The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVI-4/W5-2021
The 6th International Conference on Smart City Applications, 27–29 October 2021, Karabuk University, Virtual Safranbolu, Turkey

OPTIMIZING BUS LINES USING GENETIC ALGORITHM FOR PUBLIC TRANSPORTATION

B. Bulut[a, *], M. Gunay[a], K. Ozgun[b], J. W. Ledet[a]

[a] Akdeniz University, Computer Engineering, Antalya, Turkey - bthn.bulut@gmail.com, mgunay@akdeniz.edu.tr, josephledet@akdeniz.edu.tr
[b] Dept. of Industrial Engineering, Antalya Bilim University, Dosemealti, Antalya, Turkey - kamer.ozgun@antalya.edu.tr

Commission VI, WG IV/1

KEY WORDS: Public Transportation, Genetic Algorithm, KMeans Clustering, Transportation Network Design

ABSTRACT:

Due to increasing human population, the need for quality public transportation has also increased. This study takes stop density, stop layout, and passenger population of those stops into consideration to offer a better regulated public transportation network design that can satisfy the increased demand. In this study, the boarding data is provided by the public transportation department of the city of Antalya, Turkey. Remaining required data was automatically generated using web services and stored in a PostgreSQL database hosted on a cloud server. After visualizing inputs such as bus routes, stop layout, and passenger density on Google Maps and KeplerGL, with the use of the K-Means algorithm, data was clustered to find "hot" (i.e. attraction) areas on a macro scale. A novel means of connecting hot spots suggested by the outcome of the Genetic Algorithm was developed. To compare the effectiveness of the proposed approach with the existing network, current bus stops were mapped to the new domain. It was observed that a more efficient system was achieved by higher route efficiency and lower transfer counts.

1. INTRODUCTION

With changing urban infrastructure, public transport system designs must be updated accordingly. Where urban renewal is concerned, updating the system becomes necessary due to the technology and population changes that result as an area becomes a residential area (Özgün et al., 2021b). In such cases, adding and subtracting can be done by hand in some systems. This situation may bring the necessity of optimizing the network. Advanced optimization may be applied to automate this process effectively.

The aim of this paper is improving Antalya's public transportation efficiency by redesigning routes. The constraints are route quantity (directly related with bus quantity) which should be less than the currently active system, node coverage and transfer necessity when traveling between two points. In the process of achieving this goal, KMeans and a genetic algorithm were used.

1.1 Related Work

In context of public transport systems, the transit network design problem (TNDP) aims to find a set of optimal routes regarding the costs to the users and the operator. Since the quality of service highly depends on result of this design, TNPD, which is an NP-hard problem, has been studied for the last five decades (Yang and Yangsheng, 2020). While users expect cheap and direct transfers, comfortable terminals and vehicles, and frequent bus arrivals, the objectives of the operator is having a system that maximizes profit (Guihaire and Jin-Kao, 2008). Most of the existing research exclusively prioritizes the objectives of the operator.

In the current literature, it can be seen that the process of transit planning are referred to by a variety of names even if the problem itself is not explicitly named (Guihaire and Jin-Kao, 2008).

Depending on the nature of the formalization, studies optimize various parameters (Magnanti and Wong, 1984). In a recent study (Crainic, 2000), authors discuss various state-of-art integer programming solutions to transit network design. A branch and cut algorithm to solve the incapacitated fixed charge network flow problem was suggested (Ortega and Wolsey, 2003).

In another approach (Mandl, 1980), passengers and vehicles are assigned to the routes as two different problems and the optimal path in a graph network is sought. As a constraint, the number of vehicles is fixed. The next step was to find the optimum routes and vehicle distribution for those routes, while minimizing the travel costs for passengers.

