has the ability to identify near-optimal routes that deviate somewhat from the paths between the genotype nodes. Therefore, a cross-over between Group 2 and Group 3 *may* give a viable new path, but the cost of this new path is unlikely to be better than the near-optimal obtained using MAA* or AMAA* between low(est) cost paths in either Group 2 or Group 3. Thus, for the purpose of this work, obtaining a better path in the vicinity of other near-optimal paths is not as important as finding a family near-optimal paths that have wider geographic coverage and provide options during the implementation phase … and all within reasonable computational effort.

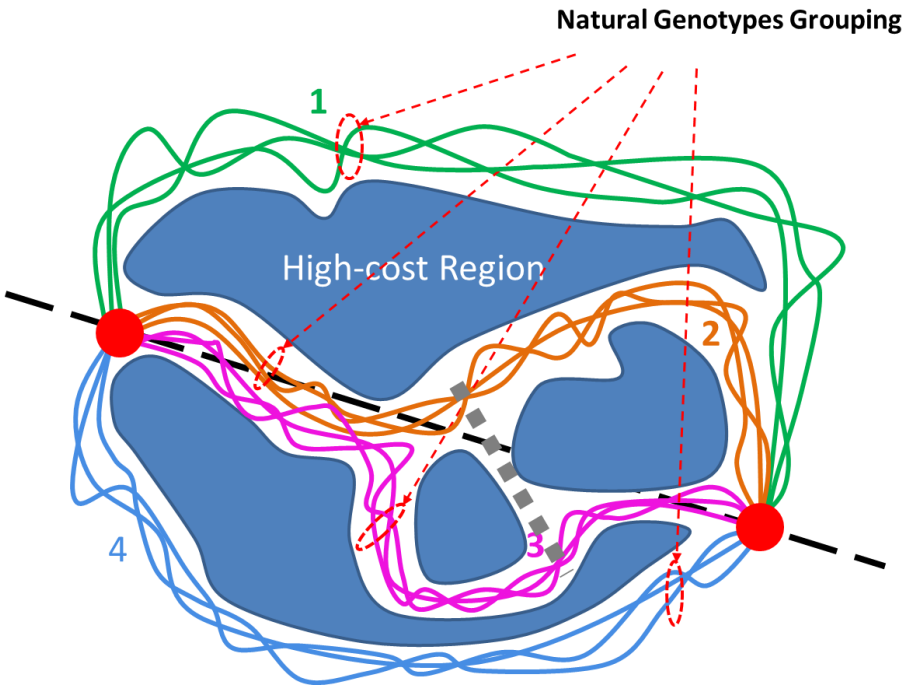


Figure 5. 9 Genotype Grouping

In this example, distance deviation from the straight-line connecting the two nodes is used as the metric for grouping and denoted as "level". Each level is assumed to be a genotype subgroup, with cross-over only within the level. In practice, the algorithm is implemented in two steps:

(a) Levelizing – in a general sense, our approach is an attempt to identify dominant near-optimal solutions that exists in different zones $Z_i, i = 1,2, \ldots n$, where n is the number of desired solutions, defined by the user. The method is not limited to three-dimensional space such as the current routing problem, but also more general optimization problems in higher dimensions. To find a dominant solution in a particular zone, starting genotypes associated with that zone must be chosen as input for the GA process. While there are arbitrarily many ways to group these genotypes, the LIGA algorithm groups genotypes according to a single metric that measures a genotype's deviation from a defined baseline. In this example, the most straightforward metric is the average deviation, or distance, of a genotype from the straight-line connecting the two nodes. Other than simple geometric distance, metrics as such cost weighted distance or more sophisticated fuzzy

classifiers can be used. Also, while the baseline in this example is chosen to be the straight-line connecting the nodes, more abstract baseline such as anisotropicity-weighted line may give different insights on designing the metric. Thus, we use the more generic term "level", instead of deviation, to describe the metric and refer the grouping process of "levelizing". For the current work, the LIGA algorithm uses the straight-line connection as baseline and average Euclidean distance as "level".

Starting with step (g) in the simple GA algorithm, the k lowest cost, randomly generated, paths are grouped according to the classification rules into levels. Three example paths in Figure 10 are classified based on their geometric distance from the straight-line connecting the start and end nodes into a bi-directional three level index scheme (i.e. +3, +2, +1, 0, -1, -2, -3), producing 7 levels.

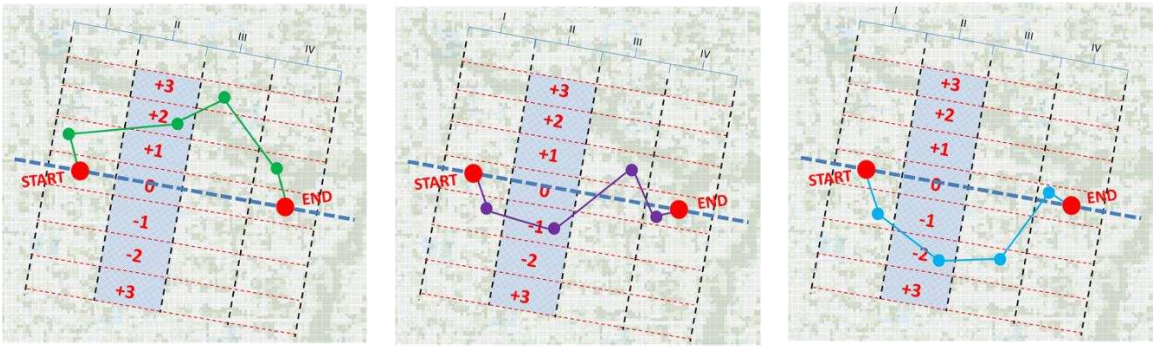


Figure 5. 10 Assigning paths to levels

In general, each level group will produce one or more coarse near-optimal solutions after applying the genetic algorithm discussed earlier, developing a family of near-optimal paths which are geographically distinct. It should be noted that, however, some zones can be so costly (e.g. very high anisotropicity or very low accessibility) that GA does not produce useful paths.

(b) Interpolation – Before proceeding for A* refinement, interpolation is used to further reduce the cost of the route. As pointed out in Section 3.2, n should not be too large when generating the random paths. However, this would limit the degrees of freedom when applying GA and hence also limit the solution optimality. Interpolation is introduced by making the mid-point of each segment a vital coordinate for the genetic algorithm (Figure 11). As such, if $n = 3$ when generating the random paths, 7 points will be available when applying the genetic algorithm.

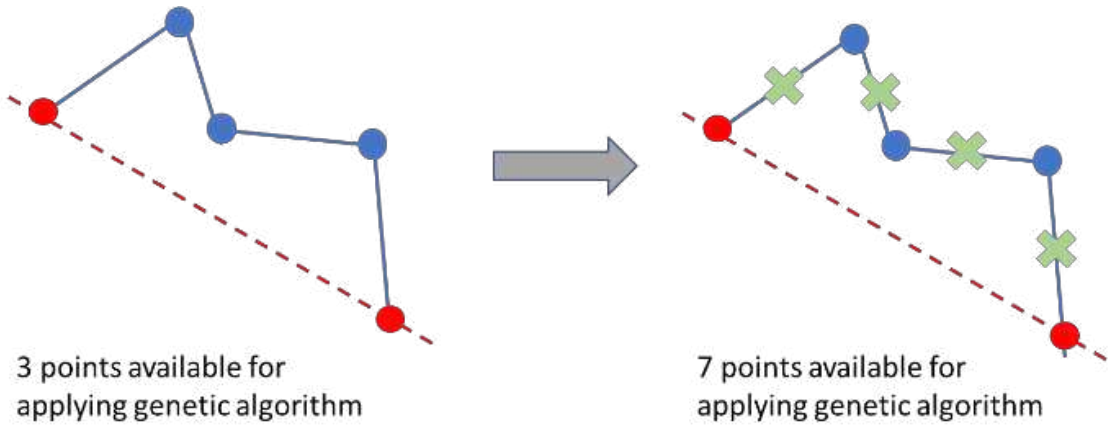
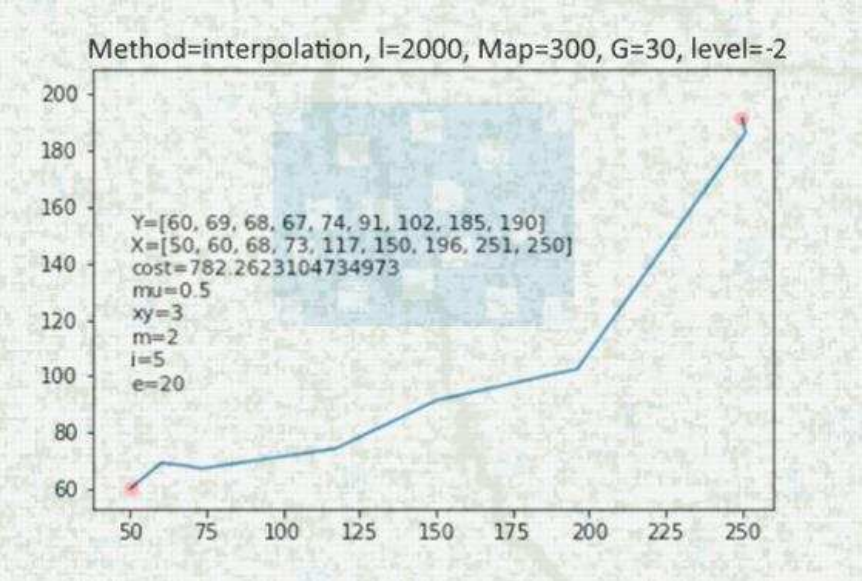


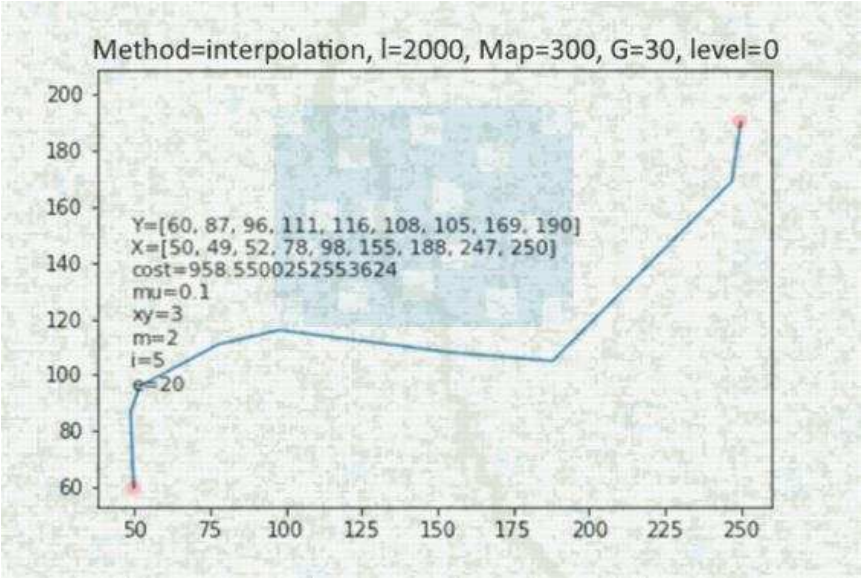
Figure 5. 11 Path interpolation

After levelization and interpolation, A*, MAA* or AMAA* refinement described earlier is separately applied to the genotypes in each level.

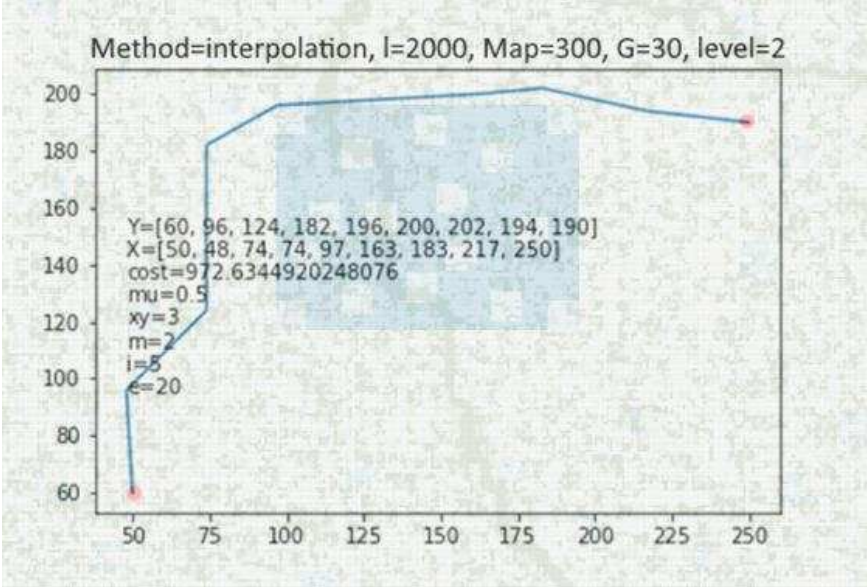
Referring to the map on Figure 4, Figure 12 displays example LIGA outputs for connecting (50,60) and (250, 190). The curves represent the physical shape of coarse near-optimal routes obtained for level = -2, 0, and 2, for routings using 30 generations. Figures 5A.1 and 5A.2 in Appendix 3A provide a more detailed view of the LIGA outputs associated with the figure, including results obtained with 15 trials and 100 generations under varying mutation rates, crossover criteria, etc. Figure 5A.1 records the actual coordinates of the path, and Figure 5A.2 shows the cost improvement against generation.



(a) Least-cost path for level = -2. This path lies near the globally optimal path (see Figure 7) computed by A*, and is the lowest cost across all levels.



(b) Least-cost path for level = 0. This level is grouped near the straight-line path between the two points. Since this group passes directly through the high-cost region at the center of the map, the lowest-cost genotype routes outside the high cost region.



(c) Least-cost path for level = 2. This level is above the straight-line path between the two points. As with (b), the algorithm routes outside the high-cost region at the center of the map, above the high-cost region.

Figure 5. 12 Example GA outputs for three levels after levelization and interpolation

5.5 Combining LIGA and MAA* / AMAA*

Using the algorithm in [29] [30], A* or MAA* can be applied to each line segments in the coarse, near-optimal, routes obtained using LIGA. As illustrated in Figure 13, MAA* refinement is used to reduce the costs of the local optima in Figure 12. As noted earlier, the speed of A*-derivative algorithms is highly impacted by the size of the search space. Therefore, providing a series of shorter routing tasks to MAA*, where it will search only the neighborhood of the line segment reduces the detailed routing cost substantially. Note, however, the MAA* algorithm may go outside the rectangle defined by the end points of the straight-line segment.

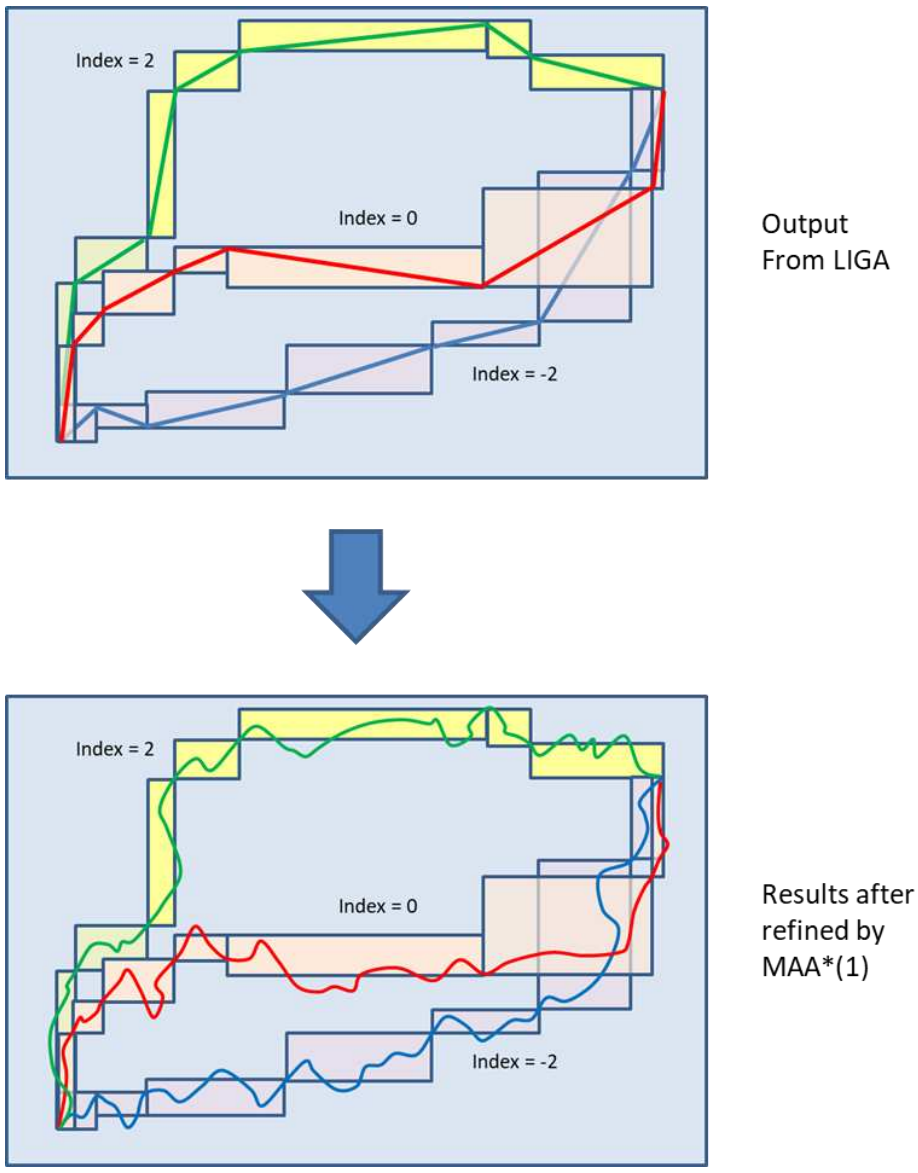


Figure 5. 13 LIGA-MAA* hybridization for family of paths

Using MAA*(1) to optimize the line segments, costs for the above three routes in end of Section 4 are reduced (a) from 782 to 410 when level = -2, (b) from 959 to 498 when level = 0, and (c) from 973 to 552 for level = 2. These cost levels are now more comparable to the global optimal 349. Furthermore, Table 5A.1 in Appendix 3A shows that the best route identified by this method has a cost of 380, which is even slightly better than the near-optimal cost 395 obtained by direct MAA* computation at m=1, but not as good as 364 obtained with $m = 0.5$ (Figure 4). Like any AI algorithm, however, one cannot guarantee which

method will give a better approximation of the global optimal because error introduced by an algorithm will depend on the anisotropicity of the search space, which is largely random in our example.

Two applications were evaluated to assess the performance and robustness of this hybrid method. The first case, which is a full illustration of the above computations, connects (50,60) and (250,190), and the second case connects (20,20) and (290,190). As can been seen from the computations, these two cases represent situations where (a) one near-optimal solution is likely dominating, and (b) multiple near-optimal solutions with similar costs co-exit. For both cases, GA are computed for 30 generations and 100 generations to examine the speed of convergence, using 15 trials with a varying mutation rates, crossover criteria, etc. MAA* with the same setting is used for all path refinement.

Case 1 – connecting (50,60) and (250,190)

An important question about this hybrid approach is correspondence: Do the few best-performing coarse routes generated by LIGA in the first step of the algorithm correspond to the best near-optimal routes after MAA* refinement? Since the highly anisotropic search space prevents correspondence in all cases, the practical question is whether applying detailed A*/MAA* routing to a small set of routes from LIGA will produce a reasonable, smaller, set of near-optimal routes.

