Kathmandu University Journal of Science, Engineering and Technology, Vol. 19, No. 1, March 2025

Kathmandu University
Journal of Science, Engineering and Technology

On optimizing travel routes in Kathmandu Valley, Nepal using genetic algorithms

Astha Pandey* , Sujan Pokharel , Ujjwal Karki , Anand Pant , Saishab Bhattarai , and Kanhaiya Jha

Department of Mathematics, School of Science, Kathmandu University, Dhulikhel, Kavre, Nepal.

Abstract

Effective route scheduling is crucial for reducing both travel time and expenses, particularly in industries like transportation and travel. One of the optimization methods known as the Travelling Salesman Problem (TSP), offers a useful structure for solving these issues, but because of its NP-complete characteristics result in the need for significant computational resources to solve it perfectly.In this study, we provide a unique approach to route optimization that makes use of a genetic algorithm (GA) and provides a heuristic approach which delivers almost optimal solutions in reasonable time. The concept applied in our methodology minimizes the distance between various destinations within Kathmandu Valley.

Keywords: Genetic algorithm; Route optimization; Traveling salesman problem; Heuristic method.

1. Introduction

For generations, one of the core challenges of mathematics and logistics has been the optimization of transportation routes. French mathematician Gaspard Monge [1] explored the transportation problem from a geometric point of view and first formalized it in 1781. The foundation for contemporary optimization techniques was laid in the 1920s when Tolstoi [2] took a mathematical approach to the transportation problem, marking a significant improvement . In the 1960s, Holland [3] made a significant contribution to the industry by introducing genetic algorithms (GAs), which offered a fresh method for using computers to solve challenging transportation problems. The Traveling Salesman Problem (TSP), which aims to determine the shortest path that makes exactly a single stop in each city and then returns to the starting point city, is one of the most well-known transportation optimization problems. Despite its seeming simplicity, TSP is an NP-hard problem that grows exponentially complex with the number of sites [4]. Large-scale real-world applications make traditional solution techniques not feasible, requiring more complex strategies like genetic algorithms.

Inspired by the concepts of evolution and natural selection, genetic algorithms are an effective tool for solving challenging optimization issues [5]. These algorithms operate by maintaining a population of potential solutions, selecting the fittest individuals, and applying genetic operators such as crossover and mutation to evolve better solutions over successive generations. Even while GAs need a lot more processing power than conventional paper-and-pen methods, they are especially useful for complicated transportation problems because of their capacity to manage numerous fitness characteristics and large search space.

The Kathmandu Valley is experiencing fast urbanization and population increase, which has made transportation issues more complicated. Effective route optimization is crucial since the valley's road system cannot handle the everyday mobility demands of its more than 2.5 million inhabitants. Route design is a complex optimization problem. The unique characteristics of the transportation network in the Kathmandu Valley are frequently not sufficiently addressed by conventional routing techniques. The use of genetic algorithms to optimize routes inside the Kathmandu Valley is examined in this study. Our approach is to produce useful, effective routes that take into account the dynamic nature of urban traffic patterns.

In 2001, Chien et al. [6] presented a genetic algorithm (GA) approach for public transport route planning and design, emphasizing in producing a cost effective route. The algorithm begins by creating an initial population of routes which are the possible solutions. This is based on a predetermined population size, road patterns in the service area of each route and estimating an objective value that represents all costs. Next, the algorithm selects the route with the least cost. With this process a new population is created to ensure overall quality in terms of cost. Additionally, crossover and mutation operators are applied to create new routes, replacing some old ones in the population. This iterative refinement guarantees that the newly generated routes are progressively more cost-effective than their predecessors.

In 2015, Herring and Eden [7] explored the use of evolutionary algorithms for de novo molecular design under multi-dimensional constraints. The authors emphasized the necessity of a large search space for the effective implementation of evolutionary algorithms. By leveraging the expansive search capabilities of GAs, the study highlighted how these algorithms are well-suited for navigating complex and vast solution spaces, ensuring the generation of optimal results in molecular design.

