Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 11, November-2021

Stadium Tour Planning and Management using Genetic Algorithm and Dynamic Programming

Kriti Chapagain
School Of Computer Science and Engineering
Vellore Institute of Technology, Vellore
India

Mitanshi Kshatriya
School of Computer Science and Engineering
Vellore Institute of Technology, Vellore
India

***Abstract*—The happening to web and empowering of downloading music, the music business is confronting a precarious decrease in CD sales which used membership-based revenue stream of music industry. The sales have endured to a greet extent because of illicit downloads and membership-based music administrations like Spotify, Ganna and so on. Presently, the primary wellspring of salary of music industry is touring. Because of poor logistics, blunder and poor route decisions numerous arena visits neglect to produce benefit. This task makes arena visiting a going great procedure and as practical as could reasonably be expected. The component of finding the most brief way will chop down the superfluous cost that are caused in transporting the stage equips just as the touring singer,band individuals, back up artists, reinforcement artists, foundation vocalists and other required staffs. This project has various modules that will help in betterment of logistics side of the tour. To generate shortest possible route consisting all required cities Genetic algorithm and dynamic programming approach is used. This problem concentrates on travelling salesman problem.**

Keywords— Traveling Salesman Problem, Genetic Algorithm, Dynamic Programming, Logistics, NP-hard

I. INTRODUCTION

This section is aimed at introducing the traveling salesman problem which we have used later to address and tackle the issues faced by the music industry in tour planning. We have also briefly introduced the two approaches that we have used for finding the optimal route. These approaches are genetic algorithms and dynamic programming. We have elaborated on the algorithm in the proposed methodology section.

A. Traveling Salesman Problem

TSP (Travelling Salesman Problem) is a classical NP-hard problem in the field of computer science and operations research and combinatorial optimization, The problem is to find the shortest path among the given cities, such that all cities are covered once in the path and return to the starting city. This problem can be applied to solve many real-world problems. Taking the instance of a courier company, they have to deliver a number of packages throughout the city, they can use TSP problem to find most optimal route thus saving time and fuel. Also while planning a route for drone flight, choosing an optimal route through the specified way-points can not only reduce time but may also extend the area that could be covered in every single flight. There can be other parameters needing optimization in each of those problems, but at the core, each can be viewed as solving a modification of TSP. It's an NP-hard problem in computational optimization important in research and theoretical technology. The amount of time required to solve this problem grows exponentially with the number of cities or inputs being increased. TSP is expressed as follows:

Let 1, 2, up to n be the marks of the n urban communities and $C = [ca,b]$ be an n x n cost lattice where each entry of the matrix represents the expense of heading out from the city a to b.

In TSP, we need to find an optimal route such that each city is visited only once and the last city visited should be the originating city.

The absolute cost of a route is given by:

$$A(n)= (i=1 \text{ to } n-1)\sum c_{a,a+1} + c_{1,n} \quad (1)$$

This paper introduces two ways to solve TSP-based on genetic algorithm and dynamic programming approach.

B. Genetic Algorithm

Genetic algorithms, presented by J. Holland (1975), are motivated from the Darwin advancement hypothesis: in the populace development, the best people, which are more adjusted to their condition, can outlast for quite a while, on the other hand, the people, which are not fits to their condition, vanish with the entry of ages. In this way, its chromosome and a proper wellness capacity to be characterized to assess people code every person. Initially, GA comprises to arbitrarily create introductory populace, at that point, genetic operators (selection, crossover, mutation), inside determined probabilities, are applied to deliver another generation that considered best than its previous versions.

The traveling salesman problem is considered as an optimization problem. It can be solved using a genetic algorithm since genetic algorithm is used for solving optimization problems.