Mathematically, consider the set of all LIGA results,

$$\Lambda = \{\lambda_1, \lambda_2 \dots \lambda_M, \dots \lambda_{N-1}, \lambda_N\} \tag{5.2}$$

ordered by the cost. Each LIGA route can be refined using detailed routing to produce an identically sized set, $\Gamma = \{\gamma_1, \gamma_2 \dots \gamma_N\}$, which is not ordered, since the transformation $\lambda_i \to \gamma_i$ is not guaranteed to preserve order. We then sort Γ by ascending cost to produce an ordered set of the same size, $G = \{g_1, g_2 \dots g_L \dots g_{N-1}, g_N\}$, maintaining the correspondence $\Lambda \to G$. Obviously, it is not possible to determine in advance which λ will lead to which g prior to preforming MAA* computation. In other words,

the indices s and e in the relation $\lambda_s \rightarrow g_e$ are indeterministic without going through detailed routing. However, it is reasonable to assume a statistically positive correlation would exist in most cases despite of anisotropicity.

In practice, detailed routing is computationally expensive. We therefore chose a subset $\widehat{\Lambda}$ containing the M lowest cost routes in Λ, and apply detailed routing to this subset, producing the corresponding set of detailed routings, $\hat{\Gamma} = \{\gamma_1, \gamma_2 \dots \gamma_M\}$. The quality of M can be measured by considering what fraction of the lowest cost detailed routings, $\hat{G} = \{g_1, g_2 \dots g_L\}$ is contained in $\hat{\Gamma}$, where L is the desired number of alternative near-optimal paths. While $\hat{G}$ will, in practice, remain unknown unless one computes all elements in G, we can still develop a metric and verify it by computing all elements for an example in order to obtain insights about the correspondence.

$$Q(M) = \frac{\mathcal{N}(\hat{G} \cap \hat{\Gamma})}{L} \tag{5.3}$$

where $\mathcal{N}$is the counting operator that returns the number of elements in a set, and the intersection operator, $\cap$, maintains correspondence $\Lambda \rightarrow G$. In general, we want to select M such that Q is as close to unity as possible, without exceeding the computational budget of project.

Figure 14, which is obtained by executing 30 generation for the GA process and refining by MAA*(1), provides some insights. With 15 LIGA routes for each of the seven levels, plus 15 routes generated using standard GA that includes top k genotypes without distinguish their level (for comparison purposes, indicated as 'level=full'), 120 coarse near-optimal routes are generated. Since our target is finding useful alternative routes, we only need to capture a reasonable number of "good" routes. From the figure, refining the best 25-30% of the 120 coarse routes would very likely serve the purpose. As an alternative view, Figure 15 depicts the correlation between costs of coarse and refined routes.

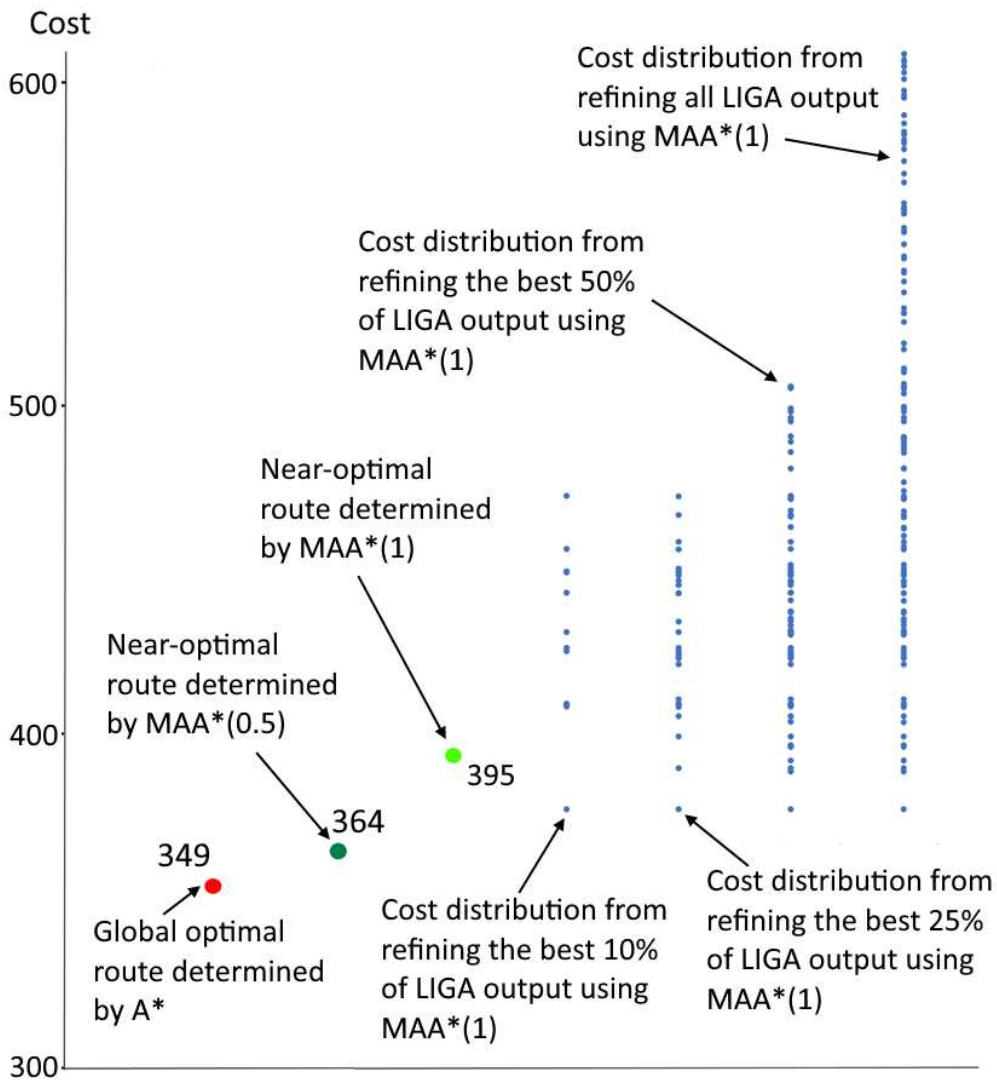


Figure 5. 14 Correspondence between LIGA output and refined routes, connection (50,60) – (250,190)

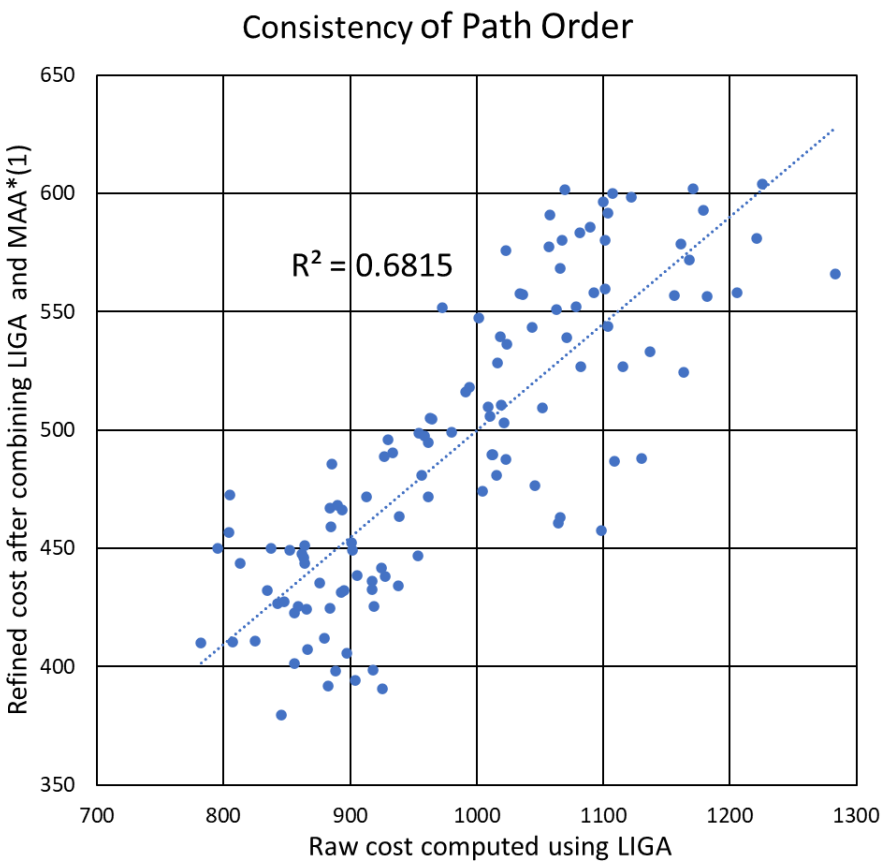


Figure 5. 15 Correlation between coarse and refined routes

It should be noted that while refining 25-30% of the LIGA outputs will likely capture enough lower cost routes, some difficult zones (i.e. highly anisotropic areas, very inaccessible areas) may not be covered in the final refined outputs. These zones are simply very costly to route, and even the refined routes in these zones are likely to cost more than other candidates. The user could, however, choose to retain a few routes from these zones, as worst-case contingency routes, in the knowledge that one or more other zones are always lower cost than this zone.

As an illustration, Table 1 and Figure 16 is an example selection of alternative routes for connecting (50,60)-(250,190). As before, the number of generations (G) for GA is chosen to be 30. Note that since all refined near-optimal routes have been computed in this case (Figure 14 and Table 5B.1, Appendix 3B), we have chosen the best refined near-optimal route for each zone in this example for demonstration. Based on

the correspondence consideration discussed above, the user can, in practice, choose to only compute the refinements of the best 25-30% the LIGA outputs and use this smaller set of results to construct the multi-path solution. Later examples adopt this approach. This example clearly illustrates that two geographically distant routes may have similar costs. Project personnel therefore retain the option to significantly modify system routing, knowing that routes north or south of the high-cost region have similar estimated costs.

Table 5. 1 Computation results using hybrid method, G = 30, connection (50,60)-(250,190)

Method	Path Cost
MAA* with m=1	394.92
Hybrid	
No. of generation in LIGA/GA= 30	
level = -2	379.61
level = -1	391.77
level = 1	475.55
level = 2	528.47
traditional GA (level = full)	410.55
Straight-line	2620.03
Global Minimum	348.66

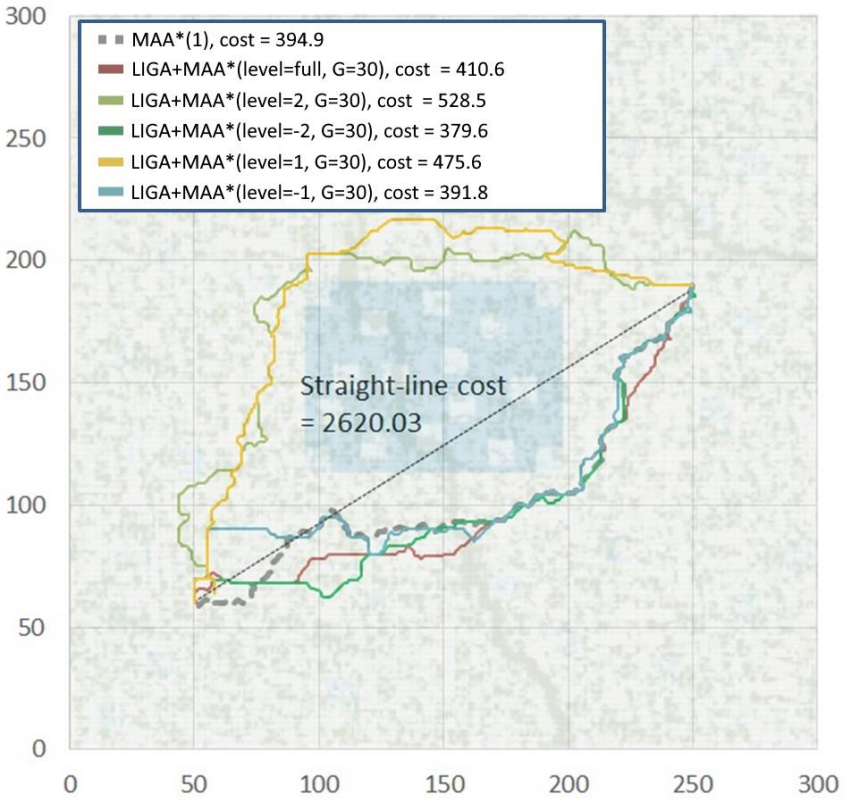


Figure 5. 16 Actual path shapes using hybrid method, G = 30, connection (50,60)-(250,190)

It is well known that the choice of number of generations in GA will affect optimality, as it takes successive steps for the genotypes to converge to an optimal solution [63]. In our case, it is expected that the GA process for genotypes at the same level will converge faster than a GA process that includes top k genotypes without distinguishing their level (denote as "level=full"). As can be seen from Table 2 and Figure 17, the refined route cost for level = full (i.e. traditional GA) has been improved by 6.3% from 410.6 to 384.9 when G has been increased from 30 to 100. For level = -2 and 2, there is no improvement as G increases. While these observations are consistent with the expectation, it is important to realize that such heuristics cannot be formally proved. Furthermore, solutions from GA are non-unique so comparing only the best solution for each level is not strictly appropriate, but it does offer insight into choosing G.

Table 5. 2 Computation results using hybrid method, G = 100, connection (50,60)-(250,190)

Method	Path Cost
Accelerated A* with m=1	394.92
Hybrid	
No. of generation in LIGA/GA = 100	
level = -3	406.68
level = -2	383.01
level =3	577.06
level = 2	540.75
traditional GA (index = full)	384.87
Straight-line	2620.03
Global Minimum	348.66

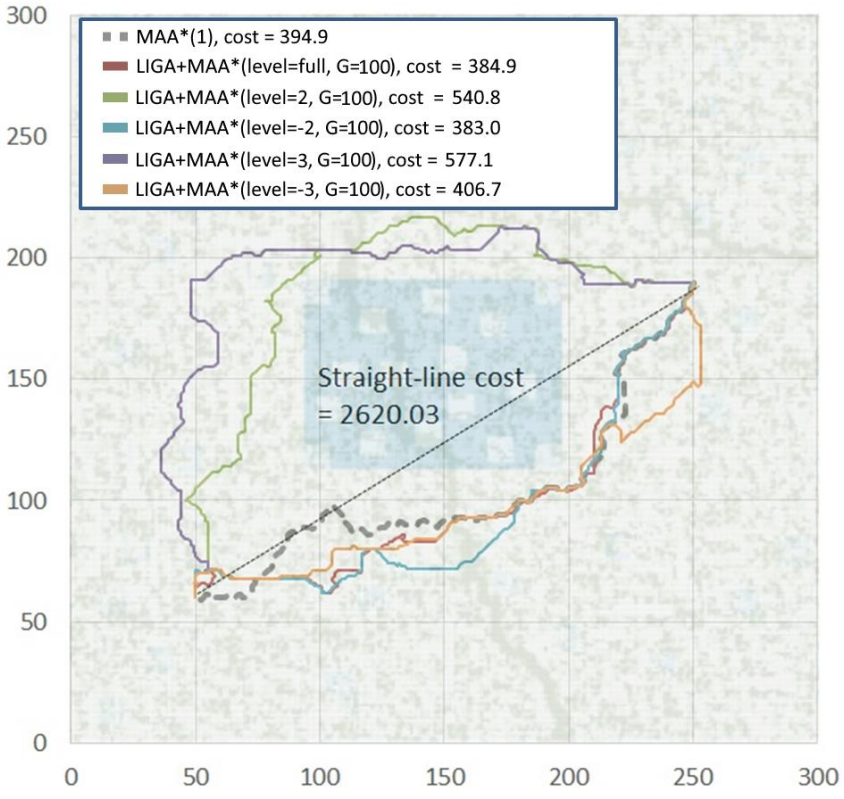


Figure 5. 17 Actual path shapes using hybrid method, G = 100, connection (50,60)-(250,190)

Case 2 – connecting (20,20) and (290,290)

This connection is significantly longer than the previous one. As demonstrated in [29] [30] [31], A* convergence of this connection is extremely slow since time complexity grows exponentially with path length. The much faster MAA* [29] [30] significantly reduces the computation time, but still took 54 hours (typical i7 personal computer running windows) to find a near-optimal route due to long length and high anisotropicity of the search space. An advantage of this hybrid method is faster computation for the shorter segments generated by LIGA. However, generating the candidates for MAA* refinement take time and, depending on the user's choice, a number of candidates have to be computed. It is not feasible to directly compare the computational efficiency for finding one near-optimal route using this hybrid method to the efficiency of direct MAA* calculation. As a rough indicator, generating 120 coarse routes (i.e. including generating genotypes, sorting, levelizing and interpolation) to connect (20,20) and (290,290) using LIGA took about 70 hours based on 30 generations GA, or about 180 hours if 100 generations GA is chosen. Time for refining a coarse route using MAA* varies significantly depending on length of segments and anisotropicity of the region these segments pass through. It can range from less than an hour to a few hours. If we assume the average computation time for refining one coarse route using MAA*(1) is 3.5 hours, generating one connection with this hybrid approach requires about 4.1 hours (for 30 generations). This is short compared to the 54 hours for a plain MAA*(1) computation, but it may not be a fair indicator because we only need a few alternative routes in practice. If the user decides to process 25% of the LIGA output, or 30 coarse routes, the total computation time of the whole hybrid algorithm will be approximately 175 hours. Assuming half of the refined outputs (i.e. 15 routes) have satisfactorily low cost for acting as alternative routes (see earlier example in Figure 16), the average computation time for each refined route is about 11.7 hours. While approximate, this time estimation provides insights on how the hybrid method compare to MAA* in terms of computation efficiency and optimality, and note, again, that the classic A* algorithm exceeded available computation time to route between these two points.

Table 3 and Figure 18 summarize the computation results with number of generations G = 30.

Table 5. 3 Computation results using hybrid method, G = 30, connection (20,20)-(290,290)

Method	Path Cost
Accelerated A*with m=1	633.99
Hybrid	
No. of generation in LIGA/GA = 30	
level = 3	771.62
level = 1	639.69
level = -2	655.39
level = -3	687.04
traditional GA (index = full)	651.66
Straight-line	3396.94
Accelerated A* with m=0.5	614.58

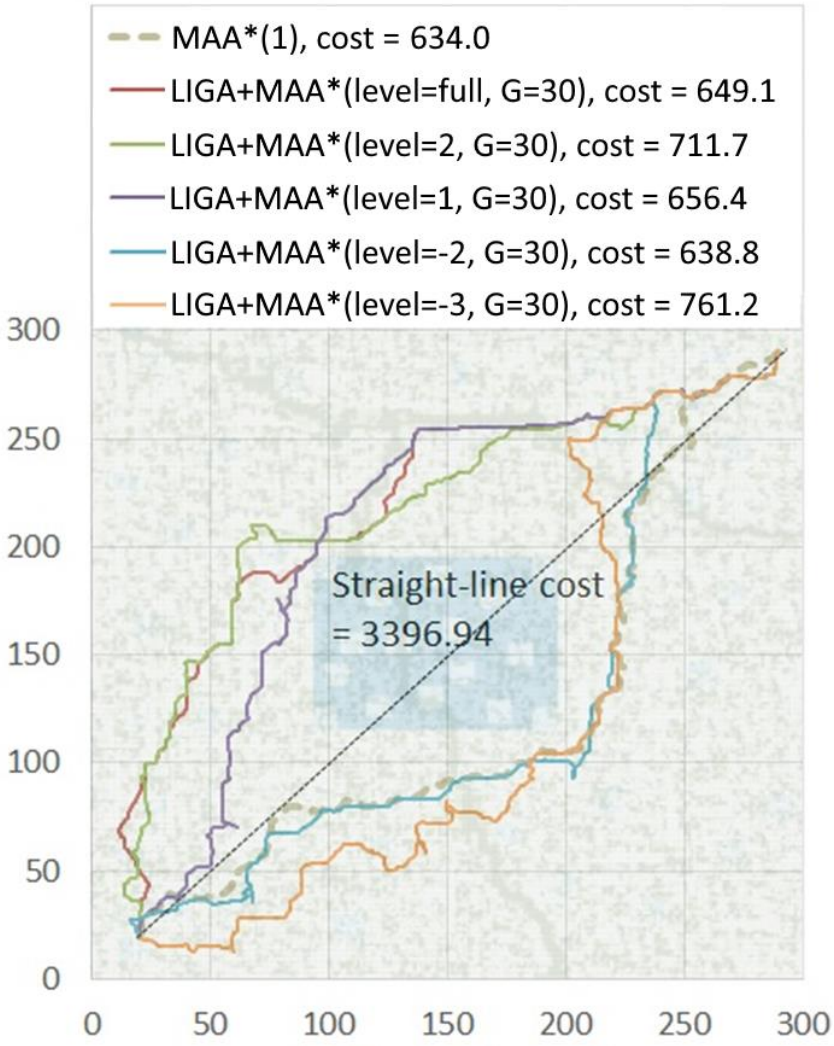


Figure 5. 18 Actual path shapes using hybrid method, G = 30, connection (20,20)-(290,290)

For this case, the most important observation is that the two lowest-cost paths identified by the hybrid method (i.e. index = 1 and index = full) are on the opposite side of the restricted area from the near-optimal path identified by MAA*(1) algorithm. On the other hand, the refined route with level = -2 is on the same side of the MAA*(1) route, though they are slightly different in shape. Costs of these three routes identified by hybrid method have very similar costs compared to the near-optimal path obtained by plain MAA*(1) even though they are in different zones of the map. The multi-path solution is thus highly desired as it

provides flexibility for network design, and the ability to change a route if an implementation problem occurs.

