공학석사학위논문

A Simulation Study of Demand Responsive Transport for the Disabled to Minimize User Waiting Time

장애인콜택시 이용자 대기시간 최소화를 위한 시뮬레이션 연구

2018년 2월

서울대학교 대학원
건설환경공학부
가 은 한

Abstract

A Simulation Study of Demand Responsive Transport for the Disabled to Minimize User Waiting Time

Ka, Eunhan
Department of Civil and Environmental Engineering
The Graduate School
Seoul National University

Demand Responsive Transport (DRT) is widely used as a door-to-door service for people with disabilities (persons with disabilities, elderly people, pregnant women, etc.). The mobility impaired have limited or difficulty in using public transportation for people who are uncomfortable to move. DRT is a suitable means of providing customized services for the mobility impaired in that it can support the mobility service according to the user's request by picking up the user at the desired place and at the desired pick-up time. In this study, we developed a simulation model to study how to minimize user waiting time, which is one of the indices of service level, as a way to improve the service level of DRT for the mobility impaired. Moreover, simulation is useful for evaluating various operating methods, and since DRT has a large problem size, simulation is developed rather than a mathematical model. Using the real data of DRT for the disabled in Seoul Metropolitan City in 2016, a simulation model was tested and calibrated to find the optimal matching time and evaluated the effectiveness compared with the existing operation method.

Keywords: Demand Responsive Transport (DRT), The Disabled, Simulation, Dial-a-Ride Problem, Performance Evaluation

Student Number: 2016-21238

List of Contents

Chapter 1. Introduction .... - 1 -

1.1. Backgrounds ....- 1 -

1.2. Research Objectives ....- 5 -

Chapter 2. Literature Review .... - 6 -

Chapter 3. Methodology .... - 10 -

3.1 Problem Definition ....- 10 -

3.2 Model Formulation ....- 11 -

Chapter 4. Data Collection and Results .... - 18 -

4.1 Data collection ....- 18 -

4.2 Results ....- 23 -

Chapter 5. Conclusion .... - 27 -

요약(국문 초록) .... - 32 -

List of Tables

Table 1 Operation status of DRT for the disabled in Korea .... - 2 -
Table 2 Literature Reviews of Dial-a-Ride Problem (revised from Cordeau et al. (2007)) .... - 7 -
Table 3 Data Attributes of DRT for the Disabled in Seoul .... - 19 -
Table 4 Parameters Used in Simulation... - 21 -
Table 5 Results of Average Waiting time by Matching Time .... - 24 -

List of Figures

Figure 1 Satisfaction Survey of DRT for the Disabled in Seoul (2015) .... - 3 -
Figure 2 Relationship between Matching Time and Waiting Time .... - 4 -
Figure 3 Types of Dial-a-Ride Problem (DARP) .... - 6 -
Figure 4 Availability of Vehicle Types by User Types .... - 10 -
Figure 5 The Procedure of DRT for the Disabled... - 11 -
Figure 6 Flow Chart of Simulation (revised from Law (2007)) .... - 15 -
Figure 7 Part of Simulation Code Implemented in Java............................... - 16 -
Figure 8 Example of Match Procedure... - 17 -
Figure 9 Vehicles Types of DRT for the Disabled in Seoul.......................... - 18 -
Figure 10 Map of Operational Area.. - 20 -
Figure 11 Results of Calibration.. - 22 -
Figure 12 Plot of Average Waiting Time by Matching Time........................ - 23 -
Figure 13 Results of Each Type for Match.. - 25 -

Chapter 1. Introduction

1.1. Backgrounds

Korea has implemented various policies for persons with disabilities, elderly persons, pregnant women, infants, and infants and children who are defined as the mobility impaired through the Mobility Enhancement for the Mobility Impaired Act. Special Means of Transportation (SMT) is used to provide door-to-door service for the disabled who are physically difficult to move and is classified as Demand Responsive Transport (DRT). The definition of DRT is as follows: “Provides transport on demand from passengers using fleets of vehicles scheduled to pick up and drop off people by their needs (Mageean et al., 2003)”. The DRT provides door-to-door service to the user by matching the serviceable vehicle with the user in the operating system when the user makes a request. The DRT differs from public transportation in that it operates on flexible routes. In DRT, the route is changed according to the demand pattern of the user. In general, the request of the user is accepted through the application for immediate use and the reservation. DRT is widely used for the elderly and the disabled in the United States (Access-A-Ride, Metro-Access), Canada (Wheel-Trans, Handy Dart). In most countries, user's request is accepted through the reservation. In Korea, DRT services for the disabled are operated according to the regulations of each local government, and 24 hours excluding Ulsan. In most cities, only the priority of dispatch is given, and in some cities, reservations can be made for vehicles only in the night time zone.

Table 1 Operation status of DRT for the disabled in Korea

<table>
<tr><th>Cities</th><th>Service Start</th><th>Number of Vehicles</th><th>Operation Time</th><th>Reservation</th></tr>
<tr><td>Seoul</td><td>2003.1</td><td>487</td><td rowspan="6">00:00 - 24:00</td><td>· Reservation is available but only priority is given.</td></tr>
<tr><td>Daejeon</td><td>2006.1</td><td>122</td><td>· Reservation is not available.</td></tr>
<tr><td>Incheon</td><td>2006.6</td><td>163</td><td>· Reservation is available, but only priority is given.</td></tr>
<tr><td>Busan</td><td>2006.10</td><td>162</td><td>· Reservation is not available.</td></tr>
<tr><td>Gwangju</td><td>2008.9</td><td>121</td><td>· Reservation is available at midnight.
· The vehicle is dispatched at desired pick-up time.</td></tr>
<tr><td>Daegu</td><td>2009.2</td><td>212</td><td>· Reservation is not available.</td></tr>
<tr><td>Ulsan</td><td>2009.3</td><td>89</td><td>07:00 – 22:00</td><td>· Reservation is available. (Midnight: the vehicle is dispatched at desired pick-up time.
Other times: only priority is given.)</td></tr>
</table>

As of 2017, the DRT service for the disabled in Seoul is operating at a lower service level, despite the fact that a total of 487 units exceeding the statutory supply are operating. As of 2016, the average waiting time of DRT service for the disabled in Seoul was 41.9 minutes/request, which was about 1.6 times the average travel time of 25 minutes/request. The satisfaction survey of DRT service for the disabled in Seoul in 2015 showed the lowest satisfaction with waiting time of users.