Researchers (Aguado, 2008) used a Lagrangian relaxation based heuristic to solve the fixed charge transportation network design problem. The author divides the process into three phases:

1. Apply Lagrangian relaxation to get reduced costs of all variables
2. Run binary search algorithm to obtain the heuristic
3. Branch and Cut

Using an algorithm based on Kuhn-Tucker conditions, Furth (Furth and Wilson, 1981) devised a method to optimize the allocation of buses to routes by maximizing the net social benefit. Researchers (Constantin and Florian, 1995) formulated a mixed integer programming model to optimize bus frequency to minimize passenger travel and waiting time. Due the high time-complexity of brute-force type search algorithms, there are other studies that attempt to optimize the network design using Genetic Algorithms (Al-Turjman et al., 2016, Al-Turjman et al., 2017). Another study (Pattnaik et al., 1998) proposed a genetic algorithm to solve transit routes and frequency optimization problems by minimizing both the operator costs and travel time of users. Researchers (Claessens et al., 1998) developed a programming model that minimizes operating costs subject to services and capacity requirements to solve the optimal rail route problem.

*Corresponding author. Melih Gunay, mgunay@akdeniz.edu.tr

Cost effective transit planning is usually divided into five elements (Guihaire and Jin-Kao, 2008, Ceder and Wilson, 1986): Route design, frequency setting, timetabling, bus scheduling, crew scheduling (Ozgun et al., 2021a, Başaran et al., 2021). The first two of these elements are considered as mathematical programming problems which can be solved with simulations and/or heuristic algorithms. However, in practice these are generally solved with intuition and experience of a local expert (Deng and Yan, 2019).

Different authors used different approaches when designing the Genetic Algorithm's chromosome. Some used genes in a chromosome as representation of expansion of a link in a network and denoted by 1 or 0 (Ukkusuri et al., 2007). Other studies represent every gene as a line. However, they all attempt to optimize the operational costs. On the other hand there is a major demand by the riders of the transportation network as they want both direct lines between the origin and the target stops and also reaching the target with few transfers as possible. Therefore, in this work we did not only optimize the operational costs but also consider the passengers preferences.

2. MATERIALS AND METHOD

A line represents a specific bus label while a route represents either the forward or backward routes of a line separately. Generally every line has two routes for this reason.

The data including line and route information was obtained as excel files. In order to increase the compatibility between all platforms, files were converted to csv files and logically merged with consistency checks. After saving, the file was analyzed on a Colab Notebook.

2.1 Sample Line and Route Data

Table 1: Line and Route Relation Samples

Line ID	Line Code	Route ID	Direction
480	CV48	4800	0
480	CV48	4801	1
670	CV67	6701	1
670	CV67	6700	0
...	...	...	...
13090	DC09	130901	1
13090	DC09	130900	0
13100	DC10	131001	1
13100	DC10	131000	0

Route code uniqueness was checked with a python script and it was assumed that every specific route should have only one match in the lines list. As a result 609 unique routes were found without a repetitive route code.

Line and route relation examples can be seen in Table 1. The line code represents a line's database id. The line code representation is what is shown to public as labels on busses and direction id represents whether it is the forward or return route.

2.2 Data Management

Remaining necessary data was harvested from the API for use with the provided data.

2.2.1 Exploration of Service API Users of this service can view routes of a specific line, its departure times and stops.

The methods of using the features offered by the portal and the authorization method have been debugged by following the HTTP requests and responses with the tools offered by the browser. Information requests can be sent to the service with the authorization code. In this study, the line code information displayed to the public transportation users and the information that there is a departure or return route, the endpoint, where the route and stop locations of that line are taken, was used to collect data. For example, when we query the bus known with the code VF01, response is quite long (2500 lines and remainder). For this reason, a brief summary of the returned content is shared below:

- Information on whether the transaction was completed successfully
- Line code information not shown to users
- The explanation of line code shown to users (VARSAK ALTIAYAK - FAKULTE)
- Array of points expressing the route when combined with each other with sequence information (in geographic coordinate system)
- Departure times depending on the days of the week of the vehicles running on that line

2.2.2 Database Design Before the data was collected, it was determined that it would be placed in a database for the following reasons; to be accessible quickly and easily, to be open to others, and to increase its sustainability. Accordingly, the database was designed to comply with the Fourth Normal Form (4NF) rules. During this design, the fields that are returned from the service's API and which are not clear are not used in accordance with the 4NF rules (Kent, 1983).