Increasing number of GA generation does not lead to remarkable changes. Results are provided in Appendix 3C.

After obtaining the family of near-optimal routes, minimum spanning tree (MST) method can be used to design the required network. Procedures are outlined in [30]. If cost of any path is changed due to unexpected conditions, MST would, based on total network cost, decide to replace it with another near-optimal route in the same family (i.e. an equivalent near-optimal network), or change the network topology (i.e. an alternative near-optimal network, as shown in Figure 3).

5.6 Discussion and Conclusion

Achieving SDG7 by 2030 is, as pointed out by IEA, increasingly challenging in recent years [5]. Electrification for the remaining population in difficult-to-reach locations could be very costly due to inevitable uncertainties in system planning and inefficient utilization of energy resources [22] [23]. To address these limitations, we have proposed a networked electrification approach that optimally connects users and centralized resources together using MAA* and AMAA* algorithms [29] [30] [31]. This study is an extension of the previous works. LIGA has been developed and applied in conjunction with MAA* / AMAA* to provide flexibility for practical implementation.

Based on the observations from the two application examples, the hybrid approach of combining LIGA and MAA* can offer a useful multi-path solution in both cases. In the family of near-optimal paths, the best member has a cost typically comparable to MAA*(1) path cost. Other members may have similar or higher costs depending on the actual geographical structure.

Like most meta-heuristic algorithms, solutions obtained by GA / LIGA are not unique. In addition, parameters for crossover and mutation will also affect the results. Therefore, some fine-tuning exercises are

desirable. For the purpose of this research, however, it is sufficient to demonstrate the viability of finding multi-path solution by combining LIGA and MAA* algorithm.

There are two areas for further investigations: (a) the cost correspondence between coarse near-optimal routes computed by LIGA and the refined near-optimal routes after MAA* refinement, as discussed in Section 5. Since one do not know exactly what will be the refined cost of a coarse route before actual computation, the current approach is to use the best 25-30% of the LIGA outputs for refinement and assume enough low-cost refined routes can be captured. While correlation analysis shows that this approach is reasonable, it is desirable if one can reliably use fewer LIGA outputs to obtain enough low-cost refined routes. It has been observed that those fast-converging LIGA coarse routes (Table 5A.2, Appendix 3A) tends to have better cost correspondence after MAA* refinement. This is probably due to lower anisotropicity along the line segments of these routes. Thus, in addition to cost of the coarse route, its speed of convergence could possibly be another selection criterion. However, more analysis will be required to validate this observation; (b) the level metric in this study is a simple distance measurement and the grouping is discrete. Cross-over between groups at different levels are not considered in order to speed up convergence. As pointed out in Section 4, crossing-over genotypes from some groups may have a better chance to give useful results than others. A more sophisticated metric (e.g. a fuzzy measurement) may help to identify more useful routes while maintaining speed of convergence.

CHAPTER 6 – ENHANCING NETWORKED RURAL ELECTRIFICATION USING MODIFIED FUZZY INFERENCE METHOD [4]

6.1 Flexible Optimal Network Design Using MST

After identifying all necessary near-optimal routes using MAA* / AMAA* and obtaining viable alternative routes using LIGA together with MAA*, optimal network can be constructed using minimum spanning tree (MST) or Capacitated Minimum Spanning Tree (CMST) algorithm [27]. A program has been written using standard Python 3 modules. The designed network is shown in Figures 6.1 and 6.2.

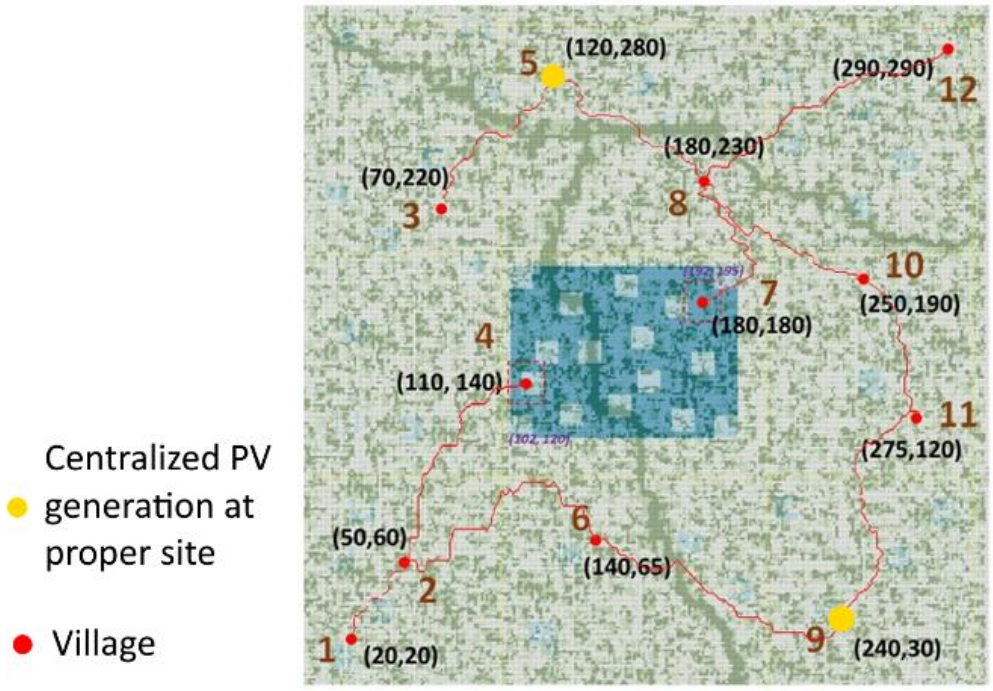


Figure 6. 1 Example optimal network for rural electrification

[4] Part of this chapter has been submitted to 2022 IEEE Global Humanitarian Technology Conference (GHTC) for publishing as a conference paper. Currently under review [33].

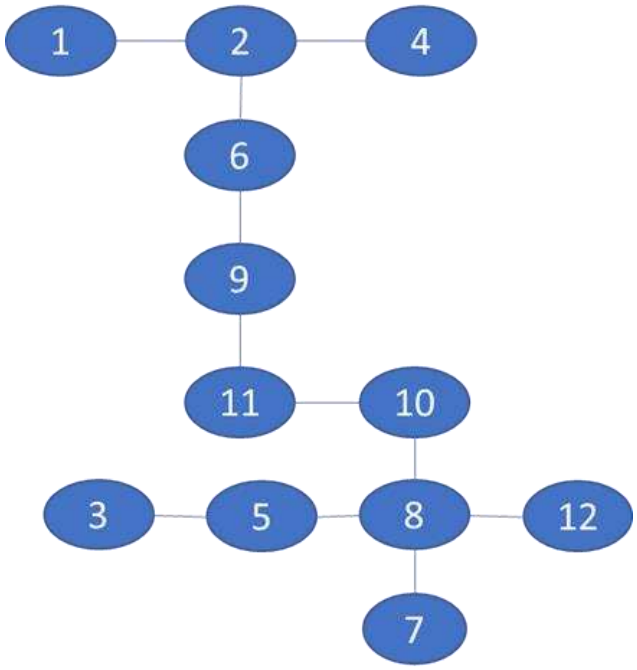


Figure 6. 2 Simplified network representation

Assuming a new obstacle on route 8-10 has been discovered (Figure 6.3) and the route cost is now increased from 130.4 to 310.5. The MST program will replace the original cost of route 8-10 with the cost of its best alternative (generated by LIGA and MAA*) and, based on total network cost, decide whether to simply replace the route or change the network topology. For example, if the best alternative for the original route 8-10 costs 185.6, the program will just replace the original route and keep the network topology unchanged because no other topology will give a lower network cost. In this case, the new network cost will become 2255. However, if the cost of the best alternative is 290.4, the program will change the topology (Figure 6.4) as the new design will give a lower network cost (total cost =2314) than direct route 8-10 replacement (total cost =2360).

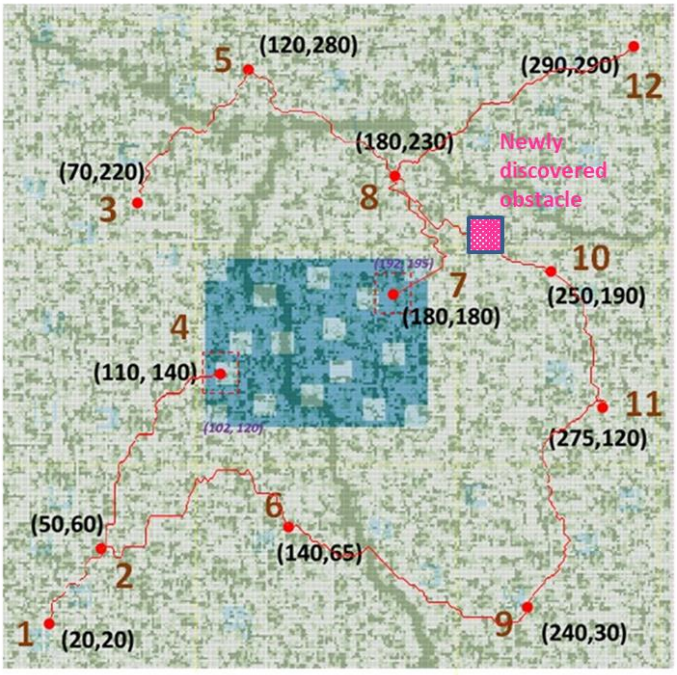


Figure 6. 3 Increasing route cost due to unexpected situation

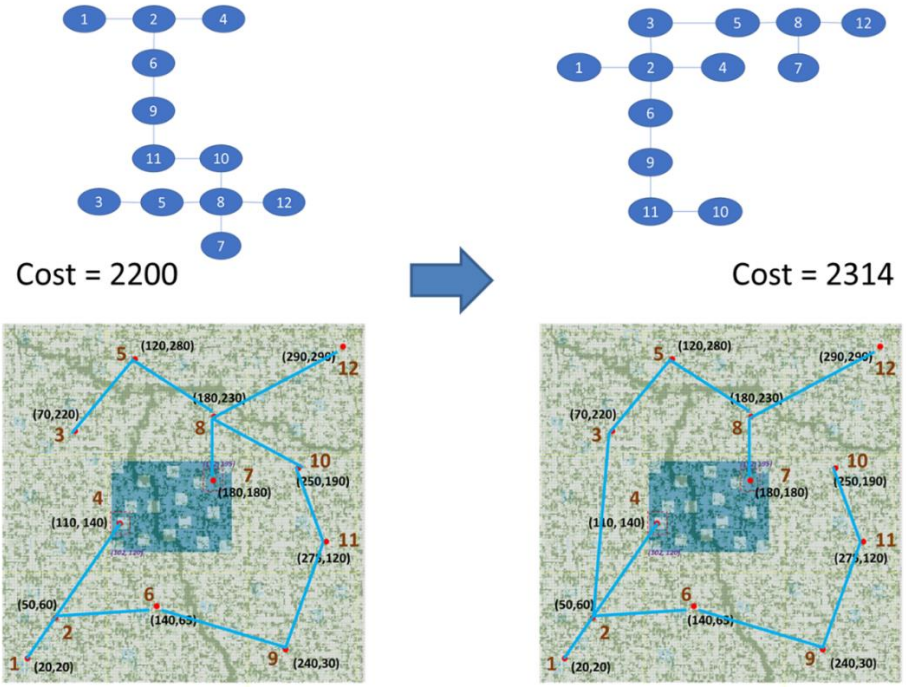


Figure 6. 4 Change topology based on total network cost

While MST and CMST can properly decide the optimal topology, it does not consider the opportunity of further optimization via reducing redundancy by joining paths. For example, routes 8-7 and 8-10 in Figure 6.5 (enlarged from Figure 6.1) run close to each other, and could be joined somewhere near the indicated ellipses to reduce distribution material and construction works.

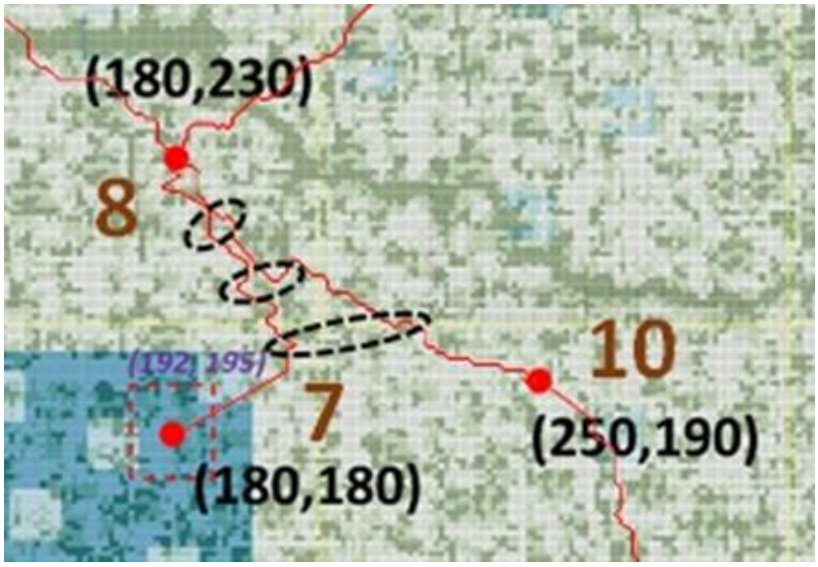


Figure 6. 5 Opportunities for further cost reduction

At the first glance, it appears that this further optimization could be done intuitively. One algorithm might simply examine the distance between two potential connection points on the paths, and compare it to the length of cable that could be saved. If the saving is significant, the connection is a reasonable one. This intuitive approach may work quite well for simple, isotropic search space because distance – i.e. the intuitive 'feel' – and actual cost for the changed routing are straightforwardly related. However, in rugged rural areas considered here, routing costs are highly anisotropic in nature: Physically proximate locations may be economically distance, due to terrain, rivers, exclusion areas, or other causes. Therefore, a more robust algorithm is needed.

For this analysis, cable capacity and similar engineering questions are not directly considered, as several alternatives, once identified, can be analyzed for total costs. This initiate analysis focusses on the required core methodology for selecting paths in challenging topological / routing conditions.

6.2 Fuzzy Inference Method for Further Optimization

6.2.1 The Question

As shown in Figure 6.5, many intuitive connections appear reasonable. In general, any one of a large number of connections could be selected, routed with a selected routing algorithm (e.g. MAA*) and compared to all other possible selection. In practice, this is computationally challenging. To speed computation, we seek a method which will down-select possible points pairs to those most promising for costly detailed routing.

To feasibly determine a 'good' but not necessarily the best connection, the challenge is to identify two nodes, one on route 8-7 and another on route 8-10, that will likely lead to cost reduction. For an isotropic space, cost is approximately proportional to the distance between the nodes. In such case, the criteria for selection are straightforward: (a) the nodes should be close to each other so required cable would be short; and (b) the nodes should be far away from end nodes (i.e. the villages) so more cable could be saved. If these two parameters were sufficient, fuzzy logic could be easily applied. Unfortunately, the search space is highly anisotropic: In most cases, cost is not proportional to distance. Significant topographical variation will affect cable length and other obstacles may drive up construction cost. The resulting shorter 2-dimensional routes may well be higher cost than the routes before interconnection. Furthermore, accessibility of each node can be different in this anisotropic space, which further complicates the optimization. In [29] [30], we modelled the cost as:

$$C_{i+1} = [(z_{i+1} - z_i)^2 + (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{\frac{1}{2}} + w_1|z_{i+1} - z_i| + w_2 a_{i+1} \quad (6.1)$$

where $x_i, y_i, z_i,$ are discretized 2D coordinates and elevation of the i^{th}point in the path and provide a surrogate for overall cable length; a_i is the accessibility of the i^{th} point and captures the difficulty of routing distribution through the i^{th} location, and w_1, w_2 are weighting coefficients; see the indicated references for additional description.

6.2.2 Monotonicity in Fuzzy Logic

While fuzzy inference seems to be a suitable approach for this application, applying standard fuzzy logic methodology to the problem is not straightforward.. To develop manageable number of implementable decision rules, variables usually need to be monotonic. The classic air conditioner control problem, for example, uses monotonic variables such as temperature and humidity as input [67] [68]. During fuzzification, monotonicity can be assumed: if an observer indicates 30° C is "hot", then the same observer must regard 40° C as "hot" or "very hot". If, instead, the observer categorized 40° C as "warm" or "cold", the complexity of the controller grows exponentially. Without monotonicity, decision rules become both complicated and computationally intractable [69] [70]. In our application, however, such monotonicity cannot be guaranteed; i.e. shorter distance does not imply lower cost. Hence, no simple membership function can be directly used for fuzzification. To properly apply fuzzy logic, constructed variables that maintain (or at least approximately maintain) monotonicity must be derived from physical variables (Figure 6.6).