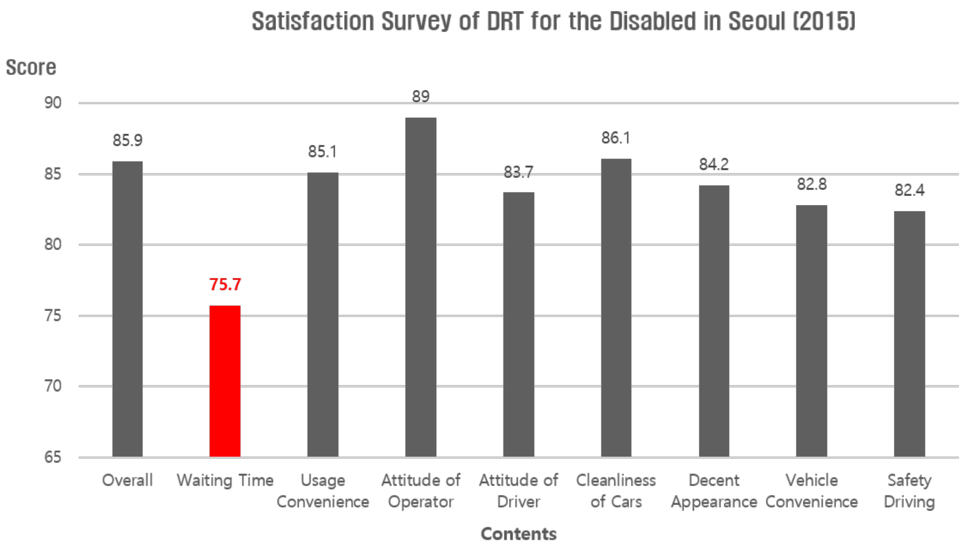


Figure 1 Satisfaction Survey of DRT for the Disabled in Seoul (2015),

(Source: Seoul Metropolitan Facilities Management Corporation, 2016)

The waiting time of the DRT user is defined as the difference between the actual pick-up time and the desired pick-up time. There are two methods to reduce the waiting time. One is to increase the number of vehicles, and the other is to improve the dispatching algorithm. Increasing the supply of vehicles can significantly reduce waiting time, but it has a disadvantage of high cost. On the other hand, although improvements in dispatching algorithms do not significantly reduce waiting time, they can operate sustainably and efficiently regarding cost.

As the service gradually evolves from development to maintenance, increasing the number of vehicles requires a way to efficiently operate the service with a massive budget burden. If the matching time is short in the dispatch procedure, the waiting time of the user first received is likely to pick up after the desired pick-up time, considering the moving time of the vehicle That is, a waiting time occurs. If the matching time is extended, the waiting time of the user with the highest order can be shortened because the probability that the vehicle will be dispatched before the desired pick-up time becomes increasing. On the other hand, the waiting time may become longer due to the effect of reducing the number of vehicles available for the next user.

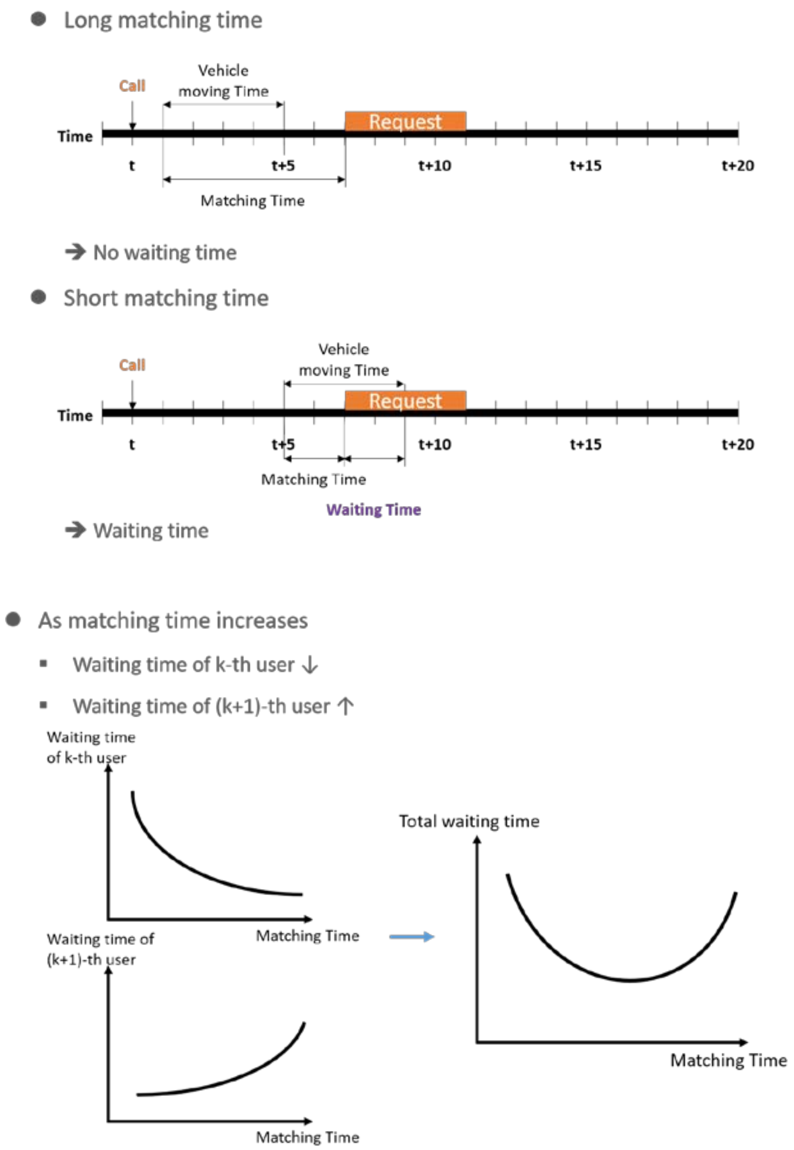


Figure 2 Relationship between Matching Time and Waiting Time

1.2. Research Objectives

This study aims to minimize the waiting time of users to increase the satisfaction of DRT service for the disabled and to find the optimal matching time in service operation and to evaluate the effectiveness of the system operated with the optimum matching time. Therefore, this study developed a simulation model for various scenarios to find the optimal matching time that minimizes the waiting time, and the reality of simulation model was improved through calibration. When DRT services are operated with the optimum matching time that minimizes the waiting time rather than the vehicle increase, the user satisfaction is increased, and the service efficiency can be improved without increasing the operating cost by increasing the number of the vehicles.