C. Dynamic Programming

Dynamic programming is a useful computer programming method as well as a mathematical for solving optimization problems. It can be used to determine an optimal sequence of interrelated decisions. The main idea followed in this method's application is dividing of a process into several stages after which an optimal result is chosen for every stage. Dynamic programming is applied to problems recursive in nature. To save the computation cost of repeating recursive calls, the recursive calls are stored. If the recursive call's value is in store

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 11, November-2021

that value can be used right away. There are two fundamental approaches for solving TSP: precise and estimated. Precise outlooks are generally founded on Branch and Bound, Dynamic Programming, Integer Applied Mathematics. All of them can give optimal solutions for TSP.

However, the methods above give exponential time complexity i.e. the time taken by the methods increases exponentially with increasing time. The time complexity for the Dynamic Programming method is calculated to be O(n^2 .2^n). Thus, while solving TSP with Dynamic Programming, it can at best handle 40-60 cities or data points using branch and bound strategy. It might be able to fathom up to 200 cities or data points.

II. LITERATURE REVIEW

In the literature, researchers the joining of Genetic calculations (GAs) with hierarchical data sets to take care of the combinatorial issue in asset enhancement and the board. [1] The authors of the paper have proposed using two levels of knowledge - procedural and declarative. It will help in addressing the issues of combinational optimization as well as numerical functions. [2] The authors of this paper showed Genetic Algorithms can be integrated in any form of DSS or decision support system to ensure economical use of resources. TSP is detailed as discovering a change of n urban areas, which has the base expense. This issue is widely known as an NP-hard problem [2, 4, 5]. Numerous calculations have been proposed to tackle this issue [2, 3, 4, 5, 7, 10, 11, 12, 14, 15, 17]. There are two fundamental methodologies for addressing TSP: exact and approximate. Definite methodologies are typically founded on Dynamic Programming, Branch and Bound, Whole number Linear Programming… and all gave the ideal answers for TSP. However, it must be noted that the calculations which are using these methodologies have remarkable running time as M. Held and R. M. Karp [2] pointed out in their research work. Dynamic Programming takes O(n^2.2^n) running time. Thus, while solving TSP with Dynamic Programming, it can at best handle 40-60 cities or data points using branch and bound strategy. It might be able to fathom up to 200 cities or data points.

.[3] The authors of this paper have used tree decomposition with dynamic programming to further optimize the solution for TSP. Let there be a traveling salesman problem (TSP), the solution for it is tour H in graph G. An operation k-move will remove k-edges from H and then proceed to add k-edges of G so as to form a new tour namely H′. The well known k-OPT heuristic for TSP uses the k-moves method. It firstly finds a local minima by starting from an arbitrary tour H and subsequently improving it using a sequence of k-moves.

Various Computational Testing showed that new Ga chromosome approcach result in a smaller search space, furthermore, much of the time, delivers preferable arrangements over past procedures.[4] The MTSP is similar problem to the popular NP-hard traveling salesperson problem (TSP) with the added constraint that each city can be visited by any one from the sales team. Previous studies investigated solving the MTSP with genetic algorithms (GAs) using standard TSP chromosomes and operators. The above paper presents a new chromosome for GA and then proceeds to compare its performance in accordance with time and space complexities.

III. PROPOSED METHODOLOGY

Stadium Tour Management framework is an attempt to defeat the logistics issue confronted while leading an arena visit. It will decrease the odds of scattering of any stage hardware as a database can be utilized to cross-check if each gear is available or not. Our venture would be a great guide to the tour manager as he/she will have all the data pretty much every one of the necessities of the visit. All the data can be seen by the administrator.

The Stadium Tour Management framework has 4 modules:

1. Stadium
2. Staff
3. Equipment
4. Merchandise

All the modules have all common features of what is expected in a database management system like - create, read, update and delete along with some additional features.