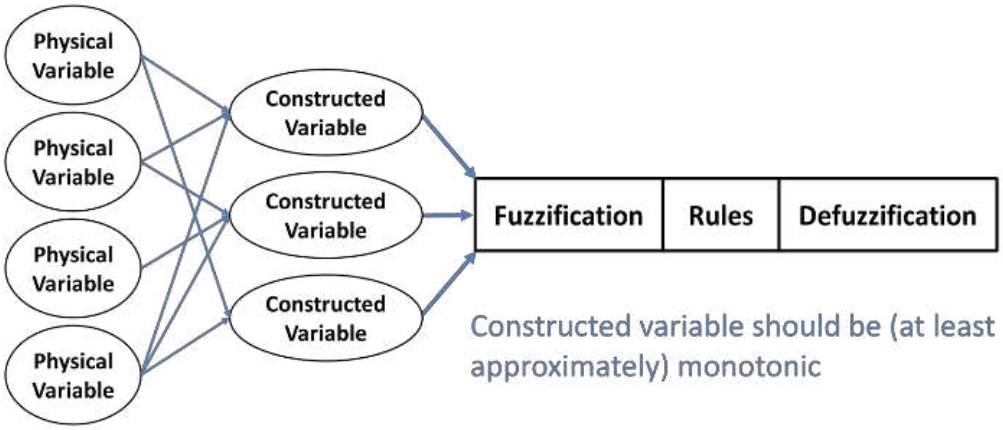


Figure 6. 6 Constructed monotonic variables

6.2.3 Variable Construction

While this transform is conceptually simple, it is either complex or computationally intractable to construct a variable that maintains monotonicity without incurring the cost of detailed routing. To address this conflict, we propose a variable that has, at least partially, captured the anisotropicity of possible routes. Thus, the following three variables are constructed (Figure 6.7):

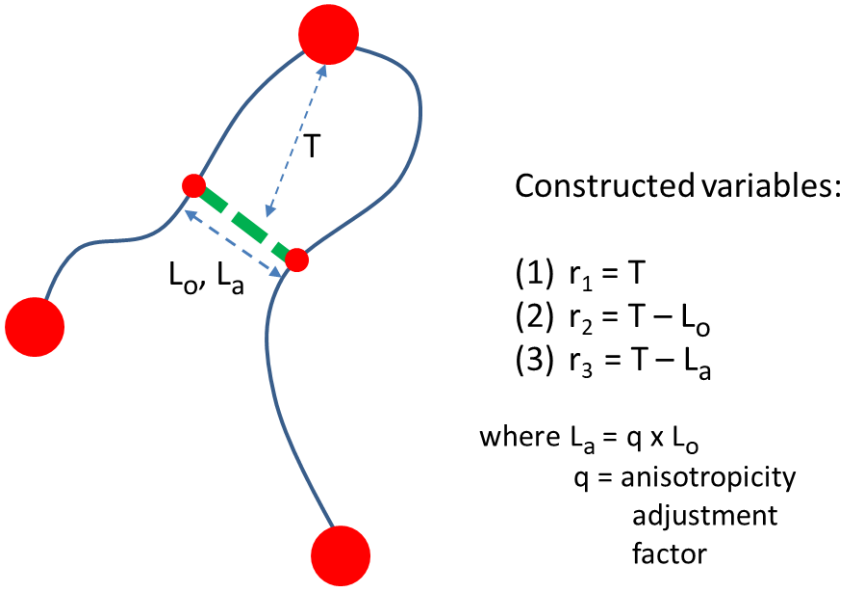


Figure 6. 7 Practical construction of conceptual variables

(i) r_1 : R^2 distance T from the mid-point of connection to the common end node. If T is small, the overall distance reduced will be limited, and therefore cost reduction will be limited. While there are exceptions in extreme cases, a larger r_1 is generally preferred;

(ii) r_2 : heuristic saving potential, i.e. the difference between T and the R^2 length of the straight-line connection (L_o). Large r_2 is preferred;

(iii) r_3 : anisotropicity-adjusted saving potential, i.e. the difference between T and an anisotropicity-modulated length of the straight-line connection. When anisotropicity of an area is high, the actual cost of constructing the straight-line connection will be likely high. Again, while exceptions exist, since cost depends on actual topographical variation, this metric is generally applicable. Thus, one can envision routing through an anisotropic area as connecting via a longer path. The anisotropicity-adjusted length L_a. is obtained by multiplying L_o. with an anisotropicity adjustment factor q (Figure 6.7), and the adjusted saving r_3 is the difference between T and L_a. The simplest choice for q is the ratio between cost for connecting the two nodes via straight-line (i.e. the sum of the cost of each step moving along the line, which

is much easier to compute than the optimal route) and the R^2 distance between the nodes. When anisotropicity is high, the cost of passing through the straight line is expected to be high and therefore L_a is becoming larger and r_3 smaller. Large r_3 is preferred.

Note that r_2 and r_3 are used together because adjustment made in r_3 may not be proper in some cases due to unusual topographical or accessibility variations. Retaining r_2 as an input to fuzzifier is helpful to enhance the validity of decision.

6.2.4 Implementation

To implement the scheme, potential nodes for connections are first identified and their corresponding variables (i.e. r_1, r_2 and r_3) are computed. This step is computationally simple but will typically generate thousands of connections. An initial screening based on the variable values will be done and a few hundred of reasonable connections will be selected for fuzzification. Then, based on defuzzification score, tens of connections will be chosen for detailed MAA* computation and determine potential for further optimization.

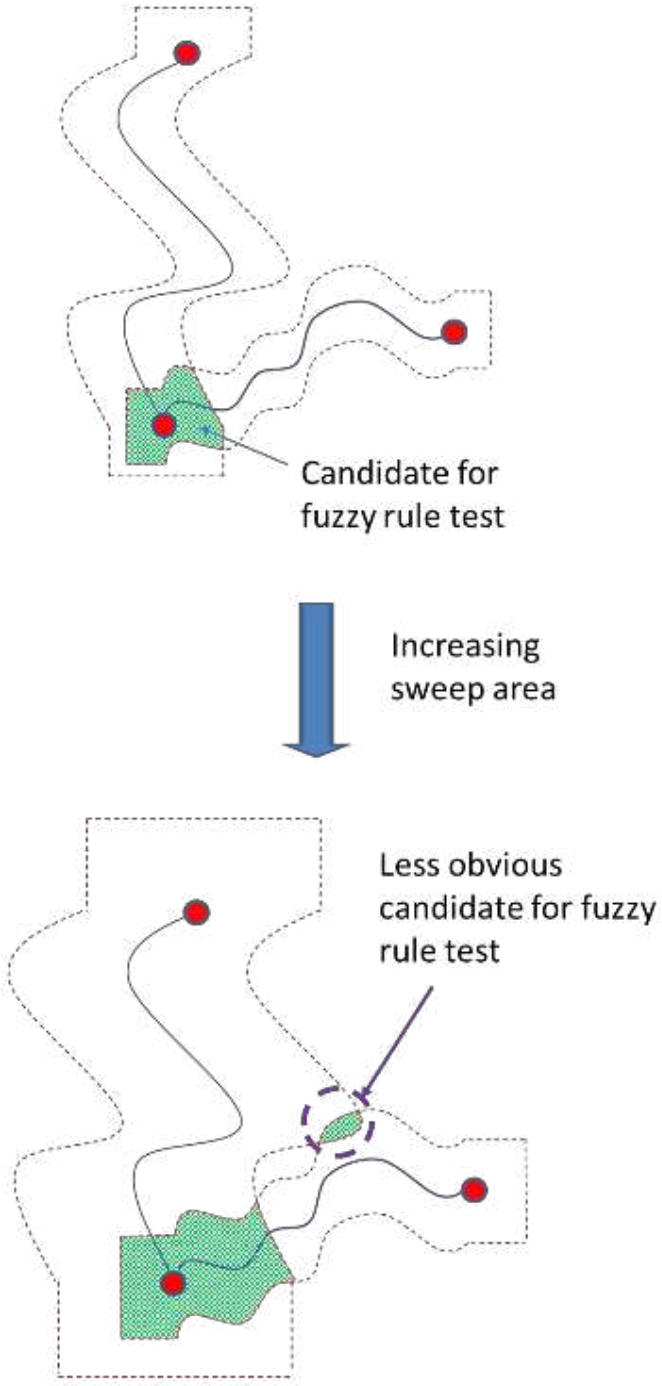


Figure 6. 8 Determining nodes for connection

Potential nodes are determined by expanding the routes vertically or horizontally and then look for overlapping points (Figure 6.8). For each overlapping point, the two corresponding nodes on the routes are

calculated based on minimum distance from the overlapping point. With the nodes determined, variables r_1, r_2 and r_3 for each possible connection are also determined. By examining r_2 and r_3 , connections with low saving potential are eliminated and the remaining will undergo fuzzy logic analysis. The membership functions need to be, in theory, adjusted according to the range of r_1, r_2 and r_3. However, since the scale of problem is relatively constant throughout Figure 1, one set of empirical function (Figure 6.9) is found applicable for many tested route pairs.

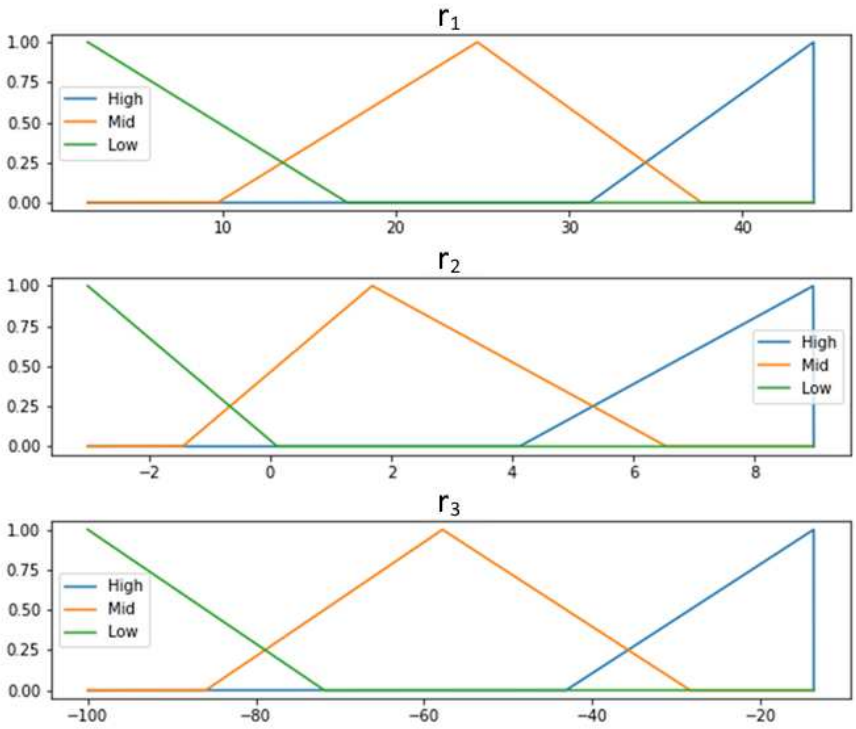


Figure 6. 9 Membership function for fuzzy classification

Scores for the selected connections are computed and defuzzied according to Figure 6.10. Optimization using MAA* is then performed.

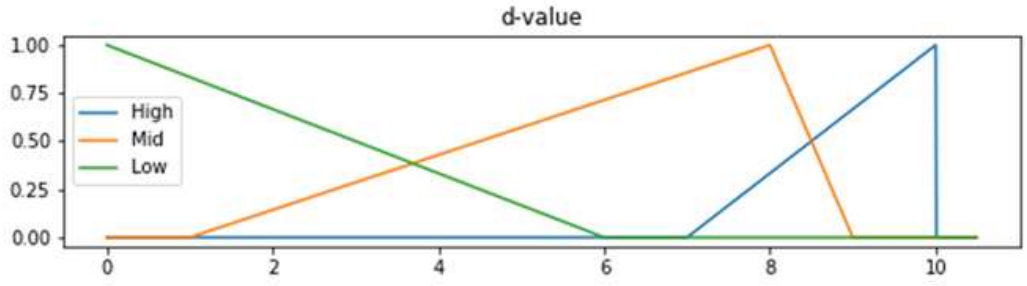


Figure 6. 10 Defuzzification rule

6.3 Example Analysis

A number of route pairs in Figure 1 have been analyzed. These examples represent different degree of difficulties in further optimization, and also provide insights on the relationship between defuzzification score and actual saving under different situations.

Example 1: Routes 8-7 and 8-10

Because of the proximity of these routes (see Figure 6.1 for actual path shapes), it is almost certain that optimization opportunities exist. However, as will be shown below, intuitive placement of connecting route may give compromised results and hence formal method is required.

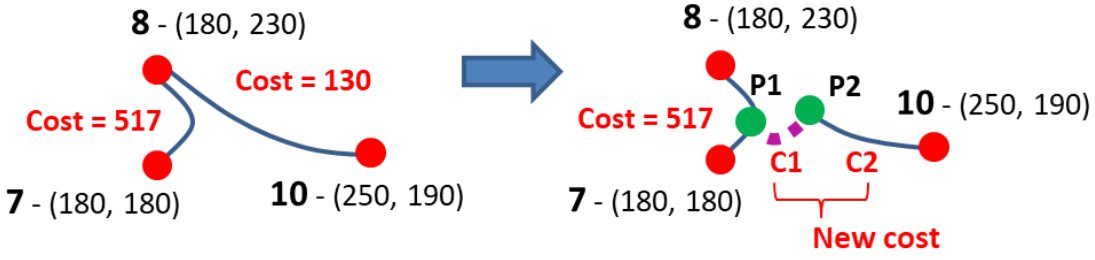


Figure 6. 11 Optimizing routes 8-7 and 8-10 using fuzzy inference

Following the steps outlined, 972 possible connections are generated. Based on r_1, r_2 and r_3, 81 out of these 972 connections are sorted out for fuzzy inference according to the estimated saving potential. For demonstration purpose, all 81 connections are analyzed using MAA*. Figure 6.12 illustrates the observed relation between defuzzification score and actual saving. In this example, saving refers to the cost reduction by replacing route 8-10 with route 8-P1-P2-10. Percentage saving is based on the original cost of 8-10, i.e. 130 as shown in Figure 6.11.

As shown in Figure 6.12, an approximately linear relation exists between score and saving, indicating that the constructed variables for fuzzy inference produce desired results. There are a small number of exceptions at the high score side – routes with good fuzzy logic scores that produce relatively low savings. However, this is well anticipated because of high anisotropicity.

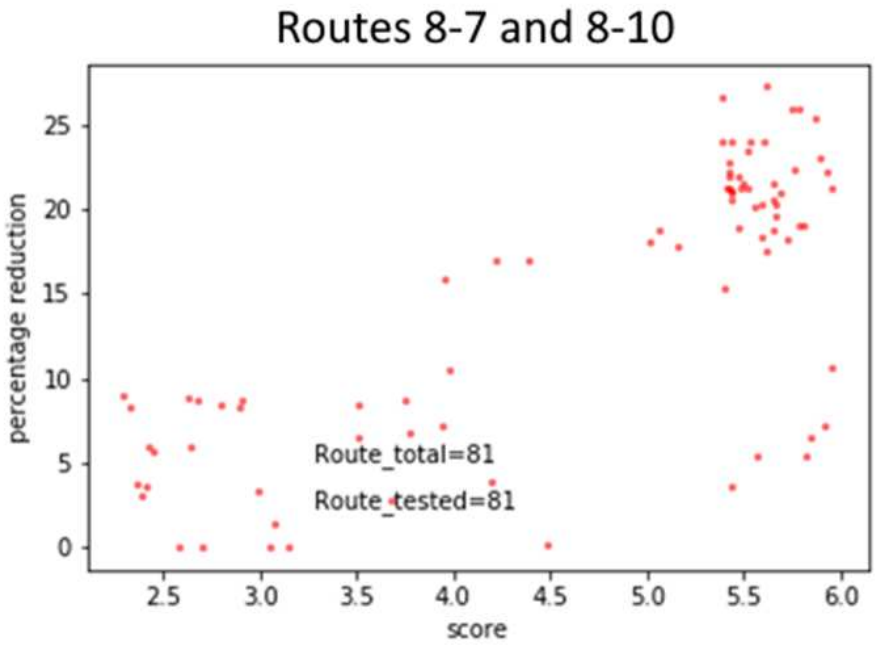


Figure 6. 12 Relation between defuzzification score and saving, routes 8-7 and 8-10

Given the approximate linear relation between score and relative cost reduction, it is not necessary to analyze all the 81 connections with MAA* in practice. Tables 6.1 and 6.2 compares 30 top score connections and 30 largest saving connections. Conducting MAA* computation for the top 30 score connections provides a sufficiently large number of good solutions for cost reduction, though, like any AI algorithm, there is no guarantee that the best solution can be captured.

In this example, the identified best connection nodes are P1 = (191,203) and P2 = (199,208). This solution has a fuzzy logic score of 5.6 and offers 27% cost reduction. A comparable solution is P1 = (190,203) and P2 = (210,206), which has a score of 5.8 and offers 26% cost reduction. If one examines Figure 6.5, these connections also appear consistent with intuitive choice, even though intuition would not necessarily pick those exact coordinates. However, with a geometrically and intuitively similar connection between P1 = (193,196) and P 2= (215,204), the score is 3.1 and provides only 1.4% cost reduction. This illustrates that a small change in selected node position can lead to large change in saving – an essential property of this anisotropic search space, and the key driver for use of a formal optimization method.

Table 6. 1 Top 30 connections with highest scores, routes 8-7 and 8-10

ID	P1		P2		score	C1	C2	New Cost	Saving %
	X	Y	X	Y					
1	193	196	199	208	5.96	32.3	83.9	116.2	10.6%
2	188	211	201	211	5.96	16.1	86.3	102.4	21.3%
3	191	203	198	208	5.93	11.1	90.0	101.0	22.3%
4	192	197	198	208	5.92	30.8	90.0	120.7	7.1%
5	191	203	204	211	5.89	18.3	81.8	100.1	23.0%
6	188	211	202	212	5.87	17.5	79.5	97.0	25.4%
7	193	196	198	208	5.85	31.5	90.0	121.5	6.5%
8	193	196	210	206	5.82	53.8	69.2	123.1	5.3%
9	191	199	198	208	5.81	15.3	90.0	105.3	19.0%
10	190	200	198	208	5.79	15.3	90.0	105.3	19.0%
11	191	203	210	206	5.78	27.0	69.2	96.2	26.0%
12	191	203	205	210	5.76	20.6	80.4	100.9	22.4%
13	191	203	211	206	5.75	28.0	68.2	96.2	26.0%
14	191	198	198	208	5.72	16.3	90.0	106.3	18.3%
15	188	211	193	213	5.69	5.8	96.9	102.8	21.0%
16	190	204	197	209	5.67	13.9	90.5	104.4	19.7%
17	188	211	192	214	5.67	5.2	98.3	103.6	20.3%
18	187	206	194	212	5.66	10.1	95.5	105.6	18.8%
19	188	210	193	213	5.65	6.2	96.9	103.2	20.6%
20	188	211	194	212	5.65	6.4	95.5	101.9	21.6%
21	191	203	199	208	5.62	10.7	83.9	94.5	27.3%
22	190	204	196	210	5.61	15.3	91.9	107.2	17.6%
23	191	203	206	209	5.60	21.1	77.6	98.8	24.0%
24	188	205	195	211	5.60	12.9	93.3	106.2	18.3%
25	188	209	193	213	5.59	6.7	96.9	103.6	20.3%
26	193	196	211	206	5.57	54.8	68.2	123.1	5.3%
27	191	203	197	209	5.55	13.3	90.5	103.9	20.1%
28	191	203	209	207	5.54	25.6	73.2	98.8	24.0%
29	191	202	198	208	5.52	12.3	90.0	102.3	21.3%
30	191	203	200	209	5.52	12.1	87.4	99.5	23.5%

Table 6. 2 Top 30 connections with largest savings, routes 8-7 and 8-10

ID	P1		P2		score	C1	C2	New Cost	Saving %
	X	Y	X	Y					
21	191	203	199	208	5.62	10.7	83.9	94.5	27.3%
47	191	203	202	212	5.38	15.9	79.5	95.4	26.6%
11	191	203	210	206	5.78	27.0	69.2	96.2	26.0%
13	191	203	211	206	5.75	28.0	68.2	96.2	26.0%
6	188	211	202	212	5.87	17.5	79.5	97.0	25.4%
23	191	203	206	209	5.60	21.1	77.6	98.8	24.0%
28	191	203	209	207	5.54	25.6	73.2	98.8	24.0%
36	191	203	209	208	5.44	24.6	74.2	98.8	24.0%
46	191	203	207	209	5.39	22.1	76.6	98.8	24.0%
30	191	203	200	209	5.52	12.1	87.4	99.5	23.5%
5	191	203	204	211	5.89	18.3	81.8	100.1	23.0%
43	191	203	201	210	5.42	14.3	86.0	100.3	22.8%
12	191	203	205	210	5.76	20.6	80.4	100.9	22.4%
3	191	203	198	208	5.93	11.1	90.0	101.0	22.3%
42	188	211	195	211	5.42	7.8	93.3	101.1	22.2%
33	188	210	196	210	5.48	9.7	91.9	101.5	21.9%
41	188	210	195	211	5.43	8.2	93.3	101.5	21.9%
20	188	211	194	212	5.65	6.4	95.5	101.9	21.6%
31	188	209	195	211	5.50	8.7	93.3	101.9	21.6%
29	191	202	198	208	5.52	12.3	90.0	102.3	21.3%
32	188	208	195	211	5.48	9.1	93.3	102.3	21.3%
40	188	210	194	212	5.43	6.8	95.5	102.3	21.3%
44	188	208	196	210	5.41	10.5	91.9	102.3	21.3%
2	188	211	201	211	5.96	16.1	86.3	102.4	21.3%
35	188	209	196	210	5.44	10.7	91.9	102.5	21.1%
15	188	211	193	213	5.69	5.8	96.9	102.8	21.0%
39	188	209	194	212	5.43	7.2	95.5	102.8	21.0%
19	188	210	193	213	5.65	6.2	96.9	103.2	20.6%
38	188	208	194	212	5.44	7.7	95.5	103.2	20.6%
17	188	211	192	214	5.67	5.2	98.3	103.6	20.3%

Example 2: Routes 2-4 and 2-6

Optimization opportunity is less obvious with this route pair (see Figure 6.1 for actual path shapes).