In 2015, Rao and Hegde [8] conducted a literature survey on the application of genetic algorithms (GA) to the traveling salesman problem (TSP), introducing a novel crossover method aimed at producing better tours. The study also highlighted the formation of subtours, which share the same fitness value, as a critical step in

*Corresponding author. Email: asthap978@gmail.com

the optimization process. These innovative operations, when combined, demonstrated the potential to achieve improved solutions, reinforcing the effectiveness of genetic algorithms in solving complex optimization problems like the TSP.

In 2017, Hussain et al. [9] explored the traveling salesman problem (TSP) using a genetic algorithm (GA) with a modified cycle crossover operator. Their experiments compared various crossover methods and revealed that all operators exhibited a similar performance pattern. Additionally, they observed minimal variation between the best, worst, and average values, indicating a consistent performance across different crossover techniques in solving the TSP.

In 2019, Mohammed et al. [10] and in 2009, Wang et al. [11] successfully applied genetic algorithms (GA) to address the vehicle routing problem (VRP), albeit in different contexts. Mohammed et al. focused on solving the VRP within a university setting, identifying optimal routes for vehicle transportation and demonstrating the GA's effectiveness in optimizing routes, reducing costs, and addressing inefficiencies. Their work highlighted the practical applicability of GAs in real-world VRP scenarios. Similarly, Wang et al. enhanced the GA framework to tackle the VRP with breakdown vehicles, introducing improvements such as abbreviated coding length to streamline computations and improve solving efficiency. Their advancements not only optimized the algorithm's performance but also underscored the potential of refined GA techniques for handling complex and dynamic routing challenges. Together, these studies showcase the versatility and robustness of genetic algorithms in addressing diverse VRP scenarios.

2. Methodology

2.1. Traveling salesman problem

The mathematical model for the TSP can be expressed as [12]:

$$
\begin{aligned}
x_{ij} &= \begin{cases} 1, & \text{if travel occurs from } i \text{ to } j, \\ 0, & \text{otherwise.} \end{cases} \\
\text{Minimize: } & \sum_{i=1}^{n-1} \sum_{j=1}^{n} d_{ij} x_{ij}, \\
\text{Subject to: } & \sum_{i=1}^{n} x_{ij} = 1, \quad \text{for } j = 1, \dots, m, \\
& \sum_{j=1}^{m} x_{ij} = 1, \quad \text{for } i = 1, \dots, n.
\end{aligned} \tag{1}
$$

2.2. Mapping TSP to GA

Traveling salesman problem aims to save costs while determining the quickest route through the specified destinations. The application of a genetic algorithm enhances the solution to the TSP. Generating the initial population for a traveling salesman challenge entails identifying every potential tour. Every chromosome is a representation of the traveler's journey. The cities to be visited are represented by each gene in the chromosome. The chromosome's length is always equal to the sum of all cities plus one.

2.3. Algorithm for GA [9]

1. Create an initial population of P chromosomes.
2. Evaluate the fitness of each chromosome.
3. Choose $P/2$ parents from the current population via proportional selection.
4. Randomly select two parents to create offspring using the crossover operator.
5. Apply mutation operators for minor changes in the results.
6. Repeat Steps 4 and 5 until all parents are selected and mated.
7. Replace the old population of chromosomes with new ones.
8. Evaluate the fitness of each chromosome in the new population.
9. Terminate if the number of generations meets some upper bound; otherwise, go to Step 3.

2.4. Terminologies

2.4.1. Encoding

Before applying the genetic algorithm to any problem, a method is used to represent the chromosomes or the individual solutions so that the computer can process it. This representation method is called encoding. There are many approaches for encoding such as Binary Encoding where the sequence of 0's and 1's are used to represent the genes, Value Encoding where the sequence of values is used, Permutation Encoding where every chromosome is a string of numbers and Tree Encoding where every chromosome is a tree of objects or nodes. In our work here, binary encoding is used. After the genetic algorithm operators are applied finally the results are converted to the required format. This process is called Decoding.

2.4.2. Initial population generation

The genetic algorithm starts by initialization of population. The initial population is generated randomly by the algorithm. It will encode all possible solutions for the problem. The initial population can be of any size [7, 13].