A. *Database*

i. Stadium Database

	Sno	Sname	city	zipco
1	2	Ohio Stadium	Cambridge	43(
2	4	Ford Field	Liverpool	48'
3	3	Mercedes-Benz Stadium	Glasgow	63!
4	0	Levis Stadium	Brigton	95(
5	1	CenturyLink Field	Bristol	98'
6	7	Tesla Stadium	Oxford	98'

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	Sno	INTEGER							NULL
2	Sname	TEXT							NULL
3	city	TEXT							NULL
4	zipcode	INTEGER	🔑						NULL
5	capacity	INTEGER							NULL

Fig. 1. Stadium Database

ii. Equipment Database

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	equipID	INT							NULL
2	Ename	TEXT							NULL
3	Qty	INT							NULL

equipID	Ename	Qty
1	Guitar	5
2	Drums	1
3	Mike	20

Fig. 2. Equipment Database

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 11, November-2021

iii. Staff Database

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
stid	INT							NULL
Sname	TEXT							NULL
job	TEXT							NULL

stid	Sname	Job
1234	melissa	Background Singer
4591	stella	Background Singer
5609	bob	technician
8905	joel	guitarist

Fig. 3. Staff Database

iv. Merchandise Database

Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
Type	CHAR (20)							NULL
merch_id	INTEGER							NULL
quantity	INTEGER							NULL
price	DOUBLE							NULL

Type	merch_id	quantity	price
Tshirt	1	10000	12.99
Hoodie	2	5000	24.99
CD	3	5500	10.99
Bands	4	20000	3.99

v. Distance Matrix Table

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate
1	Sno	INTEGER						
2	name	TEXT (25)						
3	city0	DOUBLE						
4	city1	DOUBLE						
5	city2	DOUBLE						
6	city3	DOUBLE						
7	city4	DOUBLE						
8	city5	DOUBLE						

	Sno	name	city0	city1	city2	city3	city4	city5
1	0	Brighton	0	172	145	607	329	7
2	1	Bristol	172	0	192	494	209	15
3	2	Cambridge	145	192	0	490	237	7
4	3	Glasgow	607	494	490	0	286	54
5	4	Liverpool	329	209	237	286	0	42
6	5	London	72	158	75	545	421	

Fig. 4. Distance Matrix Table

B. *Algorithms used*

- Generating the shortest route covering all cities is equivalent to traveling salesman problem. For this two approaches have been used-
- 1. Genetic algorithm
- 2. Dynamic programming
- The distance matrix is generated with the help of a table containing the distance of one city to other. Distance is zero for the same city.

Genetic algorithm:

i. Creating distance matrix – It is a2D array where the entry at i, j represents the distance from the ith city to the jth city.
ii. Creating population – Population is a collection of possible routes.
iii. Determining the fitness of the population-
iv. Select the mating pool
v. Breed/ordered crossover
vi. Mutate
vii. Repeat the above process until the required number of iterations is reached. We are using 2000, 4000, 5000 iterations.

Dynamic programming:
1. Cost ({1}, 1) = 0
2. for s = 2 to n do
2.1 for all subsets S Є [1, 2, 3, … , n] of size s and containing 1
2.1.1 C (S, 1) = ∞
2.2 for all j Є S and j ≠ 1
2.2.1 Cost(S, j) = min [Cost (S – {j}, i) + d(i, j) for i Є S and i ≠ j]
3. Return min j Cost ({1, 2, 3, …, n}, j) + d(j, i)

IV. TESTING AND RESULTS

We used DP and GA contrast to create minimum spanning path. The calculations were applied of distance matrix of eight cities. Genetic algorithm seem to discover great answers for the Traveling Salesman Problem, anyway it lays particularly on how it is encoded and which crossover and mutation strategies are utilized. Likewise, it does not have one optimal solution. However, it provides with best possible solution. Dynamic programming approach is relatively simple to code yet it requires some investment when the amount of data increments .Additionally, a beginning city must be indicated in its code. It is expensive for memory as well as time. The quality of solution depends on the code.