Figure 6. 13 Optimizing routes 2-4 and 2-6 using fuzzy inference

Similar to the previous example, 97 potential connections are identified. By processing the first 30 top score connections with MAA*, the top 10 opportunities for cost reduction are listed in Table 6.3.

Table 6. 3 Result of optimizing routes 2-4 and 2-6 based on top 30 score

P1	P2	C1	C2	New Cost	Reduction	%
(55,73)	(74,67)	21.5	95.4	116.9	34.1	22.6
(55,71)	(74,67)	21.5	95.4	116.9	34.1	22.6
(55,72)	(74,67)	21.9	95.4	117.3	33.7	22.3
(55,71)	(73,64)	25.7	98.8	124.6	26.4	17.5
(55,73)	(75,68)	22.1	104.4	126.5	24.5	16.2
(55,74)	(75,68)	22.5	104.4	126.9	24.1	16.0
(58,93)	(93,73)	48.5	82.2	130.7	20.3	13.5
(55,78)	(75,68)	29.0	104.4	133.4	17.6	11.7
(55,76)	(75,68)	29.2	104.4	133.6	17.4	11.5
(55,76)	(74,67)	38.3	95.4	133.7	17.3	11.5

Although it is unnecessary to process all 97 potential connections in order to find proper saving opportunities, all of them have actually been analyzed with MAA* so we can visualize the relation between score and saving for this route pair (Figure 6.14).

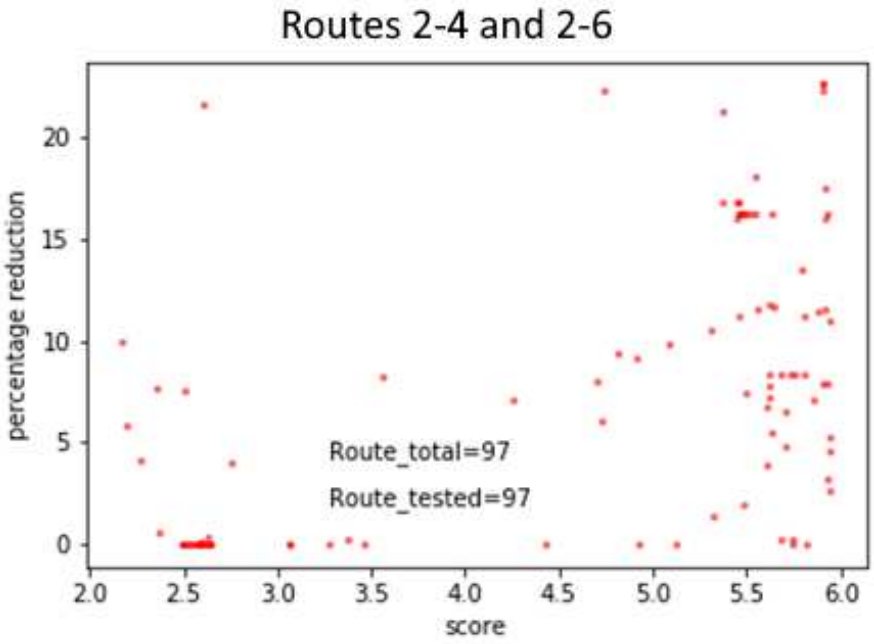


Figure 6. 14 Relation between defuzzification score and saving, routes 2-4 and 2-6

While a positive relation still exists, the correlation is not as strong as the previous example (see Figure 6.12). More exceptions (i.e. low score with high saving, or high score with low saving) are observed.

Recall that (a) only variable r_3 is used to account for anisotropicity, and (b) the method of accounting only uses the straight-line cost as an estimate, it is not surprising that some high score potential connections turn into low saving as the connection length is getting longer and more errors are being introduced by anisotropicity. For similar reason, a small number of low score potential connections turn into high saving because of inaccurate (over) estimate of anisotropicity in the region the routes pass through.

In this example, the method is still performing reasonably well because the routes are not physically very diverted. When all 97 potential connections are analyzed with MAA*, the top 10 cost reduction opportunities range from 16.8% to 22.6%. Referring to Table 6.3, the method has captured the top saving of 22.6%, and about half of the top savers. Thus, this approach is working well for optimizing the route pair. In the next example, a difficult pair will be examined.

Example 3: Routes 5-3 and 5-8

Route pair in this example is physically more divergent, with no intuitive solution for cost reduction. Thus, it is expected that the method will be less able to capture the top savers.

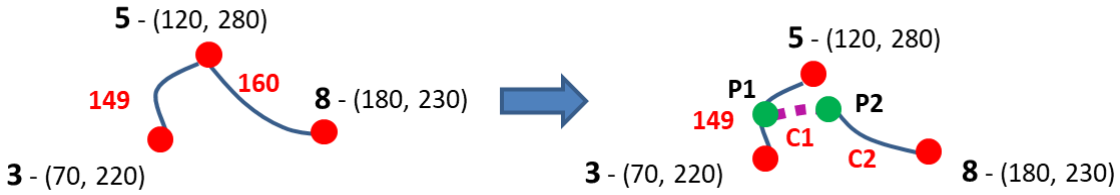


Figure 6. 15 Optimizing routes 5-3 and 5-8 using fuzzy inference – attempt 1

For this example, 87 potential connections are identified. By processing the first 30 top score connections with MAA*, the top 10 opportunities for cost reduction are listed in Table 6.4.

Table 6. 4 Result of optimizing routes 5-3 and 5-8 based on top 30 score

P1	P2	C1	C2	New Cost	Reduction	%
(102,255)	(129,263)	51.0	80.8	131.8	17.2	11.5
(100,251)	(129,263)	56.7	76.0	132.7	16.3	11.0
(108,262)	(129,263)	40.3	94.5	134.8	14.2	9.6
(109,263)	(128,265)	33.1	102.2	135.3	13.7	9.2
(108,262)	(128,265)	41.5	94.5	136.0	13.0	8.7
(103,256)	(129,263)	44.1	92.0	136.1	12.9	8.7
(110,263)	(128,265)	32.8	104.0	136.8	12.2	8.2
(110,263)	(130,262)	32.9	104.0	136.9	12.1	8.1
(110,263)	(129,263)	33.0	104.0	137.0	12.0	8.1
(103,257)	(128,265)	45.1	92.4	137.5	11.5	7.7

When all 87 potential connections are analyzed with MAA*, the top 10 cost reduction opportunities range from 10.7% to 11.7%. Referring to Table 6.4, the method has not captured the top saving opportunity,

and only captured two of the top 10 savers. Thus, the correlation between score and saving is getting even weaker (Figure 6.16). On the other hand, while the score is becoming less capable for predicting saving for this route pair, there are in fact very few connections that can offer cost reduction under this physical setting. This is analogical to the situation that the same fishing net catches less fish in an area where there are not many fish worth catching.

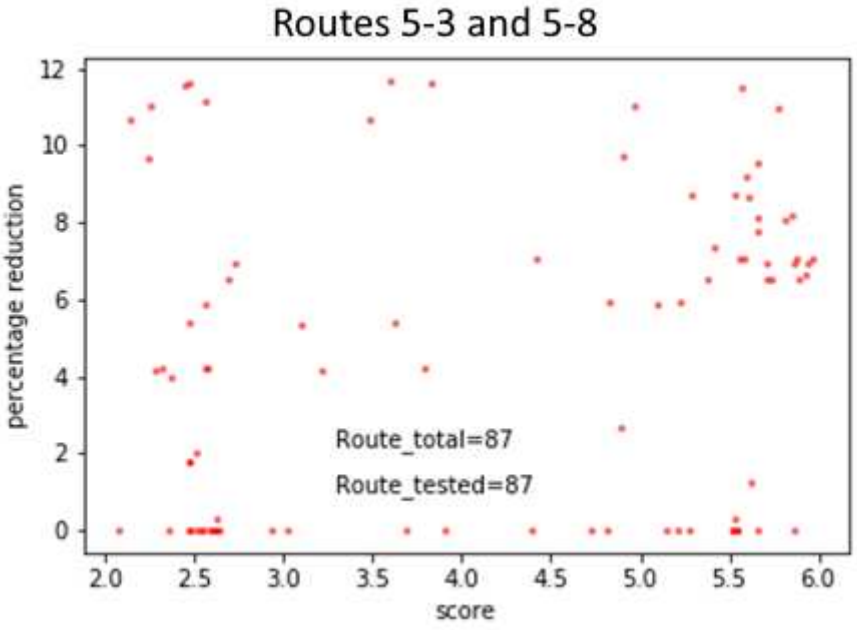


Figure 6. 16 Relation between defuzzification score and saving, routes 5-3 and 5-8

6.4 Limitations and Discussions

Because of monotonicity requirement on variables, standard fuzzy inference method cannot be directly applied to problem with a highly anisotropic search space. By constructing new conceptual variables, we have demonstrated the feasibility of applying fuzzy logic to this class of problems.

The approach illustrated here is applicable to a wide range of routing and selection problems in similarly anisotropic search spaces. At this stage of early investigation, however, the variables we have constructed

are simple and accounts only partially for the effects of anisotropicity. As demonstrated in Section 6.3, the method works well for typical route pairs, but becoming less predictive for some physical settings with less optimization opportunity. Whether this is critical depends on the user's goal. However, the method can still quite effectively indicate potential for optimization.

Several concepts could be applied to improve predictability. One approach would focus on improving the consistency of r_3 by using an estimate of anisotropicity that better matched the computed path between two locations. One possible approach is using the cost of a zip-zip path, instead of a straight-line, as the denominator of the anisotropicity adjustment factor q (Figure 6.7). This may provide slightly more accurate information about the anisotropicity of the region the route pass through. For more fundamental improvement, combining fuzzy inference with ordinal optimization [71] seems to be a possible direction. Ordinal optimization focuses on order rather than value and systematical softens optimization goal to make hard problem easier. While this is compatible to our fuzzy inference approach, extensive research work will be required for developing implementable methodologies.

CHAPTER 7 – CONCLUSIONS

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 in complex topographies is computationally prohibitive. When a complex topography is transformed into digital representation, search space for the optimization problem will become highly anisotropic. Classical search methods will be extremely inefficient under this situation, and most common acceleration methods will fail to function.

In this research, we have developed a comprehensive toolset for implementing this networked rural electrification model. The toolset consists of new computational methods and algorithms designed to overcome identified difficulties: MAA* and AMAA* greatly reduce computational complexity while maintaining good optimality for detailed routing; LIGA, in conjunction with A*, MAA*, or AMAA*, provides viable alternative plans to tackle unexpected route change problem right before or even during project implementation; and the modified fuzzy inference method allows further optimization of the network structure with minimal computation. It is hopeful that this development will play a role in accelerating SDG7 implementation.

7.1 Theoretical contributions

Effect of search space anisotropicity on optimization method is a challenging topic. There are different approaches for understanding this issue. For example, a theorist investigates methods for constructing precise heuristic function in A* algorithm in order to improve search efficiency. This approach is rigorous but results in a heuristic with numerous restrictions on the search space properties, the result of which, it is not applicable to most practical problems. In contrast, our approach to developing MAA* and AMAA* is to relax the search criteria in a controlled manner, using a traditional – and less accurate – heuristic. This

philosophy yields methods that balance computational burden while preserving optimality in highly anisotropic search spaces. In particular, AMAA* can effectively tackle the "path oscillation" effect when the search process frequently switches between two regions with similar costs in the vicinity of the search.

Another interesting aspect of anisotropic search space is its effect on hierarchical optimization methods. In principle, hierarchical methods start by finding a number of coarse solutions, select some better candidates (e.g. low cost), and refine them, typically using a more costly method, to obtain the final solution. If the search space is reasonably isotropic, hierarchical approach works well – the selection of a few good coarse solutions will likely provide a near-optimal detailed solution. This is not necessarily the case for an anisotropic search space, as demonstrated in Chapters 5 and 6 when developing LIGA and modified fuzzy inference method. Lack of correspondence between coarse and refined solutions is an essential property anisotropic search space. Certainly, impacts on different methods vary; our work illustrates this point. In case of LIGA, numerous tests have indicated that a reasonable degree of correspondence can still be assumed. For the modified fuzzy method, the impact is so large that we need to develop the constructed variables to maintain the applicability of fuzzy inference method in this anisotropic space.

7.2 Practical contributions

The elements of networked rural electrification include (a) better energy resource utilization with centralized generation; (b) better reliability via interconnecting multiple sources with multiple loads; and (c) higher flexibility to deal with changes in demand and supply. Whether this approach is viable depends on the cost of building the optimal network, which in turn depends on (a) actual topography of the area; and (b) the ability to design and build a cost-effective network.

An important advantage of our approach is the ability to support quick decision at minimal cost. With satellite imagery and GIS tools, we can quickly construct a map, and from it estimate achievable networks and calculate the economic advantage of a networked solution over individually-electrified villages.

Reducing the cost of this type of up-front analysis is a crucial input to decide whether networked rural electrification is suitable for an area.

Besides the mentioned economic advantages, our approach may bring other benefits as well. Since non-recuring engineering costs are often the dominant costs for rural electrification, use of these techniques has the potential to reduce the cost of project planning, preliminary routing, preliminary costing and acquisition of financing for village electrification projects.

In a more general sense, the methods developed in this research could address routing and network design problem under complex topography for other infrastructure projects. Thus, their applications are not limited to networked rural electrification but other important problems such as gas and oil pipeline planning, transportation system, etc.

7.3 Further development

To make these tools more effective, the following areas should be further investigated and developed.

(a) The method of choosing intermediate connection points in AMAA* algorithm. Currently, a radar screening approach is used to estimate change in anisotropicity and hence construct potentially easier paths. Connection points are projected along these paths to enable the search process to quickly escape from path oscillation. While this approach works very well, a more effective connection point estimator could lead to even faster convergence and improved optimality;

(b) The level estimator in LIGA. Level is used for group genotypes for GA processing. Currently, the level is represented by deviation from center line and is a coarse average. Thus, genotypes with very different shapes may still mixed together, potentially slowing down the algorithm. A more sophisticated level measurement could improve the algorithm's speed of convergence;

(c) Score and saving correspondence in the modified fuzzy inference method. As ready discussed in Section 6.4, representing anisotropicity in r_3 with the cost a zip-zap path may lead to better

correspondence. Combining ordinal optimization with fuzzy inference method may bring fundamental improvement.

REFERENCES

[1] United Nations (2015), "Transforming our world: the 2030 Agenda for Sustainable Development," United Nations, A/RES/70/1 , 2015.

[2] IEA, IRENA, UNSD, WB, WHO (2019), "Tracking SDG 7: The Energy Progress Report 2019," Washington DC, 2019.

[3] United Nations (2019), "The Sustainable Development Goals Report 2019," United Nations, 2019.

[4] IEA, IRENA, UNSD, WB, WHO (2021), "Tracking SDG 7: The Energy Progress Report 2021," Washington DC, 2021.

[5] IEA (2020), "World Energy Balances," IEA, Paris, 2020.

[6] M. Bhatia & N. Angelou (2015), "BEYOND CONNECTIONS - Energy Access Redefined," Energy Sector Management Assistance Program, The WorldBank, 2015.

[7] UNDP (2015), "Human Development Report 2015: Work for Human Development," United Nations Development Program, New York, 2015.

[8] M. Torero (2015), "The Impact of Rural Electrification: Challenges and Ways Forward," *Development Economics Review,* vol. 23, pp. 48-75, 2015.

[9] D. Manning, P. Means, D, Zimmerle, K. Galvin, J. Loomis, and K. Paustian (2015), "Using contingent behavior analysis to measure benefits from rural electrification in developing countries: an example from Rwanda," *Energy Policy,* vol. 86, pp. 393-401, 2015.

[10] S. Khandker, H. Samad, R. Ali, and D. Barnes (2014), "Who Benefits Most from Rural Electrification? Evidence in India," *The Energy Journal,* vol. 35, no. 2, pp. 75-96, 2014.

[11] N. Narayana, A. Chamseddinea, V. Vega-Garita, Z. Qina, et al (2019), "Exploring the boundaries of Solar Home Systems (SHS) for off-grid electrification: Optimal SHS sizing for the multi-tier framework for household electricity access," *Applied Energy,* vol. 240, pp. 907-911, 2019.

[12] J. Peters, M. Sievert, and M. Toman (2019), "Rural electrification through mini-grids: Challenges ahead," *Energy Policy,* vol. 132, pp. 27-31, 2019.

[13] S. Nolan, S. Strachan, P. Rakhra, and D. Frame (2017), "Optimized Network Planning of Mini-Grids for the Rural Electrification of Developing Countries," in *2017 IEEE PES PowerAfrica*, pp. 489-494, Accra, 2017.

[14] A. Korkovelos, H. Zerriffi, M. Howells, M. Bazilian, et al (2020), "A Retrospective Analysis of Energy Access with a Focus on the Role of Mini-Grids," *Sustainability,* Vols. 12, 1793, 2020.

[15] UNDP & ETH Energy Politics Group (2018), "Derisking Renewable Energy Investment: Off-Grid Electrification," United Nations Development Programme, New York and Zürich, 2018.

[16] F. Riva, F. D. Sanvito, F. T. Tonini, E. Colombo and F. Colombelli (2019), "Modelling long-term electricity load demand for rural electrification planning," in *2019 IEEE Milan PowerTech*, pp. 1-6, Milan, 2019.

[17] F. Riva, F. Gardumi, A. Tognollo, and E. Colombo (2019), "Soft-linking energy demand and optimisation models for local long-term electricity planning: An application to rural India," *Energy,* vol. 166, pp. 32-64, 2019.

[18] M. E. Khodayar (2017), "Rural electrification and expansion planning of off-grid microgrids," *The Electricity Journal,* vol. 30, no. 4, pp. 68-74, 2017.

[19] B. Tenenbaum, C. Greacen, T. Siyambalapitiya, and J. Knuckles (2014), "From the Bottom Up: How Small Power Producers and Mini-Grids Can Deliver Electrification and Renewable Energy in Africa," Directions in Development, The WorldBank, Washington, DC, 2014.

[20] J. Lin, J. Cai, F. Han, Y. Han, and J. Liu (2016), "Underperformance of Planning for Peri-Urban Rural Sustainable Development: The Case of Mentougou District in Beijing," *Sustainability,* Vols. 8, 858, 2016.