Chapter 2. Literature Review

Demand Responsive Transport (DRT) is classified academically as Dial-a-Ride Problem (DARP). DARP is designed to plan routes of vehicles and schedule according to user's service request. It is divided into eight types according to time dependency (static or dynamic), some classes (single or multi), and time constraint (time window or no time window) as follows.

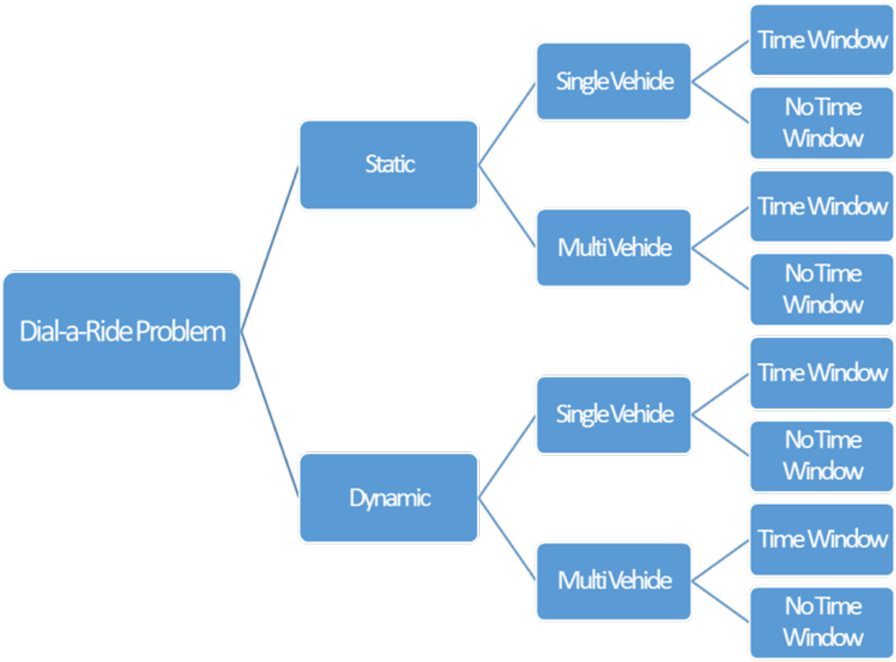


Figure 3 Types of Dial-a-Ride Problem (DARP)

In Static DARP, Psaraftis (1983) considered the problem of minimizing the route duration of a single vehicle by considering both time window and vehicle capacity, and he derived an exact solution through dynamic programming. Desrosiers et al. (1986) extended the size of the problem more than Psaraftis (1983) by including the time window in the constraint. In the case of a multi-vehicle, Jaw et al. (1986), and 250 synthetic data and 2617 real data were used for testing. After that, there are some algorithms such as Parallel Insertion, Tabu Search, Regret based insertion, Genetic Algorithm and Mixed Integer Linear Programming (MILP). However, most of the problems are limited by the size of the problem, and we can see that the size of the problem is not over 2000.

In Dynamic DARP, Madsen et al. (1995) formulated multi-vehicle dynamic DARP for real services for the elderly and the disabled and solved through the insertion algorithm. Teodorovic et al. (2000) tested synthetic data using objective function considering vehicle waiting time. In Beaudry (2010), dynamic DARP was solved using the two-phase heuristic procedure for real-world data in patients in hospitals. Dynamic DARP also has a limitation of the size of problem like static DARP.

Table 2 Literature Reviews of Dial-a-Ride Problem (revised from Cordeau et al. (2007))

Authors	Type	Objective Function	Solving Algorithm	Size of Problem
Psaraftis (1983)	Static Single-Vehicle	Minimize route duration	Dynamic Programming	$n \leq 9$
Desrosiers et al. (1986)	Static Single-Vehicle	Minimize route duration	Dynamic Programming	$n \leq 40$
Jaw et al. (1986)	Static Multi-Vehicle	Minimize combination of user preference and cost	Heuristic. Insertions	$n = 250$, $n = 2617$

Authors	Type	Objective Function	Solving Algorithm	Size of Problem
Toth et al. (1997)	Static Multi-Vehicle	Minimize total service cost	Heuristic. Parallel insertions	$276 \leq n \leq 312$
Cordeau et al. (2003)	Static Multi-Vehicle	Minimize total route length	Heuristic. Tabu search with vertex reinsertions	$24 \leq n \leq 295$
Diana et al. (2004)	Static Multi-Vehicle	Minimize a weighted sum of three elements (total distance, excess ride time, total distance)	Heuristic. Regret-based insertion procedure	$500 \leq n \leq 2000$
Rekiek et al. (2006)	Static Multi-Vehicle	Minimize the number of vehicles	Heuristic. Genetic algorithm for clustering	$n = 250$
Cornillier et al (2012)	Static Multi-Vehicle	Maximize overall net revenue	Mixed Integer Linear Programming	$n = 900$
Madsen et al. (1995)	Dynamic Multi-Vehicle	Multi-criteria objective	Heuristic. Insertions	$n = 300$
Teodorovic et al. (2000)	Dynamic Multi-Vehicle	Minimize a set of route lengths, ride times and waiting time	Sequential insertion of users in vehicle routes	$n = 900$
Coslovich et al. (2006)	Dynamic Multi-Vehicle	Minimize user dissatisfaction	Insertions in current routes Route re-optimizations with modified 2-opt	$25 \leq n \leq 50$

Authors	Type	Objective Function	Solving Algorithm	Size of Problem
Beaudry et al. (2010)	Dynamic Multi-Vehicle	Minimize a weighted sum of total travel time, total lateness, and total earliness	Insertions, Tabu search	$n = 240$ (per day, average)

In the previous research, DARP was solved by considering various objective functions and constraints. However, due to the size of the problem, there were limitations to approach with Mathematical Programming or Insertion algorithm. This study solved the size of the problem by testing the real data from DRT for the disabled in Seoul and developed a simulation model that is widely used to evaluate the real-time operation strategy. In this study, we formulated a model with Dynamic Multi-Vehicle Dial-a-Ride Problem with Time Window by simulation and evaluated the performance of real-time operation strategy for various scenarios.