As a database, PostgreSQL, which is popular among open source databases, was chosen for its performance, reliability and community (Andjelic et al., 2008). In the future, PostgreSQL can respond to these within the scope of a NoSQL structure requirement.

The designed structure was converted into models with .NET Core SDK in C# language and converted into a database with the Code First technique. Integration of PostgreSQL and Entity Framework Core, which is the ORM tool used by .NET Core, is provided with Npgsql provider.

2.2.3 Data Gathering Information requests are sent to the service with a .NET Core CLI application that includes all combinations of line codes and going / return parameters written in the database previously provided and created, and the returned responses are parsed.

2.2.4 Importing Boarding Data The json file was read and processed into the database including the boarding data provided by the service provider. During this process, logging has also been implemented and is not included in the database in order to observe that some boardings are not from a registered stop or a registered line code. If the data comes from a known stop of a known line but its route is not clear, it is included in the Orphan-Boarding table.

2.3 Clustering Boarding Data

The centers of regional densities must be identified in order to propose a new transport. The number of these centers and their distances from each other and the number of boardings of these centers are the parameters that play an important role in this determination.

KMeans algorithm was applied to create these regions. The technique that would create the most successful result was selected by examining the results.

The elbow method was applied to find the correct cluster number. In order to do this, the KMeans algorithm was trained with the same data from 2 clusters to 30 clusters. The inertia value of each model was plotted based on the corresponding k value. This method was determined to be inadequate.

In determining the k value of KMeans, a new method was developed due to the inadequacy of the standard elbow method in terms of reaching sufficient resolution. This method examines the distance between the centers of gravity and geometric centers of clusters. The number of clusters was increased to the point where the centers of gravity and geometric centers of the clusters stopped converging significantly. The number of clusters was found when it came to the point where the geometric and gravity centers did not become closer significantly even if k increased. Here the weighted center phenomenon comes from different weights of the elements in a cluster. As a result, 24 clusters obtained as can be seen in Fig 1 with letter characters (A, B,..).

2.4 Genetic Algorithm

A route connects only two nodes. A public transportation system needs more than one route to cover a target area. There are 24 nodes and permutations of those nodes with 3, 4 and 5 are shown below:

- P(24, 3) = 12.144 routes
- P(24, 4) = 255.024 routes
- P(24, 5) = 5.100.480 routes

Selecting efficient routes to achieve full coverage while considering the negative effect of overlapping paths of selected routes is computationally expensive due to the large number of possible route options. Including an efficient route to the solution can make another efficient route much less efficient because of the existence of the route that is already included. Since selecting routes is affected from prior selections, this problem is familiar with knapsack problem (Chu and Beasley, 1998). For this reason, a Genetic Algorithm is preferred.

The routes outside of the 10km-30km range were eliminated. Remaining routes were examined programmatically to ensure that each node is found at least once in the problem space.

The chromosome design is represented in Table 2. The route column lists the remaining 17,145 permutations after elimination. A chromosome represents a solution and consists of selected routes in this solution. When creating the first population, the maximum number of active genes in the chromosomes was limited to 1/3 of the total number of nodes which is 24.