[21] Y. Yi, M. Shi, C. Liu, B. Wang, H. Kang, and X. Hu (2018), "Changes of Ecosystem Services and Landscape Patterns in Mountainous Areas: A Case Study in the Mentougou District in Beijing," *Sustainability,* Vols. 10, 3689.

[22] A. Kottayil (2016), "Technologies and Application of Microgrid/Smartgrid to Asia Pacific," in *9th meeting of Asia Solar Energy Forum, Asian Development Bank*, presented in 3rd seesion, unpublished, Beijing, 2016.

[23] J.C.F. Li (2016), "Managing solar resources under micro-grid settings in remote locations," in *9th meeting of Asia Solar Energy Forum, Asian Development Bank*, presented at Session 3, unpublished, Beijing, 2016.

[24] Worldbank (2019), "Access to electricity (% of population)," 2019. [Online]. Available: https://data.worldbank.org/indicator/EG.ELC.ACCS.ZS (accessed on May 12, 2020).

[25] G. He, D. G. Victor (2017), "Experiences and lessons from China's success in providing electricity for all," *Resources, Conservation, and Recycling,* vol. 122, pp. 335-338, 2017.

[26] NEA (2019), "2019 Electricity Consumption Summary (in Chinese)," [Online]. Available: http://www.nea.gov.cn/2020-01/20/c_138720877.htm (accessed on May 12, 2020).

[27] N. Patel and K. M. Patel (2015), "A survey on enhancement of minimum spanning tree," *International Journal of Engineering Research and Applications,* vol. 5, no. 1/3, pp. 6-10.

[28] S. Russell and P. Norvig (2020), "Chapter 3: Solving Problems by Searching," in *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2020.

[29] J.C.F. Li and D. Zimmerle (2019), "Designing Optimal Network for Rural Electrification using Multiplier-accelerated A* Algorithm," in *2019 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pp. 1-5, Macao, 2019.

[30] J.C.F. Li, D. Zimmerle, and P. Young (2022), "Effective rural electrification via optimal network: Optimal path-finding in highly anisotropic search space using Multiplier-accelerated A* Algorithm," *Energy and AI,* Vols. 7, 100119, 2022.

[31] J.C.F. Li, D. Zimmerle and P. Young (2020), "Optimizing Networked Rural Electrification Design Using Adaptive Multiplier-Accelerated A* Algorithm," in *2020 IEEE / ITU International Conference on Artificial Intelligence for Good (AI4G)*, pp. 164-169, online conference, 2020.

[32] J.C.F. Li, D. Zimmerle, and P. Young (under review), "Flexible Networked Rural Electrification Using Levelized Interpolative Genetic Algorithm," *Energy and AI,* vol. under review, 2022.

[33] J.C.F. Li, D. Zimmerle and P. Young (under review), "Enhancing Networked Rural Electrification Using Modified Fuzzy Inference Method," in *2022 IEEE Global Humanitarian Technology Conference (GHTC)*, Santa Clara, under review, 2022.

[34] J. Smed and H. Hakonen (2017), "Chapter 7: Path Finding," in *Algorithms and Networking for Computer Games*, 2nd ed., Wiley, 2017.

[35] P. Yap, N. Burch, R. Holte and J, Schaeffer (2011), "Any-Angle Path Planning for Computer Games," in *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2011*, pp. 201-207, Stanford, 2011.

[36] X. Cui and H. Shi (2011), "A*-based Pathfinding in Modern Computer Games," *International Journal of Computer Science and Network Security,* vol. 11, no. 1, pp. 125-130 , 2011.

[37] X. Cui and H. Shi (2012), "An Overview of Pathfinding in Navigation Mesh," *International Journal of Computer Science and Network Security,* vol. 12, no. 2, pp. 48-51, 2012.

[38] S. Russell and P. Norvig (2015), "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.

[39] P. Hart, N. Nilsson and B. Raphael (1968), "A Formal Basis for Heuristic Determination of Minimum Cost Paths," *IEEE Transactions of Systems Science and Cybernetics,* vol. 4, pp. 100-107.

[40] N. Nilsson (1971), Problem-Solving Methods in Artificial Intelligence, McGraw-Hill, 1971.

[41] A. Pavlova and A. Pavlov (2018), "Analysis of Correction Methods for Digital Terrain Models Based on Satellite Data," *Optoelectronics, Instrumentation and Data Processing,* vol. 54, no. 5, p. 445–450, 2018.

[42] M. Klimánek (2006), "Optimization of digital terrain model for its application in forestry," *Journal of Forest Science,* vol. 52, no. 5, p. 233–241, 2006.

[43] V. Singh, P. Ray and A. Jeyaseelan (2010), "Orthorectification and Digital Elevation Model (DEM) Generation Using Cartosat-1 Satellite Stereo Pair in Himalayan Terrain," *Journal of Geographic Information System,* vol. 2, pp. 85-92, 2010.

[44] T. Miyasaka, T. Okuro, X. Zhao and K. Takeuchi (2016), "Classification of Land Use on Sand-Dune Topography by Object-Based Analysis, Digital Photogrammetry, and GIS Analysis in the Horqin Sandy Land, China," *Environments,* vol. 3, no. 17, 2016.

[45] J. Bosch, F. Batlles, L. Zarzalejo and G. López (2010), "Solar resources estimation combining digital terrain models and satellite images techniques," *Renewable Energy, ,* vol. 35, pp. 2853-2861, 2010.

[46] M. Martínez-Durbán, L. Zarzalejo, J. Bosch, S. Rosiek, J. Polo and F. Batlles (2009), "Estimation of global daily irradiation in complex topography zones using digital elevation models and meteosat images: Comparison of the results," *Energy Conversion and Management,* vol. 50, pp. 2233-2238, 2009.

[47] J. Tovar-Pescador, D. Pozo-V´azquez, J. Ruiz-Arias, J. Batlles, G. López and J. Bosch (2006), "On the use of the digital elevation model to estimate the solar radiation in areas of complex topography”,," *Meteorological Applications,* vol. 13, p. 279–287, 2006.

[48] H. Dinh, A. Russell, and Y. Su (2007), "On the value of good advice: The complexity of A* with accurate heuristics”,," in *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, pp. 1140-1145, Vancouver, 2007.

[49] H. Dinh, H. Dinh, L. Michel and A. Russell (2012), "The Time Complexity of A* with Approximate Heuristics on Multiple-Solution Search," *Journal of Artificial Intelligence Research,* vol. 45, pp. 685-729, 2012.

[50] A. Felner, R.E. Korf and S. Hanan (2004), "Additive Pattern Database Heuristics," *Journal of Artificial Intelligence Research,* vol. 22, pp. 270-318, 2004.

[51] R.E. Korf, M. Reid and S. Edelkamp (2001), "The Time Complexity of iterative-deepening-A*," *Artificial Intelligence,* vol. 129, pp. 199-218, 2001.

[52] G. Sharon, A. Felner and N. Sturtevant (2014), "Exponential Deepening A* for Real-Time Agent-Centered Search," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14)*, pp. 871–877, Québec City, 2014.

[53] V. Fursov, G. Ye and A. Kotov (2016), "The hybrid CPU/GPU implementation of the computational procedure for digital terrain models generation from satellite images," *Computer Optics,* vol. 40, no. 5, pp. 721-728, 2016.

[54] P. Cooper, D. Friedman and S. Wood (1987), "The Automatic Generation of Digital Terrain Models from Satellite Images by Stereo," *Acta Astronautica,* vol. 15, no. 3, pp. 171-180, 1987.

[55] P. Giles, M. Chapman and S. Franklin (1994), "Incorporation of a Digital Elevation Model Derived from Stereoscopic Satellite Imagery in Automated Terrain Analysis," *Computer & Geosciences,* vol. 20, no. 4, pp. 441-460, 1994.

[56] R. Amatya, M. Barbar, Y. Borofsky, M. Brusnahan, P. Ciller, T. Cotterman, et. al (2018), "Computer-aided electrification planning in developing countries: The reference electrification model (REM)," MIT & IIT-Comillas Universal Energy Access Lab, 2018. [Online]. Available: https://www.iit.comillas.edu/docs/IIT-18-112A.pdf (accessed on May 12, 2020).

[57] P. Ciller, S. Lumbreras (2020), "Electricity for all: The contribution of large-scale planning tools to the energy-access problem," *Renewable and Sustainable Energy Reviews,* vol. 120, pp. 1-16, 2020.

[58] H. Zerriffi (2011), Rural Electrification – strategies for Distributed Generation, Springer, 2011.

[59] V. Kishore, D. Jagu and E. Nand Gopal (2013), "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.

[60] S. Rabin (2000), "A* speed optimizations," in *Game Programming GEMS*, 2000, pp. 264-271.

[61] D. Harabor and A. Grastien (2011), "Online Graph Pruning for Pathfinding on Grid Maps," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI-11)*, pp.1114-1119, San Francisco, 2011.

[62] AP News (2021), "Hundreds rally in VA. In opposition to natural gas pipeline," AP News, 12 Dec 2021. [Online]. Available: https://apnews.com/article/business-mountains-richmond-north-carolina-virginia-a0d6f170fd06d86e589a45a7f5f9a93b (accessed on Feb 18, 2022).

[63] D. Goldberg (1989), Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, 1989.

[64] D. Simon (2013), Evolutionary Optimization Algorithms, Wiley.

[65] A. Pavai and T.V. Geetha (2018), "New crossover operators using dominance and co-dominance principles for faster convergence of genetic algorithms," *Soft Computing,* vol. 23, no. 11, pp. 3661-3686, 2018.

[66] G. Chakraborty and K. Hoshi (1999), "Rank based crossover – a new technique to improve the speed and quality of convergence in GA," in *Proceedings of the 1999 Congress on Evolutionary Computation, Vol. 2*, pp.1595-1602, Washington DC, 1999.

[67] S. A. u. R. Omer and E. Muhammad (2017), "Design of intelligent air conditioner controller using fuzzy logic," in *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, pp. 1-5, Karachi, 2017.

[68] S. M. Sobhy, W. M. Khedr (2015), "Developing of Fuzzy Logic Controller for Air Condition System," *International Journal of Computer Applications,* vol. 126, no. 15, pp. 1-8.

[69] M. Daňková, M. Štěpnička and B. De Baets (2011), "Grades of Monotonicity of Fuzzy Relations and Their Application to Fuzzy Rule Bases," in *2011 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pp. 37-44, Paris, 2011.

[70] H. Seki, H. Ishii and M. Mizumoto (2010), "On the Monotonicity of Fuzzy-Inference Methods Related to T–S Inference Method," *IEEE Transactions on Fuzzy Systems,* vol. 18, no. 3, pp. 629-634, 2010.

[71] Y.C. Ho, Q.C. Zhao and Q.S. Jia (2007), Ordinal Optimization: Soft Optimization for Hard Problems, Springer, 2007.

APPENDIX 1 – PUBLISHED AND PENDING PAPERS

J.C.F. Li, D. Zimmerle and P. Young (under review), "Enhancing Networked Rural Electrification Using Modified Fuzzy Inference Method," in *2022 IEEE Global Humanitarian Technology Conference (GHTC)*, Santa Clara, under review, 2022.

J.C.F. Li, D. Zimmerle, and P. Young (under review), "Flexible Networked Rural Electrification Using Levelized Interpolative Genetic Algorithm," *Energy and AI*, vol. under review, 2022.

J.C.F. Li, D. Zimmerle, and P. Young (2022), "Effective rural electrification via optimal network: Optimal path-finding in highly anisotropic search space using Multiplier-accelerated A* Algorithm," *Energy and AI*, Vols. 7, 100119, 2022.

J.C.F. Li and D. Zimmerle (2021), "Networked Rural Electrification", poster presentation, *3rd International Conference on Solar Technologies & Hybrid Mini Grids to improve energy access*, Sept 2021, Palma de Mallorca, Spain.

J.C.F. Li, D. Zimmerle and P. Young (2020), "Optimizing Networked Rural Electrification Design Using Adaptive Multiplier-Accelerated A* Algorithm," in *2020 IEEE / ITU International Conference on Artificial Intelligence for Good (AI4G)*, pp. 164-169, online conference, 2020.

J.C.F. Li and D. Zimmerle (2019), "Designing Optimal Network for Rural Electrification using Multiplier-accelerated A* Algorithm," in *2019 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pp. 1-5, Macao, 2019.

J.C.F. Li (2016), "Managing solar resources under micro-grid settings in remote locations," in *9th meeting of Asia Solar Energy Forum, Asian Development Bank*, presented at Session 3, unpublished, Beijing, 2016.

APPENDIX 2A – EXAMPLE OF RURAL ENERGY USERS, CONSUMPTION, AND DEVELOPMENT (SUPPLEMENTARY INFORMATION FOR CHAPTER 3)

Rural electrification in China is at a more advanced stage than most developing countries. In 2015, the country had already declared 100% electrification rate [24]. However, detailed statistics about service level are not publicly available. Based on installed capacity, average level of electrification for Chinese rural, off-grid households is approximately tier 3+ in the UN multi-tier framework (min. 200W and 1kWh/day for a household) [25]. Based on the total rural consumption (excluding productive use) outlined in national statistics [26], another estimate of rural electricity consumption is 2000-3000KWh per household per year. Since this total consumption figure includes some grid-connected rural households which likely have higher energy consumption, average consumption of rural households without grid connection could be lower. However, the 2000-3000KWh/year average consumption is large enough that even off-grid households are likely at tier 3+. Therefore, the two estimations indicate that the service level of the average non-grid-connected rural household in China is approximately tier 3+. Such capacity can support basic services such as lighting, TV, phone charging and some light appliances for daily life, with some support for higher-load appliances such as refrigerators. For developmental purposes, however, this capacity level is not sufficient for supporting productive use activities, such as energy-intensive farming or raising livestock for commercial purposes.

The Mentougou district of Beijing, China, Figure 3A.1, represents a typical scenario for rural electrification, and is a near-standard example of rural electrification in China. Villages in the southeastern area have growing populations and have been gradually urbanized. The central grid connects these villages and has been improved to support productive activities. In contrast, most other areas of the district have lower population, and population growth. Further development is becoming more difficult in these areas because of limited electric power. Adding new capacity by increasing SHS penetration or building larger, local minigrids, is typically not economically competitive for productive use activities.

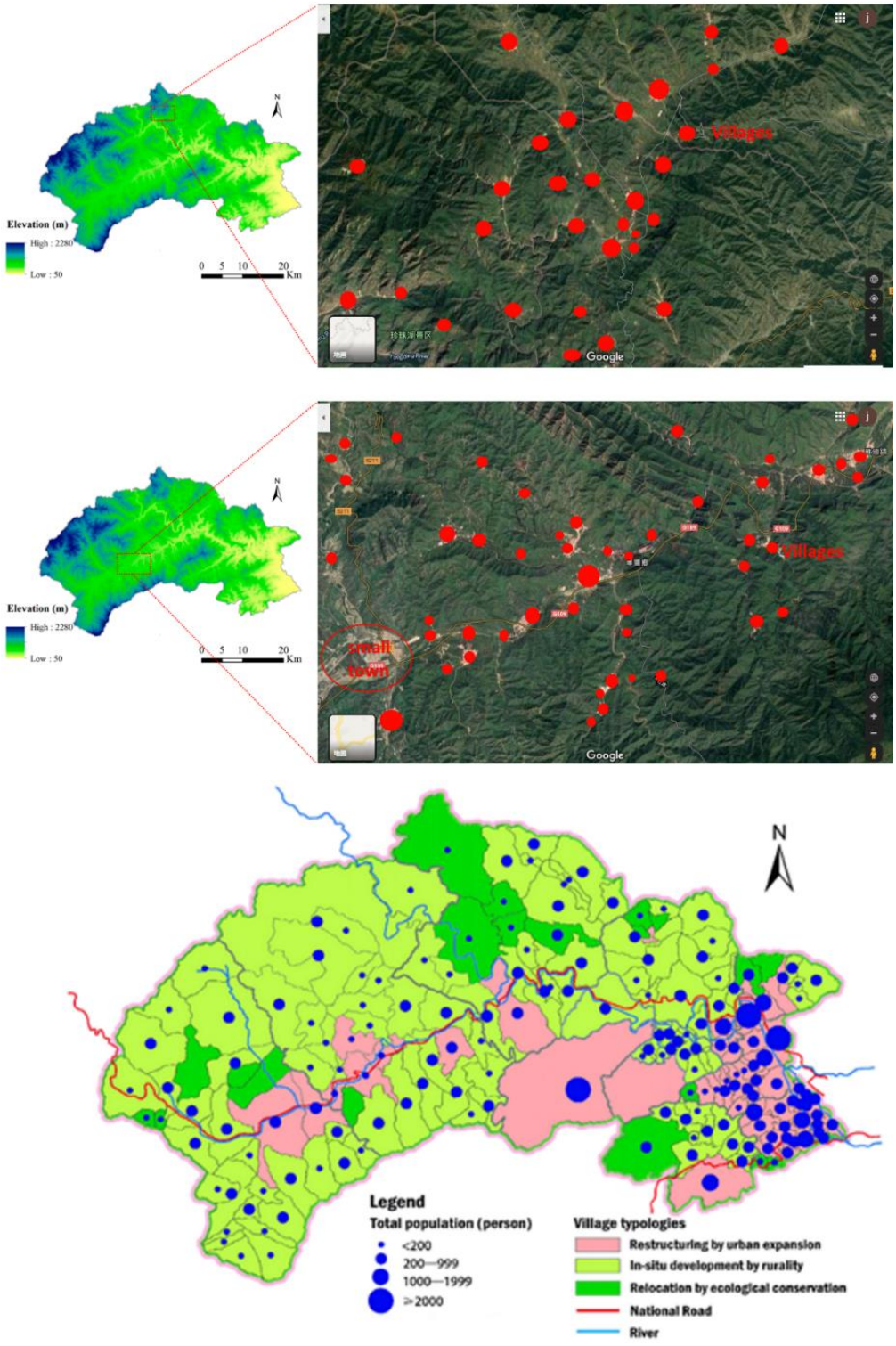


Figure 3A. 1 Geographical structure and population distribution of Mentougou district of Beijing (source: [20] [21]).

APPENDIX 2B – COMPUTATIONAL EFFORT VS HEURISTIC ERROR FOR SEARCHING IN ANISOTROPIC SPACE (SUPPLEMENTARY INFORMATION FOR CHAPTER 3)

Computation burdens of two apparently similar paths can be very different when the search space is highly anisotropic. As explained in section 3.2.1, anisotropicity leads to large error in heuristic estimates, ϵ. Effect of this error is illustrated in Table 3B.1. The heuristic estimate used is the 2D Euclidean distance between the nodes. Average error is average of ϵ computed along each point of the optimal path identified by MAA*(1). Note that the average time required for making one effective move on the optimal path (T/d) grows exponentially as ϵ increases (Figure 3B.1). This trend closely resembles the $O(b^{\epsilon d})$ prediction.