Chapter 3. Methodology

3.1 Problem Definition

A simulation model was developed to find the optimal matching time for uncertain demand to minimize the average waiting time for users. Information about all demands (types of users, origin, destination, desired pick-up time) is unknown in the simulation model, and types of users are wheelchair users and non-wheelchair users. Requests of users would be dispatched at the point of time after the matching start time, and the vehicles would be driven according to the operating time. Transportation network design consists of nodes and links in a directed network, and locations of users and vehicles are located in nodes. The number of vehicles is limited, and classes of vehicles are wheelchair special vehicles and taxis for the disabled. Wheelchair special vehicles are available for all users, but taxis for the disabled do not have wheelchair boarding facilities, so only non-wheelchair users can be boarded in taxis. The capacity of vehicles is one person per vehicle.

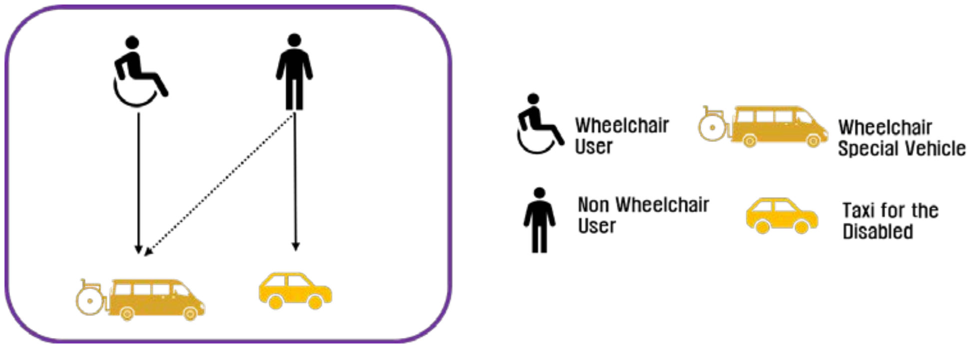


Figure 4 Availability of Vehicle Types by User Types

In the DRT system, when the user requests the destination and the desired pick-up time, the operator starts the match between users and vehicles before the matching time than the desired pick-up time. The match is completed at various times according to the way the service is operated. In the case of service with many users who are not operated in the reservation, the timing at which the match starts according to the matching time differs.

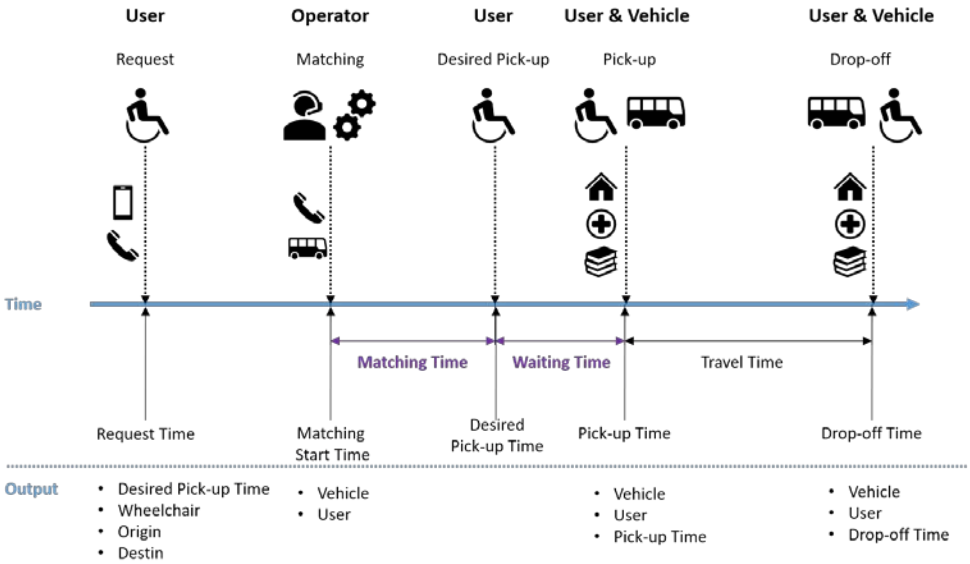


Figure 5 The Procedure of DRT for the Disabled

3.2 Model Formulation

Events ($E = \{e_1, e_2, e_3, e_4\}$)

There were four events that occur in the simulation.: e_1 (Service start), e_2 (Requests of users), e_3 (Movements of vehicles), e_4 (Service end). In 'service start', it was ready to accept requests of users, and the entire system was initialized. In 'requests of users', new requests were received and, matches were made between users and vehicles. Various dispatching algorithms were applied at this stage. In this study, dispatch procedure using various matching time was used. 'Movements of vehicles' consisted of moving to origin of user request for pick-up and movement from origin to destination for service. Finally, when all requests were serviced, the service was finished computing estimates of interest. All events occured and were performed in modules of simulation.

Decision Variables ($\boldsymbol{X = \{x_1, x_2, x_3\}}$)

Operators of DRT service for the disabled determined which vehicle and which user to start when pick-up. Therefore, the decision variables were as follows.

$x_1 = u_j$: The j-th user (1, …, w = wheelchair, w+1, …, u = no wheelchair)

$x_2 = v_i$: The i-th vehicle (1, …, k = special vehicle, k+1, …, I = taxi)

$x_3 = t_s(i, j)$: Start time of i-th vehicle for pick-up j-th user

System Parameters ($\boldsymbol{C = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7. c_8, c_9\}}$)

System parameters represented the parameters of network design, demand, and operation that make up the system. In this simulation model, nine system parameters were used as follows.

$c_1 = t_m(k, l)$: Moving time from station k to station l

$c_2 = T_{mt}$: Matching time

$c_3 = t_t(j)$: Travel time for the j-th user

$c_4 = t_d(j)$: Desired pick-up time for the j-th user

$c_5 = t_d(j)$: Request time for the j-th user

$c_6 = o_j$: Origin of the j-th user

$c_7 = d_j$: Destination of the j-th user

$c_8 = v_1$: Number of wheelchair special vehicles

$c_9 = v_2$: Number of taxis for the disabled

State Variables ($\boldsymbol{S} = \{\boldsymbol{s_1}, \boldsymbol{s_2}, \boldsymbol{s_3}\}$)

State variables represented the state of the system and were determined by decision variables and system parameters ($\boldsymbol{S} = \boldsymbol{f}(\boldsymbol{C}, \boldsymbol{X})$). State variables were dependent on each occurrence of events. In this model, we used the following three state variables.