The new efficiency function, E, is defined as below:

Table 2: Chromosome Design

Route	Is selected
ABC	1
ACB	0
...	...
ZYXTA	0
XZYAC	1

$$E = f(x) \times \frac{\sum_{i=0}^{N} node_i potential}{\sum_{i=0}^{N-1} distance(node_i, node_{i+1})} \quad (1)$$

where:

$$f(x) = \begin{cases} y, & y > 0.8 \\ 0.9y, & 0.7 < y \leq 0.8 \\ 0.8y, & 0.6 < y \leq 0.7 \\ 0.6y, & 0.5 < y \leq 0.6 \\ 0.2y, & y \leq 0.5 \end{cases}$$

and

$$y = \frac{X_{euclidean}}{\sum_{i=0}^{N-1} distance(node_i, node_{i+1})}$$

The $node$ parameter represents the nodes of a specific route. The formula represents the node potentials to find the route potential. Then divides it by the total distance which is the sum of the road lengths between sequential route nodes. Then we multiply by the directness factor which is calculated by dividing euclidean distance between the route start node to the route end node (X) with an easing function (f) multiplication by total distance.

The fitness function sums up the E score of each active route within the genome. If there is a repetitive edge use in same direction between 2 nodes in the route suggestions, there is a 1 point penalty from the fitness score for each encounter. 10 points are penalized for each node that is not included in solution due to full coverage constraint.

A selection function is defined to select the candidates to crossover from among the population. Each time this function is called, it randomly selects two parents from the population. During this random selection, the probability of being selected was defined in direct proportion to their fitness scores.

The crossover function describes the way the crossover process between 2 genomes is performed. The minimum active index and the maximum active indexes were found for 2 genomes and random numbers were generated in this range. From the point of this random number generated, the genomes were split in two and each piece was combined to correspond with the other partner. Thus, two new genomes were produced. Here, instead of directly producing value between zero and chromosome number for random selection, the process of finding the minimum and maximum value range is that the ratio of active chromosome number to total number of genomes is very small. This creates the possibility that all active genes remain to the right or left of the chosen point when the range is not determined.

The identified mutation function has a %50 probability of logically inverting a random gene. It's performed this experiment three times in a row.

The algorithm has been tested with different parameters and the values have been adjusted. The population member count is initially set to 30. At first, the algorithm is set to stop after 100 generations if it cannot exceed the desired fitness value. But a

patience counter was implemented which increases when the solution score is equal to previous and resets when it is greater. This was a better approach since it stops iterations when no improvement was seen.

2.5 Domain Migration of Current System and Analysis

The routes of the existing system were converted into routes in the proposed system domain. To do this, every stop in the current system was included to the nearest node in the proposed system. Converted routes with less than 3 nodes were eliminated because of external connections outside of the problem boundary. For the forward direction there are 249 lines which consist of 117 unique converted routes. After the elimination filter was applied 125 lines were left with 98 unique converted routes. Efficiency, directness and transfer counts of active and the proposed system were compared. An algorithm was proposed for adding extra routes to the proposed system in need of decreasing transfer counts.

3. RESULTS AND COMPARISON

After 75 generations, the max score of the run was reached and no further improvement was observed on next generations until patience ran out. The solution route scores ranged from 0.16 to 15.22. Routes with low scores were eliminated until the point that the full node coverage rule was not broken. Node names and coverage can be seen in Fig. 1.

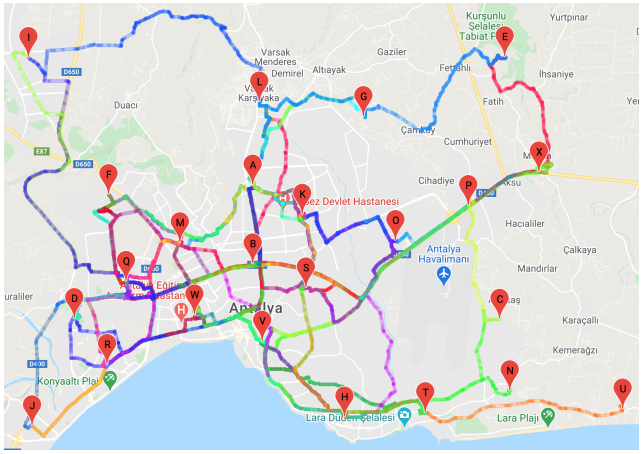

Figure 1: Nodes with All Selected Routes

Routes are drawn in different colors in order to distinguish them from each other. The colors of the routes using common roads are mixed due to the existence of many routes, the routes cannot be clearly distinguished from this figure.