Comparison of algorithms used -

TABLE I. RESULTS

Algorithm	Time Taken	Result(cost)	Time Complexity	Advantage	Disadvantage
Genetic Algorithm	77.041	1355.0	O(Kmn)	Best possible solution is generated	No optimal solution is reached
Dynamic programming	0.2513	1355.0	O(n2 *2n)	Generates an optimal solution Easy to code	Expensive for memory as well as time Quality of solution depends on the code

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 11, November-2021

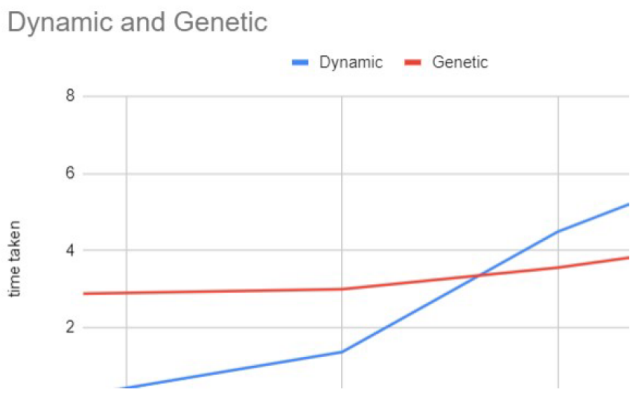

Fig. 5. Time taken by GA and DP approaches with various amount of data

The accompanying chart shows that with less amount of data dynamic programming give a lot quicker outcome than the hereditary calculation for tackling traveling salesman problem. However, as the size of information begins to expand the time taken by powerful programming builds exponentially when contrasted with that of genetic calculation.

The exhibitions of the two calculations are unequivocally influenced by their parameters just as their applications. GA is skilled at looking for ideal arrangements in low input cases. Now and again, for example, with various input greater than 100, the presentation of GA is better than that of DP. In any case, for a lower amount of data GA takes additional time than DP.(DP has a decent presentation in acquiring steady and top-notch solutions, even in looking through huge search space. Be that as it may, because of its exhaustive enumerating nature, it costs an abundance of time in a huge amount of data cases.

As of late, touring has become the significant wellspring of salary for the music industry. Arena visits the executives' framework makes the logistics part of touring

simpler and increasingly practical. It will decrease any utilization of paper in keeping up information about visits.

V. CONCLUSION

In sum up, this research contains a function for generating the shortest-path spanning all the cities. This will help in covering all cities in a much smaller total distance. Anyway, this venture doesn't mull over a few things like the different levies forced while traveling interstate or the accessibility of arenas on required dates, and so on. With everything taken into account whole undertaking may come convenient for touring chiefs.

Tools used: The project is implemented using python 3 and sqlite3 packages for the database.

REFERENCES

[1] UGWU, O.O. and TAH, J.H.M. (2002), "Development and application of a hybrid genetic algorithm for resource optimization and management", Engineering, Construction, and Architectural Management, Vol. 9 No. 4, pp. 304-317. https://doi.org/10.1108/eb021225

[2] Held, M., Karp, R.M.: A dynamic programming approach to sequencing problems. Journal of the Society for Industrial and Applied Mathematics. vol. 10, pp. 196--210 (1962)

[3] Marek Cygan, Lukasz Kowalik, Arkadiusz Socala “Improving TSP Tours Using Dynamic Programming over Tree Decompositions”, ACM Transactions on AlgorithmsVolume 15Issue 4 October 2019 Article No.: 54pp 1–19.

[4] Arthur E. Carter, Cliff T. Ragsdale, “A new approach to solving the multiple traveling salesperson problem using genetic algorithms” European Journal of Operational Research Volume 175, Issue 1, 16 November 2006, Pages 246-257

[5] Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2006). The traveling salesman problem: a computational study.

[6] Stoltz, E. (2018, July 17). Retrieved from https://towardsdatascience.com/evolution-of-a-salesman-a-complete-gen etic-algorithm-tutorial-for-python-6fe5d2b3ca35

[7] Wei, J.-D. (2008). Approaches to the Travelling Salesman Problem Using Evolutionary Computing Algorithms. Traveling Salesman Problem. doi: 10.5772/5584

[8] Al-Dallal, A. (2015). Using Genetic Algorithm with Combinational Crossover to Solve Travelling Salesman Problem. Proceedings of the 7th International Joint Conference on Computational Intelligence. doi: 10.5220/0005590201490156