$s_1 = p_i(t)$: Position of the i-th vehicle at time t

$s_2 = s_i^v(t)$: State of the i-th the vehicle at time t (availability of service or not)

$s_3 = s_j^u(t)$: State of the j-th the user at time t (Completion of service or not)

Objectives and Assumptions

This paper formulated the objective function to minimize the total waiting time of users. Matching time was an external variable, and different values of the matching time were selected as parameters to evaluate total waiting time of users for various matching times. Constraints were as follows for i-v: i) Wait time is 0 or positive. ii) In the time step j, the vehicle assigned to the user k is one. iii) The vehicle assigned to user k is the same vehicle (not transferable). iv) Wheelchair users can only use wheelchair special vehicles. v) Non-negative condition for other variables. The assumptions of this model were:

- No vehicle break-down
- No drivers rest time

- Constant vehicle speed
- Negligible boarding time
- Negligible search time for matching

Modules of Simulation

The simulation developed in this study consists of three modules: ①: Initialization Module, ②: Timing Module, ③: Event Module. First, when the simulation started, the initialization module was invoked to initialize variables and states of all simulations. Next, the simulation clock was advanced by the timing module, and the simulation proceeded to the event module. The event module compared the current time with the matching start time of requests, received requests to start matching, sorted the list of waiting users who had not yet received the service, and checks whether the current time was after the matching start time (= desired pick-up time - matching time). It matched vehicles and users by a sequence of requests. The order of matching vehicles and users in the simulation was as follows.

Step 1. Check if there is a serviceable vehicle within a certain radius.

Step 2. Check if service is available during moving time and travel time.

Step 3. If it does not match, repeat step 1. ~ step 2. at the next time step.

In the match between vehicles and users, wheelchair users were available only for wheelchair special vehicles, and non-wheelchair users were available for all vehicles. When all requests were serviced, the waiting time and state variables were computed to yield the final result.

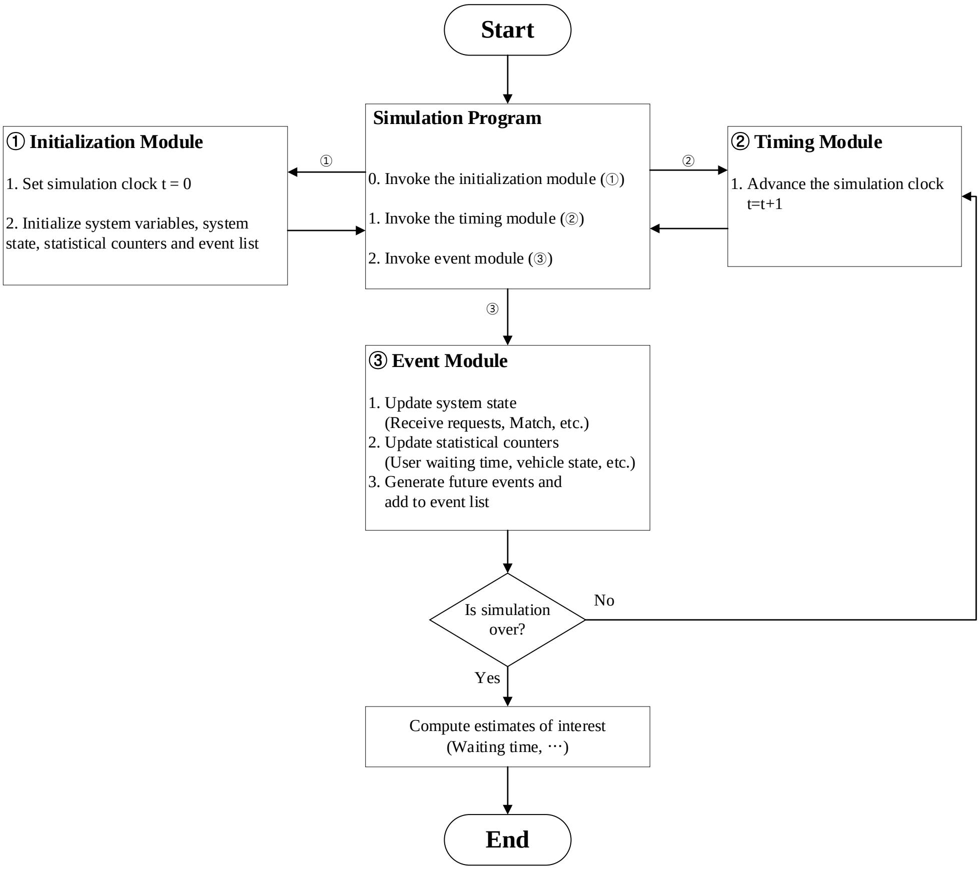


Figure 6 Flow Chart of Simulation (revised from Law (2007))