Top 5 efficient routes in the solution are shown in Table 3. Boarding column represents total boarding potential of nodes in that route. Duration data was obtained using Google Maps API.

Table 3: First 5 Genetic algorithm generated routes

Route	Boarding	Length	Duration	Score	Stops
HVWQF	569984	20.18	46.67	15.23	817
XPBWJ	417525	28.70	41.33	12.05	574
XPBWQ	431656	24.48	33.27	11.62	677
LABRJ	301584	21.84	38.45	9.42	718
UHVBQ	402299	29.84	56.13	8.77	671

In this solution, the total number of unique edges is 105 with the total count of 200. Thus the average edge reuse amount is 1.90.

Top 5 most used edges in routes in the solution are seen in Table 4. These edges connect high potential nodes.

Table 4: Edge Use Count In Routes

Edge	Use Count
WQ	12
QB	8
BW	6
QD	5
BQ	5

Route repetition ratio for the currently active system was found to be 0.216 from the $R_r = \frac{R_t - R_u}{R_t}$ formula where R_r is route repetition ratio, R_t is total route count and R_u is unique route count.

Directness improvement was found to be 10% from the $I = \frac{D_1 - D_0}{D_0}$ formula where I is improvement, D_1 is directness value of proposed system and D_0 is directness value of currently active system.

The currently active system repeats similar routes and is not efficient. In Fig. 2 we can see the drastic efficiency drop of active system compared to suggested when edge repeat penalty increases. System efficiency corresponds to sum of fitness values of routes in solution.

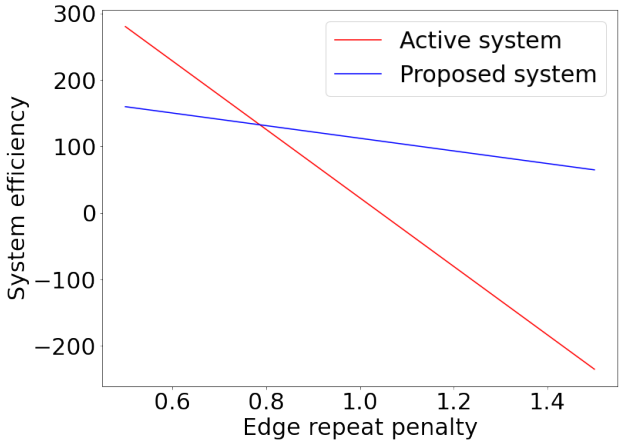

Figure 2: Edge Repeat Penalty - Efficiency Relation

Transfer counts were measured for both systems by counting every node pair that is not included in any route in the system. Analysis results shows that the transfer amount in the active system is 42 while it is 112 in proposed system. Which is expected since the fitness function does not have the information related to the importance of transfer. Edges that need to transfer to travel shown in Table 5.

Those from-to edges in the same row in Table 5, connected to each other with permutations of 5 (3 or 4 if row doesn't have minimum 5 nodes) and selectively appended into the solution to decrease transfer count in the proposed system. 915,726 candidate routes were generated. Route efficiency values were calculated for every generated route and ordered by efficiency, descending. These routes were evaluated iteratively. All covered nodes were kept in a hashset during iteration. If the next route did not have enough nodes that were not covered, it was discarded.