Table 3B. 1 Effect of ϵ

Path		Average Error ϵ	Size of closed list	Optimal Path Length (d)	Optimal Path Cost	Computation time (T), sec	Time per move (T/d), sec
Short	(20,20)-(50,60)	67.9%	191	54	88	23	0.43
	(180,180)-(180,230)	80.9%	1454	53	517	1168	22.0
Medium	(140,65)-(240,30)	73.2%	1010	104	186	688	6.61
	(240,30)-(275,120)	76.9%	2016	105	175	2492	14.2
Long	(120,280)-(275,120)	78.7%	2556	247	363	4652	18.8
	(50,60)-(250,190)	82.9%	3822	270	384	10054	37.2

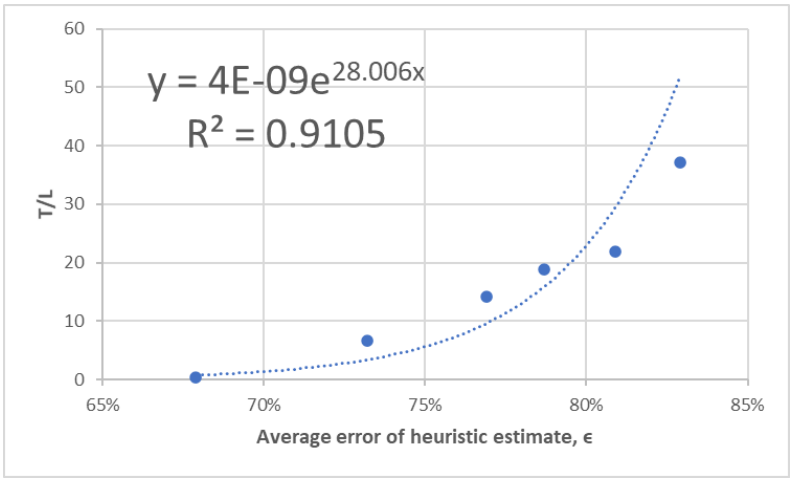


Figure 3B. 1 Effect of error in heuristic estimate, ϵ

It should be emphasized again that computation time is not an authoritative measurement as it will be affected by many factors such as coding proficiency, machine specification, and load variation. However, the observed trend serves as a coarse verification of the validity of $O(b^{\epsilon d})$ complexity prediction.

APPENDIX 2C – FURTHER EXAMPLE – CONNECTING (20,20) AND (290, 290), (SUPPLEMENTARY INFORMATION FOR CHAPTER 3)

Example in this appendix extends the discussion in the main paper by considering a significantly longer path length d than the example in Section 3.2.2; the computational burden is much heavier. This is well $O(b^{\epsilon d})$ complexity estimation. However, MAA* can still effectively accelerate the search while trading off some optimality. Results are summarized in Table 3C.1 and Table 3C.2

Table 3C. 1 Computation Results using MAA* for connecting (20,20)-(290,290)

Path (20,20) - (290,290), time measured or projected based on i5					
Cut-off	Multiplier	Time	Closed Point	Path Length	Cost
10	0.5	1046660.62 (290.74 hrs)	41973	433	614.58
	1	192432.06 (53.5 hrs)	14501	421	633.99
	2.5	7741.62 (2.2 hrs)	3915	409	858.12
	5	4591.88 (1.3 hrs)	409	394	1254.74
Global Min	0	not determined			

Table 3C. 2 Comparison among different m values, connection (50,60)-(250,190)

	Connection: (20,20) - (290,290) @ c = 10 Direct connection cost = 3396.94				
	Computed Optimal Cost	Computation Time (sec)	Normalized Computation Time	% Over-estimate w.r.t. optimal	% improvement w.r.t. direct connection
Best MAA* (m=0.5)	614.58	1046660.62	100.0%	0.0%	76.5%
MAA*					
m = 1	633.99	192432.06	18.4%	3.2%	75.8%
m = 2.5	858.12	7741.62	0.7%	39.6%	67.2%
m = 5	1254.74	4591.88	0.4%	104.2%	52.1%

Computation times for this example are significantly longer than the example in Section 3.2.2. Indeed the global optimum could not be computed with the available computational resources. Since the global optimal has not been determined, the best MAA* result is used for comparison. Besides the obvious fact

that the separation between the point pair is larger, a more symmetrical distribution of costly elements (i.e. for this case, the low accessibility area in the middle of the map) is also an important factor. Any such structure leads to many paths of comparable cost on both side of the line jointing the point pair, making the spectrum of estimated cost even denser and the search even more jumpy. Figure 3C.1 shows the physical shapes of the computed paths.

In practice, this type of high-cost obstacle to routing is common. Examples include river crossings in rough terrain, lakes or wetlands, and reserved areas such as nature or wildlife preserves or military areas, or areas with impermeable or difficult soil structures, such as exposed bedrock or unstable sand. In all of these cases, lines need to be routed around *and within* difficult regions, as illustrated in Figure 4.2.

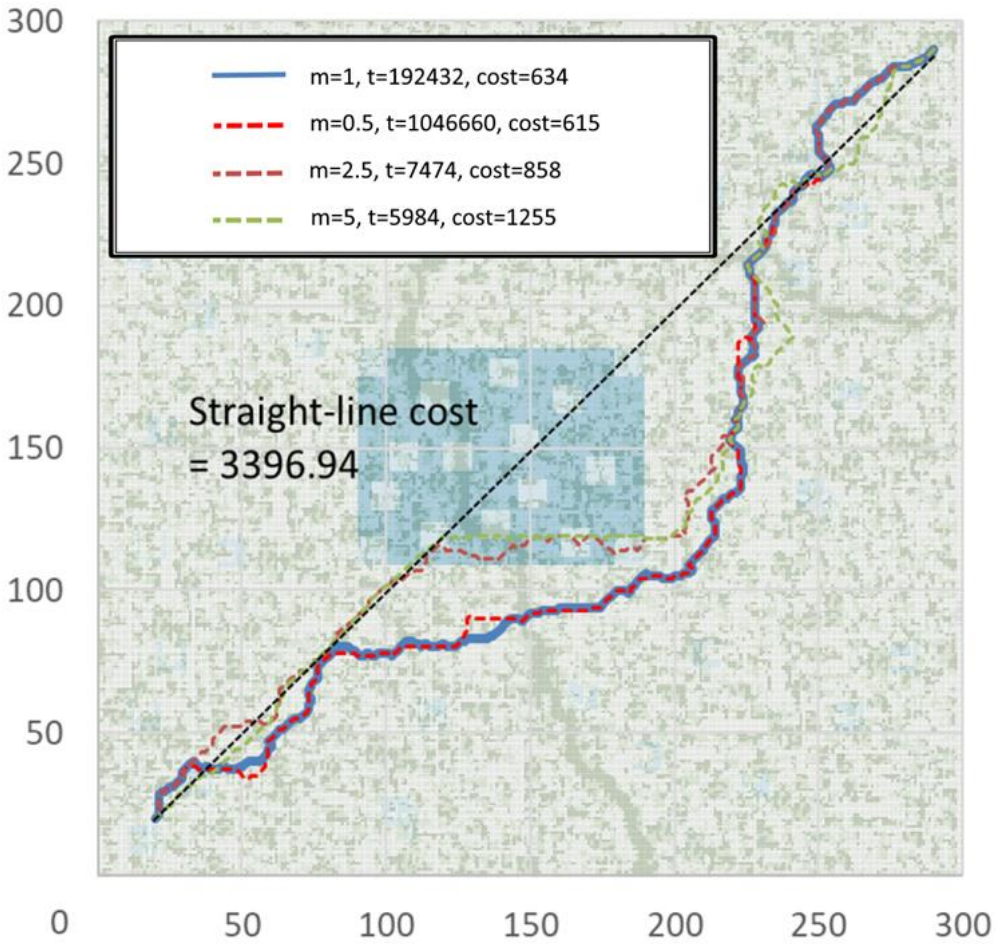


Figure 3C. 1 Actual path shapes: A* and MAA* @ c = 10, connection (20,20)-(290,290)

APPENDIX 2D – COMPARISON OF RESULTS BASED ON DIFFERENT m VALUES (SUPPLEMENTARY INFORMATION FOR CHAPTER 3)

Table 3D.1 summarizes the computation results of the 66 optimal routes based on Figure 3.9. It can be seen that computation efficency varies significantly with the value of m (color-coded). The accuracy-speed trade off among different m values are discussed in section 3.3.2.

Table 3D.1 Computation Results of MAA* Algorithm at m=1, 2 and 5

Route ID	Point 1		Point 2		Straight-line Cost (S)	R^2-distance (D)	MAA*(1)			MAA*(2)				MAA*(5)				
	x	y	x	y			Cost (C)	Compu-time (T)	T/D	Cost	Compu-time	T/D	optimality impact	compu-time reduction	Cost	Compu-time	T/D	optimality impact	compu-time reduction
1	20	20	50	60	290	50	88	23	0.5	89	5	0.1	0.3%	79.2%	108	2	0.0	22.2%	90.3%
2	20	20	70	220	1008	206	328	1329	6.4	384	1527	7.4	17.1%	-14.9%	490	110	0.5	49.6%	91.7%
3	20	20	110	140	1049	150	444	7505	50.0	500	656	4.4	12.7%	91.3%	527	193	1.3	18.8%	97.4%
4	20	20	120	280	1399	279	435	16970	60.9	544	5359	19.2	25.1%	68.4%	629	1725	6.2	44.6%	89.8%
5	20	20	140	65	616	128	217	1615	12.6	243	145	1.1	11.9%	91.0%	259	71	0.6	19.5%	95.6%
6	20	20	180	180	2367	226	910	22502	99.4	979	5219	23.1	7.7%	76.8%	1182	5229	23.1	29.9%	76.8%
7	20	20	180	230	3047	264	471	95499	361.7	571	1875	7.1	21.2%	98.0%	694	3563	13.5	47.4%	96.3%
8	20	20	240	30	1009	220	359	9691	44.0	417	3600	16.3	16.4%	62.8%	595	149	0.7	66.0%	98.5%
9	20	20	250	190	2661	286	470	17199	60.1	609	816	2.9	29.7%	95.3%	796	411	1.4	69.4%	97.6%
10	20	20	275	120	1279	274	443	25170	91.9	519	720	2.6	17.0%	97.1%	570	461	1.7	28.6%	98.2%
11	20	20	290	290	3397	382	604	59141	154.9	763	3662	9.6	16.6%	93.8%	1182	2765	7.3	80.8%	95.3%
12	50	60	70	220	757	161	225	1685	10.5	301	217	1.3	33.9%	87.1%	417	80	0.5	85.6%	95.2%
13	50	60	110	140	759	100	335	202	2.0	371	162	1.6	10.6%	19.9%	391	80	0.8	16.7%	60.6%
14	50	60	120	280	1121	231	359	215	0.9	397	102	0.4	15.2%	85.8%	531	68	0.3	47.8%	68.5%
15	50	60	140	65	482	91	151	665	7.4	195	95	1.0	2.5%	85.7%	193	138	1.5	27.3%	79.2%
16	50	60	180	180	2238	177	809	7034	39.8	867	2668	15.1	7.2%	62.1%	1014	3249	18.4	25.3%	53.8%
17	50	60	180	230	2738	214	364	1475	6.9	395	612	2.9	8.4%	58.5%	578	42	0.2	58.7%	97.1%
18	50	60	240	30	994	192	326	4946	25.7	386	1212	6.3	18.6%	75.5%	367	49	0.3	12.9%	99.0%
19	50	60	250	190	2620	239	384	10054	42.1	521	324	1.4	35.6%	96.8%	651	159	0.7	69.3%	98.4%
20	50	60	275	120	1005	233	380	15187	65.2	443	139	0.6	16.7%	99.1%	568	495	2.1	49.7%	96.7%
21	50	60	290	290	3319	332	594	49922	150.2	681	1549	4.7	14.6%	96.9%	944	168	0.5	58.9%	99.7%
22	70	220	110	140	812	89	339	3045	34.0	370	470	5.3	9.2%	84.6%	406	146	1.6	19.7%	95.2%
23	70	220	120	280	364	78	149	1594	20.4	185	177	2.3	24.0%	88.9%	207	16	0.2	38.8%	99.0%
24	70	220	140	65	1604	170	299	6622	38.9	315	453	2.7	5.5%	93.2%	444	107	0.6	48.7%	98.4%
25	70	220	180	180	1432	117	666	6213	53.1	723	2064	17.6	8.6%	66.8%	741	700	6.0	11.2%	88.7%
26	70	220	180	230	1872	114	439	6130	55.5	236	2220	20.1	18.2%	63.8%	243	159	1.4	22.1%	97.4%
27	70	220	240	30	2857	255	461	15873	62.3	512	4569	17.9	10.9%	71.2%	693	240	0.9	50.3%	98.5%
28	70	220	250	190	676	182	328	1462	8.0	372	241	1.3	13.6%	83.5%	404	116	0.6	23.1%	92.0%
29	70	220	275	120	3526	228	430	54630	239.5	505	12976	56.9	17.4%	76.2%	604	246	1.1	40.4%	99.5%
30	70	220	290	290	1065	231	377	10206	44.2	451	1264	5.5	19.4%	87.6%	555	210	0.9	47.1%	97.9%
31	110	140	120	280	2098	140	420	8627	61.5	451	3919	27.9	7.4%	54.6%	533	642	4.6	26.9%	92.6%
32	110	140	140	65	923	81	360	1197	14.8	395	421	5.2	9.7%	64.8%	414	177	2.2	14.9%	85.2%
33	110	140	180	180	1838	81	932	2390387	29649.1	942	918011	1138.5	1.1%	61.6%	1031	253078	3139.1	10.6%	89.4%
34	110	140	180	230	1872	114	439	13558	118.9	464	3811	33.4	5.5%	71.9%	617	1231	10.8	40.4%	90.9%
35	110	140	240	30	1319	170	515	45450	267.4	541	2097	12.3	5.4%	51.8%	699	715	4.2	31.4%	81.6%
36	110	140	250	190	1349	149	564	5248	35.3	697	2297	13.2	7.4%	54.6%	772	1499	10.0	30.8%	71.5%
37	110	140	275	120	2745	166	538	10213	61.4	570	2301	13.8	6.0%	77.5%	970	1577	9.5	80.3%	84.6%
38	110	140	290	290	2820	234	648	21591	92.1	673	2226	9.5	3.9%	89.7%	853	1141	4.9	31.7%	94.7%
39	120	280	140	65	2849	216	399	18339	84.9	444	2852	13.2	11.4%	84.4%	587	1219	5.6	47.2%	93.4%
40	120	280	180	180	880	117	660	14109	121.0	677	1913	16.4	2.6%	86.4%	716	508	4.4	8.5%	96.4%
41	120	280	180	230	298	78	160	3022	38.7	179	718	9.2	11.8%	76.3%	197	118	1.5	23.4%	96.1%
42	120	280	240	30	2981	277	493	54939	198.1	569	3338	12.0	15.4%	93.9%	791	1746	6.3	60.4%	96.8%
43	120	280	250	190	597	158	256	755	4.8	324	215	1.4	26.7%	71.5%	389	33	0.2	52.0%	95.6%
44	120	280	275	120	874	223	363	4652	20.9	453	5581	25.1	24.8%	-20.0%	593	227	1.0	63.6%	95.1%
45	120	280	290	290	794	170	289	2503	14.7	389	947	5.6	34.7%	62.2%	399	114	0.7	38.4%	95.5%
46	140	65	180	180	2057	122	729	11111	91.3	793	5319	43.7	8.8%	52.1%	842	6445	52.9	15.5%	42.0%
47	140	65	180	230	2621	170	364	32354	190.6	461	4113	24.2	26.7%	87.3%	523	809	4.8	43.7%	97.5%
48	140	65	240	30	366	106	186	688	6.5	192	131	1.2	3.4%	80.9%	227	43	0.4	22.1%	93.8%
49	140	65	250	190	1096	167	277	676	4.1	388	114	0.7	40.5%	83.1%	453	32	0.2	63.9%	95.2%
50	140	65	275	120	676	146	248	1075	7.4	294	404	2.8	18.4%	62.4%	353	137	0.9	42.1%	87.2%
51	140	65	290	290	2285	270	493	51035	188.7	584	4889	18.1	18.3%	90.4%	711	1062	3.9	44.7%	97.9%
52	180	180	180	230	682	50	517	1168	23.4	517	701	14.0	0.0%	39.9%	522	318	6.4	1.0%	72.8%
53	180	180	240	30	2153	162	736	4454	27.6	738	4476	27.7	0.3%	-0.5%	799	1763	10.9	8.5%	60.4%
54	180	180	250	190	866	71	545	869	12.3	556	382	5.4	1.9%	56.0%	544	136	1.9	-0.2%	84.4%
55	180	180	275	120	1157	112	645	5891	52.4	661	1476	13.1	2.4%	74.9%	668	423	3.8	3.6%	92.8%
56	180	180	290	290	1020	156	764	15548	100.5	779	2210	14.2	1.9%	85.8%	770	191	1.2	9.4%	98.1%
57	180	230	240	30	1983	209	360	2895	13.9	386	1791	8.6	7.0%	38.1%	453	191	0.9	25.8%	93.4%
58	180	230	250	190	402	81	130	81	1.0	159	27	0.3	21.8%	66.5%	177	11	0.1	35.4%	86.0%
59	180	230	275	120	581	145	261	2726	18.8	274	1009	6.9	4.9%	63.0%	370	37	0.3	41.7%	98.6%
60	180	230	290	290	486	125	209	750	6.0	240	182	1.5	14.7%	75.7%	327	38	0.3	56.4%	94.9%
61	240	30	250	190	758	160	279	1271	7.9	306	914	5.7	9.5%	28.1%	331	65	0.4	18.7%	94.9%
62	240	30	275	120	470	97	175	2492	25.8	177	222	2.3	1.1%	91.1%	224	100	1.0	28.3%	96.0%
63	240	30	290	290	1590	265	445	20119	76.0	515	3995	15.1	15.8%	80.1%	649	1278	4.8	46.1%	93.7%
64	250	190	275	120	404	74	125	35	0.5	128	14	0.2	3.2%	61.6%	193	28	0.4	54.6%	19.9%
65	250	190	290	290	705	108	253	6064	56.3	275	278	2.6	9.0%	95.4%	351	78	0.7	39.1%	98.7%
66	275	120	290	290	1043	171	312	10291	60.3	347	289	1.7	11.0%	97.2%	467	61	0.4	49.4%	99.4%

Further to the discussion in Section 3.3.1, Figure 3D.1 provides a more comprehensive comparison between MAA*(1) and MAA*(2).

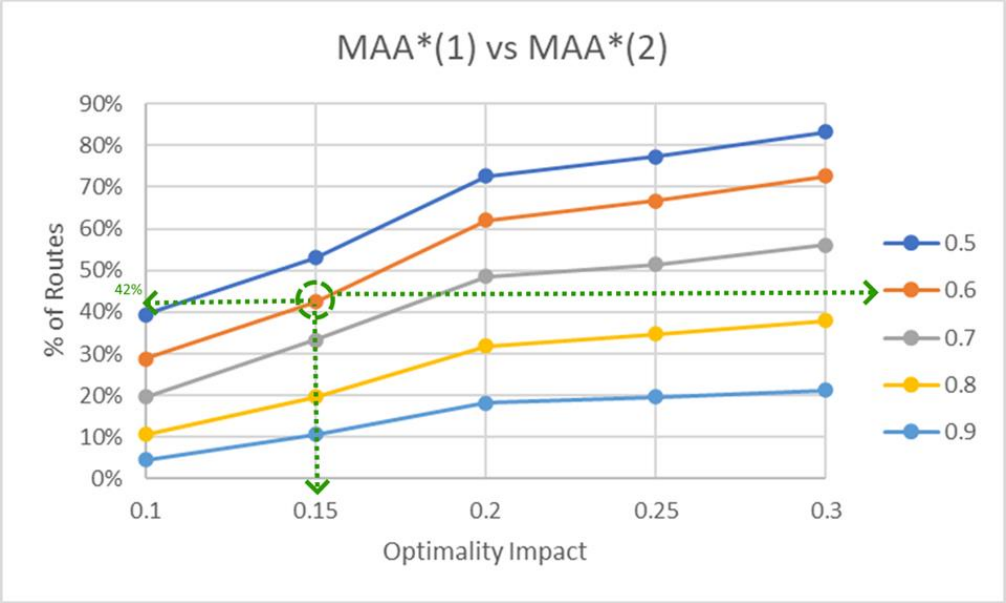


Figure 3D. 1 Optimality-Speed Comparison, MAA*(1) vs MAA*(2)

As shown by the green arrows in the figure. when 15% optimality impact is allowable, 42% of the MAA*(2) routes can offer 60% or more reduction in computation time (the orange line). Users can, according to their purposes and accuracy requirements, decide the appropriate trade-off. For example, if the user is conducting an initial screening, a higher optimality impact may be allowed but computation speed need to be fast. With 20% optimality impact, the deep-blue line indicates that 72% MAA*(2) routes can offer more than 50% computation time reduction, the orange line indicates that 61% MAA*(2) routes can offer more than 60% speed improvement, and so on. As pointed out in Section 3.3.2, actual optimality impact and computation time reduction will depend on topography and hence vary case by case. However, this example acts as a guideline for users to choose suitable m value for their application. It should be reminded that m is not necessarily an integer. One can, for example, choose $m = 1.6$ in order to find the right trade-off for his / her problem. It is also possible to use $m < 1$ for more demanding applications. As

long as $m > 0$, MAA* method will be faster than canonical A*. In another extremely, large m value can significantly reduce computation burden but may not be able to provide meaningful accuracy. This is illustrated in Figure 3D.2 with $m = 5$. Only small percentage of MAA*(5) routes can preserve reasonable optimality.