The simulation model was coded in the Java language to utilize and link various APIs in the future.

```java
int Matching_starttime=0;

// Matching starttime에 따라 계산
for (int m = 30; m<=30; m+=5){
    Matching_starttime=m;
    try {

        //// Declaration and Initialization of Variables
        // 1. Decision Variables (X)
        int[] User    = new int[N_of_Requests]; // User
        int[] User_type  = new int[N_of_Requests]; // User

        // 2. System Parameters (C)
        int[][][][] OD_traveltime   = new int[7][24][N_of_Depots][N_of_Depots]; // Moving time for pick-up
        int[][] OD_order         = new int[N_of_Depots][N_of_Depots];
        int[][] OD_order_dist    = new int[N_of_Depots][N_of_Depots];
        int[] User_traveltime    = new int[N_of_Requests];
        int[] User_desiredtime   = new int[N_of_Requests];
        int[] User_calltime      = new int[N_of_Requests];
        int[] User_origin        = new int[N_of_Requests];
        int[] User_destin        = new int[N_of_Requests];
        int[] User_hopeday       = new int[N_of_Requests];
        int[] User_hopedays      = new int[N_of_Requests];
        int[] User_hopetod       = new int[N_of_Requests];
        int[][] Vehicle_station = new int[37][3]; //차량 차고지

        // 3. State Variables (S)
        int[][] V1_position  = new int[N_of_V1][N_of_TimeSteps+1]; // Position of Special vehicle
        int[][] V2_position  = new int[N_of_V2][N_of_TimeSteps+1]; // Position of Taxi
        int[][] V1_state     = new int[N_of_V1][N_of_TimeSteps+1]; // State of Special vehicle
        int[][] V2_state     = new int[N_of_V2][N_of_TimeSteps+1]; // State of Taxi
        int[] User_state     = new int[N_of_Requests]; // State of User
        int[] V1_stock = new int[N_of_Depots]; //초기 차량재고
        int[][] V1_check = new int[N_of_Days][N_of_V1]; //처리한서비스개수
        int[][] V2_check = new int[N_of_Days][N_of_V2]; //처리한서비스개수
        int[] Servicevehicle_Type = new int[N_of_Requests];

        // 4. System Performance Criterion (Y)
        int[] User_waitingtime = new int[N_of_Requests];
        int[] User_realwaitingtime = new int[N_of_Requests];


        // Input OD travel time
        for (int d=0; d<7;d++){
            for (int o=0; o<24; o++){
                String Input_ODtt = "C:\\Users\\Eunhan\\Desktop\\Java_workspace\\Master_Paper\\OD\\OD_traveltime_real_" + String.valueOf(d) + "_" + String.valueOf(o) + "_f.csv";
                File csv = new File(Input_ODtt);
                BufferedReader br0 = new BufferedReader(new FileReader(csv));
                String line = "";
                int row_csv0 =0 ,i;
                while ((line = br0.readLine()) != null) {
                    // -1 옵션은 마지막 "," 이후 빈 공백도 읽기 위한 옵션
                    String[] token0 = line.split(",", -1);
                    for(i=0;i<N_of_Depots;i++)    OD_traveltime[d][o][row_csv0][i] = (int) Float.parseFloat(token0[i]);
                    row_csv0++;
                }
                br0.close();
            }
        }
```

```java
        // Simulation Time clock
        for (int t=0+Matching_starttime;t<=N_of_TimeSteps;t++){

            int start_position = 0;
            int movetime = 0;

            // 시간대 확인 (차량의 movetime을 정하기 위해 필요함.
            for (int td = 0; td<24;td++){
                if(t>=1440+60*td && t<1440+60*(td+1)){
                    tod=td;
                } else if (t<1440){
                    tod=0;
                } else {
                    tod=23;
                }
            }
            // Receive requests and sort sequence of request in list of waiting users
            for (int r=0;r<N_of_Requests;r++) {
                // (waiting user) 처리되지 않고, 배차를 고려해야하는 request 고려
                if(User_state[r] == 0 && t>=User_desiredtime[r]-Matching_starttime && t>=User_calltime[r]) {
                    // 가까운 지점부터 차량찾기
                    for (int s=0;s<N_of_Depots;s++) {
                        start_position = OD_order[s][User_origin[r]]; // 가까운 지점
                        movetime = OD_traveltime[User_hopeday[r]][tod][start_position][User_origin[r]]; //가까운지점에서 origin까지 시간
                        //평소시간대에는 5km이내만 검색
                        if(OD_order_dist[start_position][User_origin[r]]<5 || t>1440+23*60 || t<1440+5*60){
                            // 해당지점에서 서비스차량있는지 조회
                            for (int v= 0; v<N_of_V1;v++) {
                                //휠체어 유저 1, 아니면 0
                                if(User_type[r]==1) {
                                    // 서비스가능차량을 찾기
                                    if(V1_position[v][t]==start_position && V1_state[v][t]==1) {
                                        // 대기시간이 생기는지 여부
                                        if (t+movetime>=User_desiredtime[r]) {
                                            // 끝까지 서비스 가능한 차량인지 조회
                                            for (int t1=t; t1<t+movetime+User_traveltime[r]+6;t1++) {
                                                if(V1_state[v][t1]==1) {
                                                    //마지막까지 가능하면 매치 (대기시간계산, 유저 서비스처리, 차량 서비스중 처리, 차량 도착지로 위치 변경)
                                                    if(t1==t+movetime+User_traveltime[r]+6-1) {
                                                        User_waitingtime[r]=t+movetime-User_desiredtime[r]; // 대기시간계산
                                                        User_state[r]=1; // 유저 서비스처리
                                                        Servicevehicle_Type[r]=0;
                                                        for (int t2=t;t2<t+movetime+User_traveltime[r]+6;t2++) {
                                                            V1_state[v][t2]=0; //차량 서비스중 처리
                                                        }
                                                        V1_check[f-1][v]++; //차량이 서비스처리한건수 체크
                                                        for (int t2=t+movetime+User_traveltime[r]+6;t2<N_of_TimeSteps;t2++) {
                                                            V1_position[v][t2]=User_destin[r]; //차량 도착지로 위치 변경
                                                        }
                                                    } // if t1
                                                } // if V1_state
                                                else {
                                                    break;
                                                } // else V1_state
                                            } // for t1
                                        } // waitingtime
```

Figure 7 Part of Simulation Code Implemented in Java

As shown in the figure below, in case 1, no match was made because there was no serviceable vehicle within a certain radius. A serviceable vehicle must be located within a certain radius based on the origin of the user, and a serviceable vehicle was searched for every time step until a user request was matched. In the next time step, situation 2, there was a serviceable vehicle within a certain radius, which made a match between the vehicle and the user.