Accepting routes that did not have any nodes covered resulted in 14 new routes: COSRJ, KVWRJ, UTHVW, OVWRJ, POSVJ, JRDML, NTHVW, EOVWR, XPOVR, JRMLG, HVWFI, DVTNU,

Table 5: Transfer Necessity

	FROM	TO NODES
0	A	C, N, U
1	B	C, N
2	C	D, E, F, G, I, J, K, L, M, N, O, R, S, U, X
3	D	E, G, J, L, N, T, U, V
4	E	F, H, I, J, K, N, O, Q, R, T, U, V, W
5	F	G, I, J, K, L, N, P, U
6	G	H, I, J, N, O, P, S, T, U, V, X
7	H	I, J, P, R, X
8	I	J, M, N, P, S, T, U, X
9	J	K, M, N, O, T, U
10	K	N, P, T, U
11	L	N, P, T, U, X
12	M	N, U, X
13	N	O, P, R, T, U, V, X
14	O	P, T, U
15	P	T, U
16	R	T, U, X
17	S	T, U
18	T	U, X
19	U	W, X

THVWR, THVWF. When those routes were added to the proposed system the total route count increased from 50 to 64 and transfer count decreased from 112 to 83. Accepting routes that had maximum 1 covered node resulted in 39 new routes: QKTNU, CGLMF, JQKTN, THVSG, XOKSR, POSVJ, JRDML, NTHVW, UNHKQ, QOTNU, DVTNU, COSRJ, LVTNU, IFQRJ, EOVWR, XPOVR, UNOKM, JRMLG, JRSXU, JDMLE, HSOPX, UTHVW, OVWRJ, XOKSD, UNPGL, MKTNU, EGLFD, HVWFI, THVWR, THVWF, UNTHS, IFQOT, KVWRJ, JDMLG, XSKMR, JRDMI, XOKMF, NOKMF, MSTNU. When those routes were added to the proposed system total route count increased from 50 to 89 and transfer count decreased from 112 to 36. Adding the first 25 of these 39 increased from 50 to 75 and transfer count decreased from 112 to 46. The transfer count of the active system is 42 and total length of active system is 3428 km. These values can be seen in Table 6.

Table 6: Transfer and Length Comparison

Proposed System Transfer Count	Proposed System Route Count	Proposed System Total Length
112	50 +0	1296 km
83	50 +14	1651 km
46	50 +25	2047 km
36	50 +39	2440 km

Finally, active system entities were migrated to the new system domain for comparison. Comparisons were made using different aspects such as route amount, transfer amount and efficiency. Transfer count was adjusted by adding new efficient routes that eliminates some necessary transfers. Additional side routes can be appended to the evolutionary solution manually. For better genetic algorithm solutions we must parameterize efficiency function coefficients and fitness function constants and then execute hyperparameter optimization to tune the algorithm. Adaptive mutation mechanics may help to escape from local maxima which leads the algorithm to stop after the patience limit is exceeded.

When a clear result cannot be obtained from the elbow method during clustering with KMeans, a method that will be successful on spatial data was proposed in order to find the k value, which is the maximum number of significant clusters. In this method, the average distance between the cluster center of all clusters and the center of gravity of the cluster was calculated. For a set of k values, this distance was calculated, the sorted chart was drawn, and the k value at the beginning of the line where the value converge was taken. This can represent the maximum number of significant classes.

4. CONCLUSION

This study proposes a method to optimize the type of public transport by balancing maximum passenger transport with minimum distance travel with shared edge restriction between routes. An efficiency function was defined so that the number of passengers taken in heuristic sense is rewarded while the distance covered is penalized. Route directness calculated and used as a parameter in the efficiency function. This efficiency function was used in the genetic algorithm's fitness function to obtain results. Since transfer amount is not considered as a negative effect, it was not included in the efficiency function. A new system with new routes were proposed using a genetic algorithm.

Unlike other studies, we located attraction/population centers via clustering and connected these centers with routes using a genetic algorithm. Every gene represents a possible route in a chromosome of all network routes. Consequently, this causes the chromosome to be larger than in other studies. However, negative effects are regulated by the optimization of the fitness function using cache and heuristics.

ACKNOWLEDGEMENTS

The authors thank Antalya Munuplicity for providing valuable data and insight.