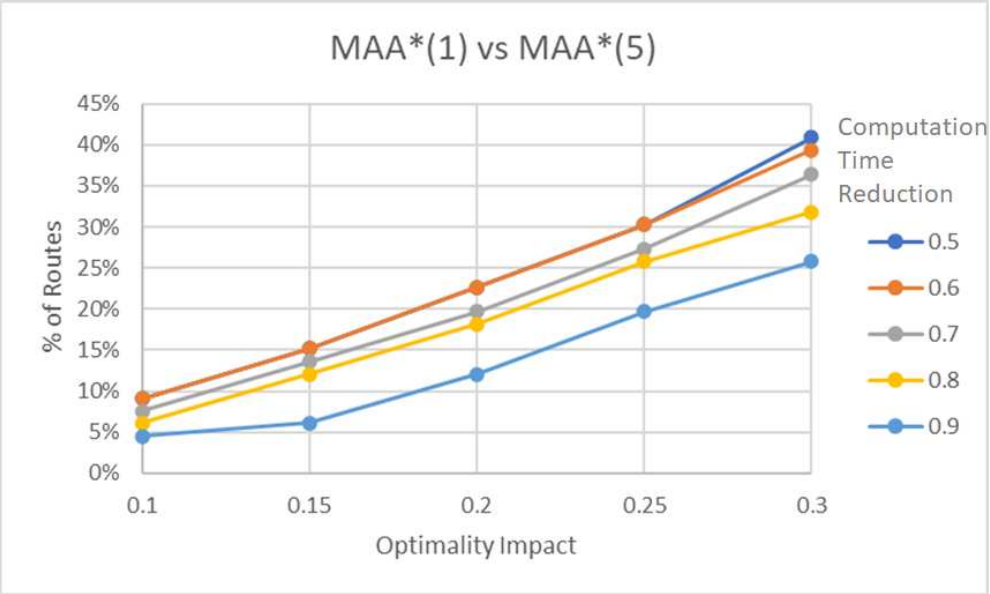


Figure 3D. 2 Optimality-Speed Comparison, MAA*(1) vs MAA*(5)

APPENDIX 3A – EXAMPLE OUTPUT FROM LIGA, CONNECTING (50,60) AND (250,190), (SUPPLEMENTARY INFORMATION FOR CHAPTER 5)

The following figures are example LIGA output for level = 2. Figure 5A.1 shows the shape of the coarse near-optimal routes. Figure 5A.2 shows the speed of convergence.

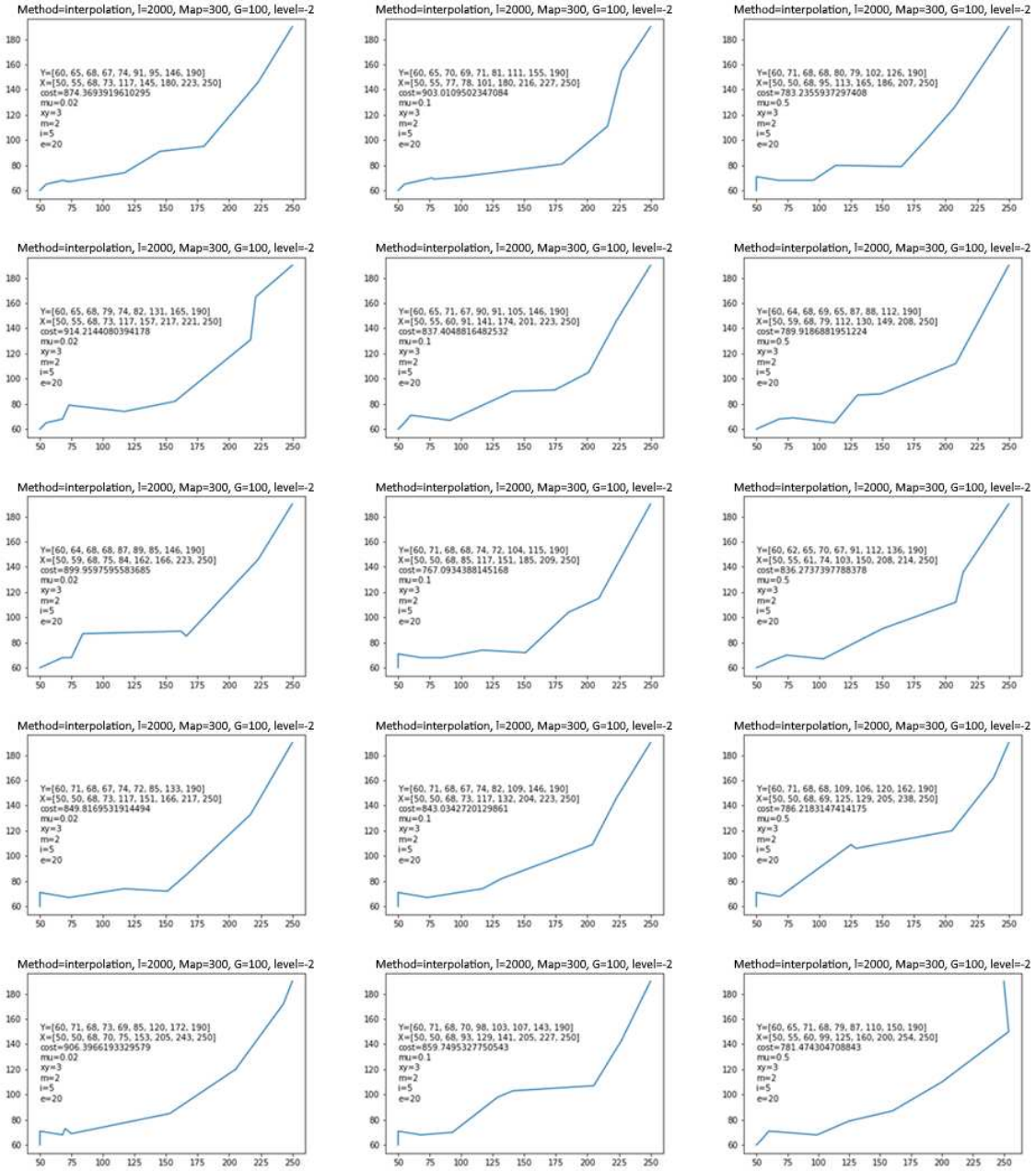


Figure 5A. 1 Physical shape, LIGA Path Output for connecting (50,60) and (250,190), level = -2, G = 100

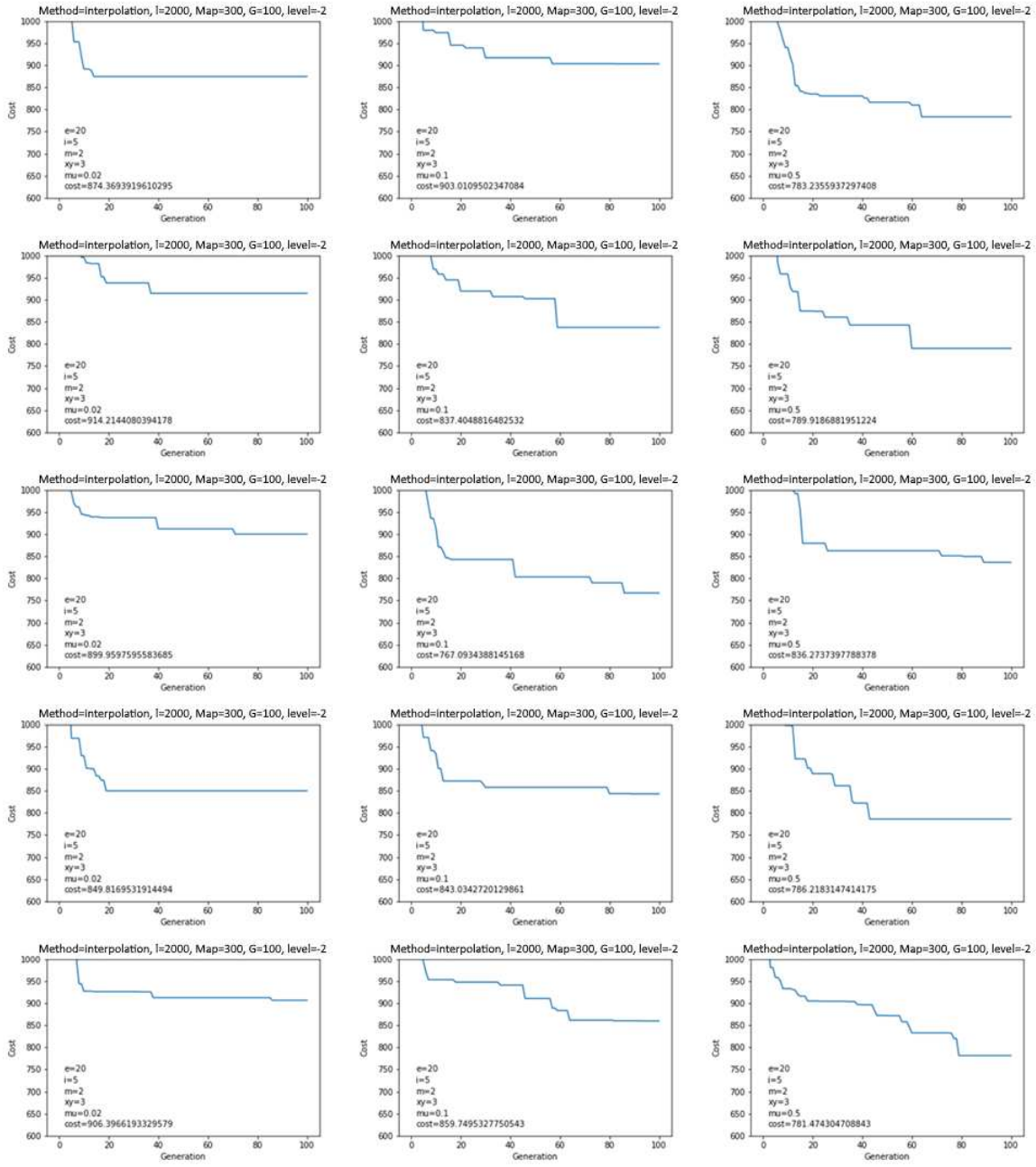


Figure 5A. 2 Convergence, LIGA Cost Output for connecting (50,60) and (250,190), level = -2, G = 100

APPENDIX 3B – CORRESPONDENCE BETWEEN COARSE AND REFINED ROUTE COST (SUPPLEMENTARY INFORMATION FOR CHAPTER 5)

Section 4 discussed the correspondence between coarse and refined route cost. A total of 120 coarse routes are refined. Table 5B.1 shows the associated levels of these routes and the numerical values plotted in Figure 14.

Table 5B. 1 Correspondence between coarse and refined route cost, Connection (50,60)-(250, 190), G=30

Route ID	Level	Test ID	Coarse Cost	Refined Cost
21	-2	C5	782.3	410.0
27	-2	C3	795.5	449.9
28	-2	B3	804.4	456.8
118	full	C1	804.6	472.6
105	full	C3	807.0	410.5
39	-1	C5	812.9	443.8
106	full	B4	825.1	411.0
34	-1	A5	834.3	432.2
5	-3	B4	837.7	450.2
110	full	C4	842.6	426.5
15	-2	C2	845.9	379.6
33	-1	C2	847.9	427.5
26	-2	C1	852.5	449.1
107	full	A3	855.8	422.6
19	-2	C4	856.0	401.2
109	full	B2	858.9	425.5
4	-3	C2	861.5	447.5
40	-1	B4	863.6	446.1
112	full	C2	863.7	443.8
114	full	B5	864.1	451.1
1	-3	C4	865.8	424.4
20	-2	B1	866.0	407.2
24	-2	A5	875.7	435.4
22	-2	A2	879.5	412.2
30	-1	B5	882.5	391.8
108	full	A1	883.7	424.7
116	full	C5	884.2	467.0
41	-1	C1	885.0	459.0
119	full	A2	885.8	485.6
0	-3	C5	888.7	398.1
117	full	B1	890.0	468.2

23	-2	A3	892.8	431.4
6	-3	B1	893.9	466.4
35	-1	B1	895.1	432.2
31	-1	C3	897.1	405.5
115	full	B3	901.2	452.2
113	full	A5	902.0	449.1
17	-2	A4	904.3	394.1
38	-1	C4	905.6	438.4
43	-1	A3	913.2	472.0
37	-1	A1	916.9	436.2
36	-1	A4	917.3	432.7
18	-2	A1	918.1	398.5
32	-1	B2	918.8	425.6
25	-2	B2	924.5	441.6
16	-2	B4	925.5	390.8
7	-3	C3	927.0	488.7
111	full	A4	927.7	438.0
54	0	C4	929.6	495.8
53	0	C5	933.2	490.3
2	-3	B3	937.7	434.2
42	-1	B3	938.9	463.3
3	-3	B2	953.7	447.0
29	-2	B5	954.4	498.6
48	0	C1	956.6	480.8
55	0	B4	958.6	497.7
47	0	B5	961.5	471.7
9	-3	C1	961.9	494.7
57	0	C2	963.0	505.1
56	0	C3	964.3	504.5
80	2	C1	972.6	551.5
10	-3	A4	980.1	499.2
13	-3	A5	991.3	516.2
59	0	B2	994.2	518.1
67	1	B1	1001.3	547.5
61	1	B2	1004.3	474.1
44	-1	A2	1009.0	509.7
58	0	B1	1010.3	506.0
8	-3	A1	1012.0	489.7
52	0	A1	1012.9	489.5
49	0	A4	1015.9	480.9
75	2	A1	1016.1	528.5
78	2	B5	1018.4	539.6
12	-3	B5	1019.4	510.5
64	1	C4	1021.3	503.0
83	2	A2	1022.9	575.7

51	0	B3	1023.1	487.8
77	2	C3	1023.4	536.4
94	3	C2	1034.2	557.9
14	-3	A2	1036.0	557.4
65	1	C3	1043.6	543.2
62	1	B3	1045.8	476.7
11	-3	A3	1051.7	509.4
84	2	B4	1056.9	577.4
87	2	C5	1057.9	591.0
79	2	A5	1063.1	551.0
45	0	A5	1064.3	460.8
82	2	C4	1065.8	568.4
46	0	A2	1066.2	462.9
99	3	C1	1067.6	580.4
74	1	C5	1069.8	601.7
93	3	C3	1070.8	538.9
68	1	C2	1078.7	552.2
86	2	B1	1081.3	583.4
92	3	C5	1082.0	527.0
72	1	C1	1089.8	585.8
81	2	B3	1092.1	558.2
60	1	A5	1098.3	457.6
73	1	A2	1100.1	596.5
71	1	B5	1101.2	559.7
85	2	A4	1101.3	580.1
66	1	B4	1103.4	543.8
88	2	C2	1103.7	591.8
102	3	B1	1107.0	600.2
50	0	A3	1108.6	486.6
91	3	C4	1115.5	526.9
89	2	A3	1122.4	598.4
63	1	A3	1129.9	487.9
76	2	B2	1137.1	533.1
70	1	A1	1156.2	556.8
98	3	B4	1161.1	578.5
90	3	B3	1163.1	524.4
97	3	A3	1168.3	572.1
103	3	B5	1170.8	602.1
101	3	B2	1178.9	593.1
69	1	A4	1182.0	556.5
95	3	A2	1205.5	558.2
100	3	A4	1221.0	580.9
104	3	A1	1225.4	604.0
96	3	A5	1282.9	565.8

APPENDIX 3C – FURTHER CONNECTION EXAMPLE: (20,20)-(290,290), (SUPPLEMENTARY INFORMATION FOR CHAPTER 5)

By increasing the number of generations G in genetic algorithm from 30 to 100, further results for connecting (20,20) and (290,290) are obtained and shown in Table 5C.1 and Figure 5C.1.

Table 5C. 1 Computation results using hybrid method, G = 100, connection (20,20)-(290,290)

Method	Path Cost
Accelerated A*with m=1	633.99
Hybrid	
No. of generation in LIGA/GA = 100	
level = 2	711.73
level = 1	656.41
level =-2	683.78
level = -3	761.16
traditional GA (index = full)	649.11
Straight-line	2620.03
Accelerated A* with m=0.5	614.58

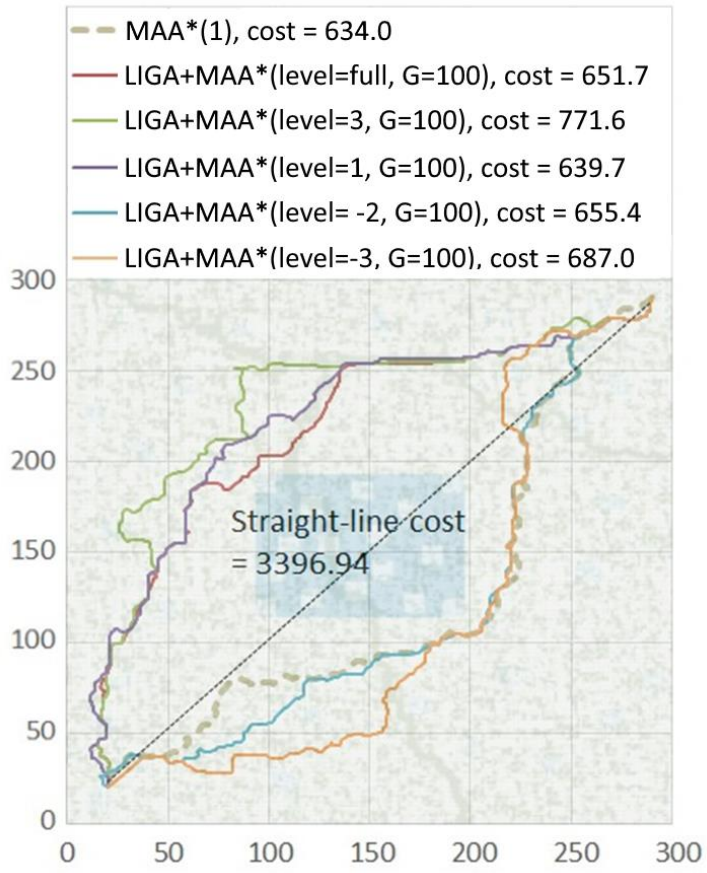


Figure 5C. 1 Actual path shapes using hybrid method, G = 100, connection (20,20)-(290,290)

Compared to Table 3 and Figure 17 in Section 5, no significant change is observed by increasing G to 100.