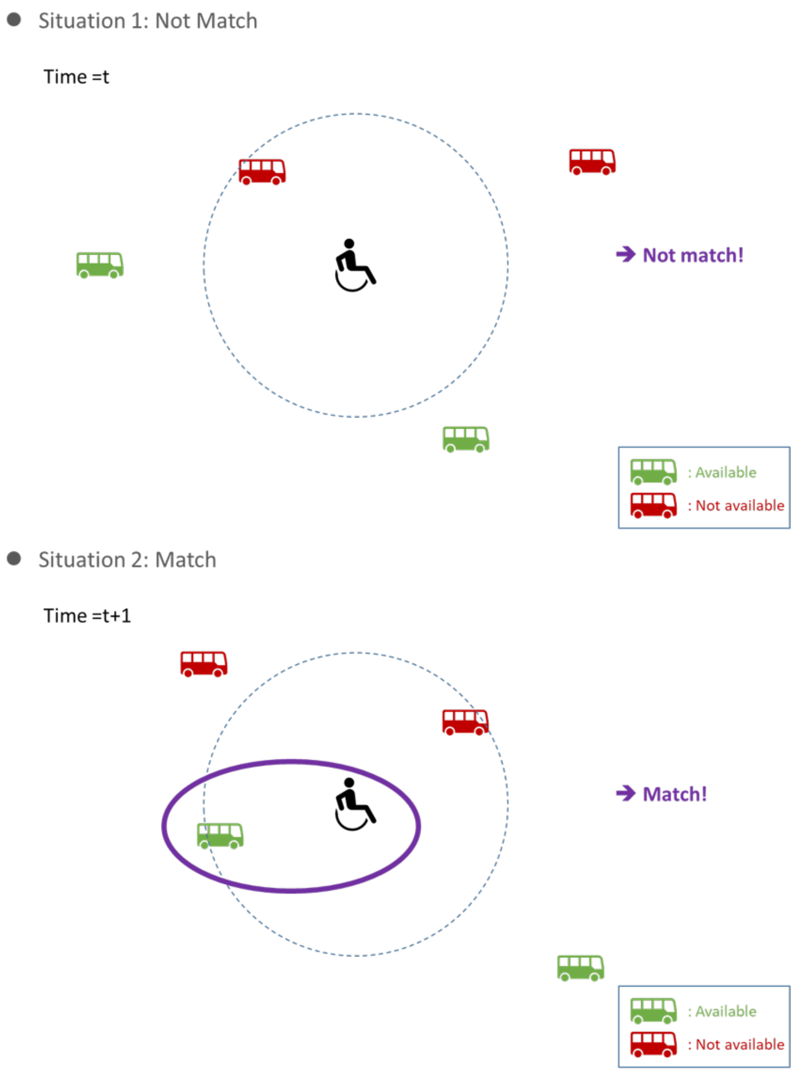


Figure 8 Example of Match Procedure

Chapter 4. Data Collection and Results

4.1 Data collection

We obtained real data for DRT service for the disabled in Seoul in 2016 in order to test the real data. DRT for the disabled in Seoul operates a total of 487 vehicles (437 wheelchair special vehicles and 50 taxis for the disabled) in 2016.

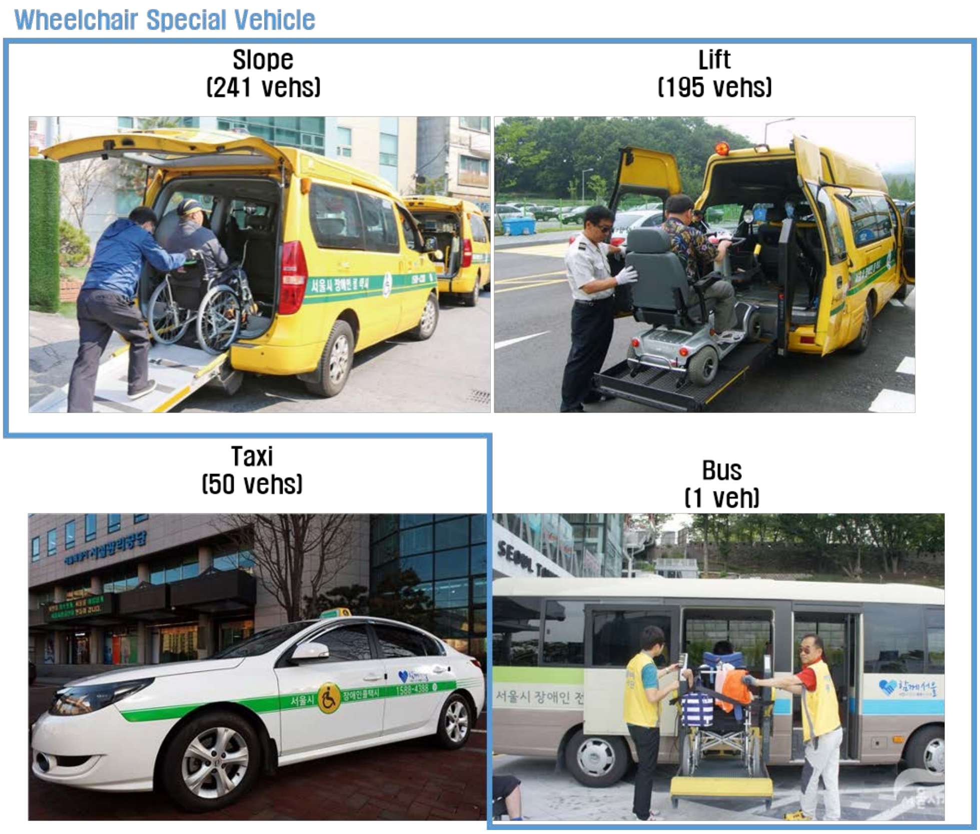


Figure 9 Vehicles Types of DRT for the Disabled in Seoul
(source: http://www.seouland.com/arti/society/society_general/55.html, http://heraldk.com)

The number of users of the DRT for the disabled in Seoul was 84,529 people, and only the handicapped with disability level 1 and 2 living in Seoul can use the DRT service. The fare of DRT for the disabled in Seoul was legally estimated to be less than three times the city rail fare. The period of data was from January 1, 2016, to December 31, 2016, for a total of 366 days. Service areas were in Seoul city and the neighboring areas. In 2016, service was operated in 716 administrative districts (Korean name is dong). The number of requests was 1,246,898 requests, with an average of 3,400 users per day. The data was as follows.

Table 3 Data Attributes of DRT for the Disabled in Seoul

Number	Name	Type (Examples)
1	Request Date	Date (2016-03-31)
2	Vehicle Number	ID (78마1185)
3	Request Time	Time (2016-03-31 7:00)
4	Desired Pick-up Time	Time (2016-03-31 8:00)
5	Match Time	Time (2016-03-31 7:41)
6	Pick-up Time	Time (2016-03-31 8:11)
7	Drop-off Time	Time (2016-03-31 8:35)
8	Origin	Character (Daehak-dong)
9	Destination	Character (Sadang-dong)

The number of locations (origin, destination) was 716 administrative districts (Seoul: 424, Neighboring areas: 292) because the Privacy Policy Terms limited the availability of detailed location information. Types of users (wheelchair or non-wheelchair) were randomly assigned using the ratio of wheelchair users (= 64.5%)

announced by Seoul Metropolitan Facilities Management Corporation (2017). In addition, the trip purpose of users was not considered in the simulation model by the Privacy Policy Terms, and all trip purposes of users were assumed to be the same.