2.4.3. Fitness evaluation

The fitness evaluation phase assigns a fitness value for each individual solution which is produced in the previous step. Based on the user requirement the fitness value is calculated. In our work we have used, $F = 1/f$ [8] where 'F' is the fitness value and 'f' is the total path length of the individual. The fitness value shows how fit the chromosome is. Here, the path length is calculated by using the distance formula which assumes Earth to be a perfect sphere.

2.4.4. Parent selection

After fitness value is assigned to each population, a fit parent, with the highest value, is selected as parents for further processing. Elitism method, Tournament selection methodRoulette Wheel method are some of the methods for parent selection [8]. Different selection methods select a different parent. The next step is crossover process.

2.4.5. Crossover

This is the important stage in genetic algorithm. The crossover phase takes two parents and combines them to produce a new child solution, known as offspring. The offspring is fed to the next generation. There are different ways for crossover to happen. It can be one point crossover, multipoint point crossover, ring crossover, and so on [14].

2.4.6. Mutation

This step introduces some changes to the solution so that it can generate new values to reduce duplicity of the solution and to produce better generations. Mutation can be done in many ways: flip mutation, inversion mutation, interchanging mutation, uniform mutation, and so on [15].

Table 1: Coordinates of places around Kathmandu Valley.

Place	Latitude	Longitude
Pashupatinath (A)	27.71067554	85.3489966
Bouddha (B)	27.72211403	85.3620749
Swayambhunath (C)	27.71510075	85.29036648
Dakshinkali (D)	27.60774938	85.26347831
Nagarkot (E)	27.717096	85.50414549
Chandragiri (G)	27.66015984	85.14549196
Bhaktapur (H)	27.67234037	85.42846706
NamoBuddha (I)	27.57158121	85.58180087
Phulchoki (J)	27.57524835	85.40048634
Switzerland Park (K)	27.71959370	85.24442133
Kirtipur (L)	27.66338125	85.27473867
Budanilakantha (M)	27.76476718	85.36312542

3. Conclusion

3.1. Data collection

The data of latitude and longitude of the places near Kathmandu Valley in Table. 1 was collected from Google [16].

3.2. Results

After plotting the latitude and longitude of the collected data points, the plot was found as in Fig.1:

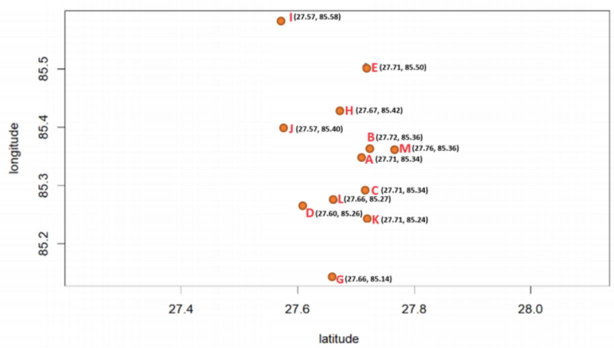

Figure 1: Plot of the latitudes and longitudes.

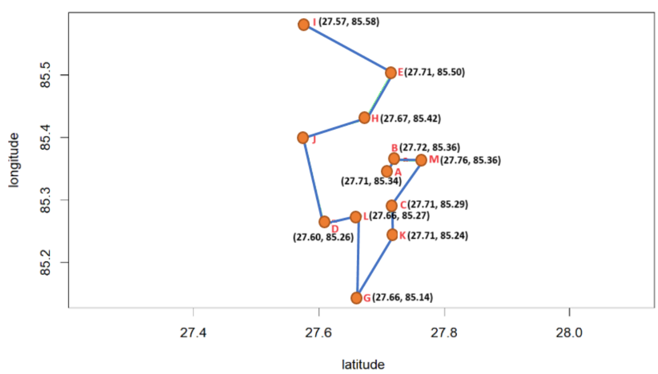

Figure 2: Output after 300 iterations.

The result of the first iteration was found to be $A \to C \to G \to K \to L \to D \to M \to B \to I \to J \to H \to E$, where the total distance traveled was 228.7 km. After deriving output of the implemented algorithm and recording the plot, the best shortest path is obtained as $A \to B \to M \to C \to K \to G \to L \to D \to J \to H \to E \to I$, as shown in Fig.2. Total distance is obtained as 221.5 km. The distance was reduced by 3.15%. Finally, the set of optimal solution was ensured by looking at the minimum distance traveled using this route.