REFERENCES

Aguado, S., 2008. Fixed charge transportation problems: a new heuristic approach based on lagrangean relaxation and the solving of core problems. Annals of Operations Research.

Al-Turjman, F., Karakoc, M. and Gunay, M., 2017. Path planning for mobile dcs in future cities. Annals of Telecommunications 72, pp. 119–129.

Al-Turjman, F., Karakoc, M., Gunay, M. and Noureldin, A., 2016. Routing mobile data couriers in smart-cities. In: 2016 IEEE International Conference on Communications (ICC), pp. 1–6.

Andjelic, S., Obradovic, S. and Gacesa, B., 2008. A performance analysis of the dbms - mysql vs postgresql. 10, pp. 53–57.

Başaran, B. D., Özgün, K. and Günay, M., 2021. Boarding pattern classification with time series clustering. In: Lecture Notes on Data Engineering and Communications Technologies, Springer International Publishing.

Ceder, A. and Wilson, N. H., 1986. Bus network design. Transportation Research Part B: Methodological 20(4), pp. 331–344.

Chu, P. C. and Beasley, J. E., 1998. A genetic algorithm for the multidimensional knapsack problem. Journal of heuristics 4(1), pp. 63–86.

Claessens, M., van Dijk, N. and Zwaneveld, P., 1998. Cost optimal allocation of rail passenger lines. European Journal of Operational Research 110(3), pp. 474–489.

Constantin, I. and Florian, M., 1995. Optimizing frequencies in a transit network: a nonlinear bi-level programming approach. International Transactions in Operational Research 2(2), pp. 149–164.

Crainic, T. G., 2000. Service network design in freight transportation. European Journal of Operational Research 122(2), pp. 272–288.

Deng, Y. and Yan, Y., 2019. Evaluating route and frequency design of bus lines based on data envelopment analysis with network epsilon-based measures. Journal of Advanced Transportation pp. 1–12.

Furth, P. G. and Wilson, N., 1981. Setting frequencies on bus routes: Theory and practice. Transportation Research Record.

Guihaire, V. and Jin-Kao, H., 2008. Transit network design and scheduling: A global review. Transportation Research Part A: Policy and Practice 42(10), pp. 1251–1273.

Kent, W., 1983. A simple guide to five normal forms in relational database theory. Commun. ACM 26(2), pp. 120–125.

Magnanti, T. L. and Wong, R. T., 1984. Network design and transportation planning: Models and algorithms. Transportation Science 18(1), pp. 1–55.

Mandl, C. E., 1980. Evaluation and optimization of urban public transportation networks. European Journal of Operational Research 5(6), pp. 396–404.

Ortega, F. and Wolsey, L. A., 2003. A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem. Networks 41(3), pp. 143–158.

Ozgun, K., Basaran, B. D. and Gunay, M., 2021a. Determination of peak times in public transportation. In: Innovations in Intelligent Systems and Applications Conference (ASYU).

Özgün, K., Günay, M., Başaran, B. D., Bulut, B., Yürüten, E., Baysan, F. and Kalemsiz, M., 2021b. Analysis of public transportation for efficiency. In: J. Hemanth, T. Yigit, B. Patrut and A. Angelopoulou (eds), Trends in Data Engineering Methods for Intelligent Systems, Springer International Publishing, Cham, pp. 680–695.

Pattnaik, S. B., Mohan, S. and Tom, V. M., 1998. Urban bus transit route network design using genetic algorithm. Journal of Transportation Engineering 124(4), pp. 368–375.

Ukkusuri, S. V., Mathew, T. V. and Waller, S. T., 2007. Robust transportation network design under demand uncertainty. Computer-Aided Civil and Infrastructure Engineering 22(1), pp. 6–18.

Yang, J. and Yangsheng, J., 2020. Application of modified nsga-ii to the transit network design problem. Journal of Advanced Transportation.