In the similar study in 2017, Mohammed et al. [10], used GA to minimize the route within the university for two routes, and they succeeded in reducing the path to be traveled by 8.5%. They are however optimistic that this reduction can be more significant if longer routes are taken for a longer period of time.

References

[1] Rüschendorf L, Monge-kantorovich transportation problem and optimal couplings, *Jahresbericht der DMV*, 3 (2007) 113–137.

[2] Tolstoi A N, On the solution of the transportation problem, *Mathematics of Operations Research*.

[3] Holland J. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI (1962). https://doi.org/10.7551/mitpress/1090.001.0001.

[4] Larrañaga P, Kuijpers C M H, Murga R H, Inza I & Dizdarevic S, Genetic algorithms for the travelling salesman problem: A review of representations and operators, *Artificial Intelligence Review*, 13(2) (1999) 129–170. ISSN 0269-2821. https://doi.org/10.1023/A:1006529012972.

[5] Holland J H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT press (1992). ISBN 9780262275552. https://doi.org/10.7551/mitpress/1090.001.0001.

[6] Chien S, Yang Z & Hou E, Genetic algorithm approach for transit route planning and design, *Journal of Transportation Engineering*, 127(3) (2001) 200–207. https://doi.org/10.1061/(ASCE)0733-947X(2001)127:3(200).

[7] Herring R H & Eden M R. Evolutionary algorithm for de novo molecular design considering multi-dimensional constraints. In: Gernaey K V, Huusom J K & Gani R, eds., *12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*, vol. 37 of *Computer Aided Chemical Engineering*. Elsevier (2015), pp. 1433–1438. https://doi.org/10.1016/B978-0-444-63577-8.50084-X.

[8] Rao A & Hegde S K, Literature survey on travelling salesman problem using genetic algorithms, *International Journal of Advanced Research in Eduation Technology (IJARET)*, 2(1) (2015) 42. URL https://api.semanticscholar.org/CorpusID:212671404.

[9] Hussain A, Muhammad Y S, Nauman Sajid M, Hussain I, Mohamd Shoukry A & Gani S, Genetic algorithm for traveling salesman problem with modified cycle crossover operator, *Computational intelligence and neuroscience*, 2017(1) (2017) 1–7. ISSN 1687-5273. https://doi.org/10.1155/2017/7430125.

[10] Mohammed M A, Abd Ghani M K, Hamed R I, Mostafa S A, Ahmad M S & Ibrahim D A, Solving vehicle routing problem by using improved genetic algorithm for optimal solution, *Journal of computational science*, 21 (2017) 255–262. https://doi.org/10.1016/j.jocs.2017.04.003.

[11] Wang X, Wu X, Wang Z & Hu X. A model and an improved genetic algorithm for the vehicle routing problem with breakdown vehicles. In: *2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC)*. IEEE (2009), pp. 696–699. https://doi.org/10.1109/ICICIC.2009.28.

[12] Potvin J Y, Genetic algorithms for the traveling salesman problem, *Annals of Operations Research*, 63(3) (1996) 337–370. ISSN 1572-9338. https://doi.org/10.1007/bf02125403.

[13] Diaz-Gomez P A & Hougen D F. Initial population for genetic algorithms: A metric approach. In: *IEEE Games Entertainment Media Conference* (2007). URL https://api.semanticscholar.org/CorpusID:12013119.

[14] Hassanat A B & Alkafaween E, On enhancing genetic algorithms using new crossovers, *International Journal of Computer Applications in Technology*, 55(3) (2017) 202–212. ISSN 1741-5047. https://doi.org/10.1504/ijcat.2017.10005868.

[15] Soni N & Kumar T, Study of various mutation operators in genetic algorithms, *International Journal of Computer Science and Information Technologies*, 5(3) (2014) 4519–4521.

[16] Google LLC. Google maps (2024). URL https://www.google.com/maps/@27.7172446,85.3239595,15z, accessed: 2024-11-29.