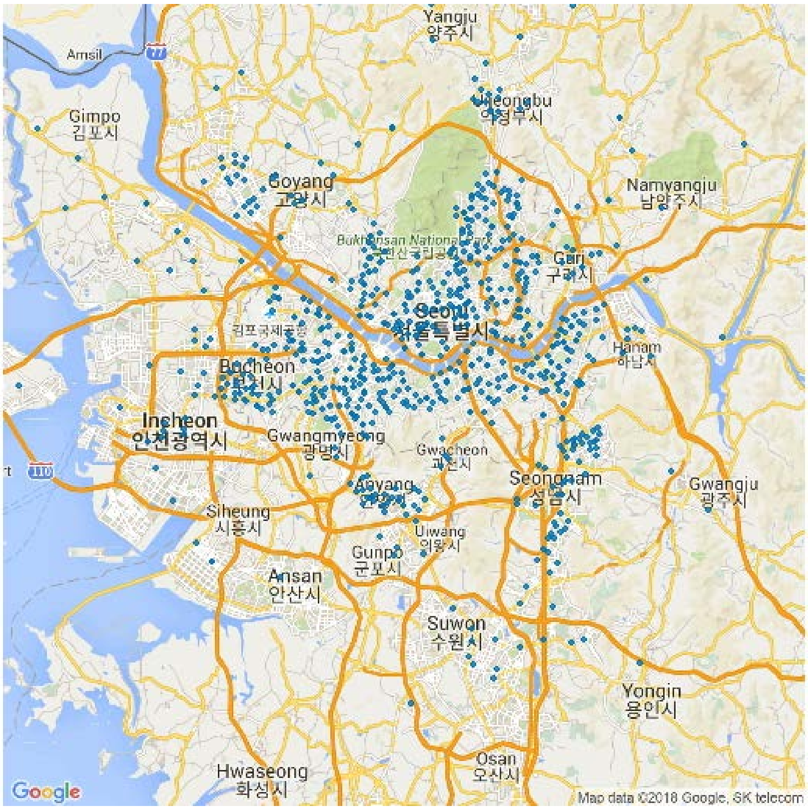


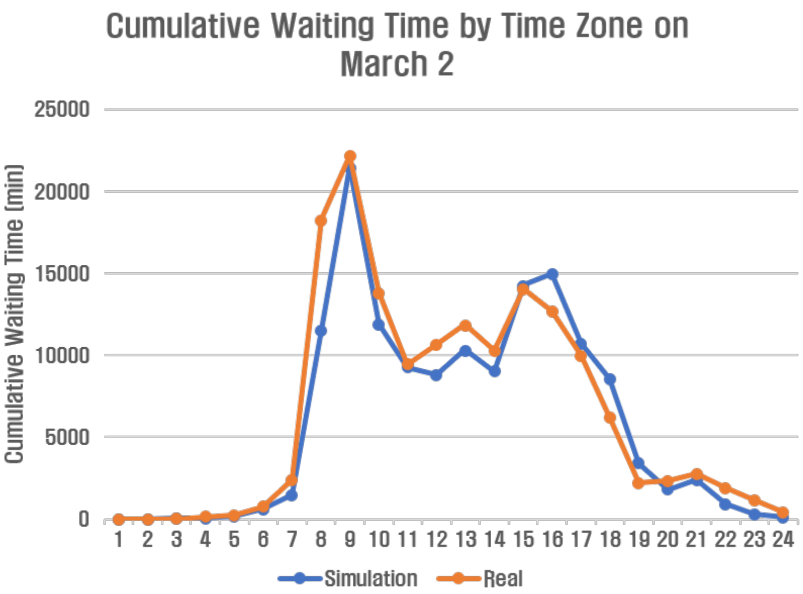
Figure 10 Map of Operational Area (716 administrative districts)

In the case of travel time for service and moving time for pick-up, we calculated the real travel time of DRT for the disabled in Seoul by day and hourly time. The number of vehicles operated by the time of day was estimated based on Seoul Metropolitan Facilities Management Corporation (2017) due to operational information security. The parameters used in this study are as follows.

Table 4 Parameters Used in Simulation

Number	Name	Definition
1	c_1: Moving Time	= Drop-off Time – Pick-up Time
2	c_2: Matching Time	= 0, 5, 10, …, 60 (by Various Scenarios)
3	c_3: Travel Time	= Drop-off Time – Pick-up Time
4	c_4: Desired Pick-up Time	= Desired Pick-up Time
5	c_5: Request Time	= Request Time
6	c_6: Origin	= Origin
7	c_7: Destination	= Destination
8	c_8 : Number of Wheelchair Special Vehicles	= 437
9	c_9 : Number of Taxis for the Disabled	= 50

This study was conducted to test the data of DRT for the disabled in Seoul from March 20, 2016, to March 31, 2013 (total number of days = 30). A total of 105,462 requests were simulated, and the simulation model was calibrated using estimated parameters. The calibration results were compared with real data and simulation results for the cumulative waiting time by time zone and average waiting time, respectively.



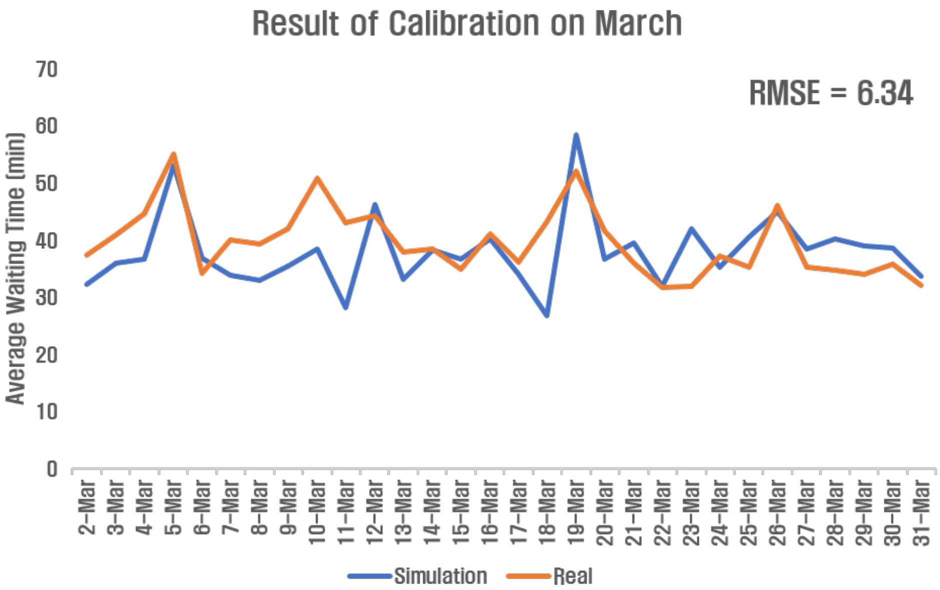


Figure 11 Results of Calibration

4.2 Results

This study computed the average waiting time for each of the various matching times (0, 5, ..., 60) and estimated the matching time with the minimum average waiting time as the optimal matching time. Simulation results showed that as the matching time increased, the average waiting time gradually decreased and then tended to increase.

If the optimal matching time is longer than the average moving time of vehicles (26 minutes), there is a high probability that the vehicle will wait before picking up the user, thus reducing the number of available vehicles. That is, the number of vehicles available to the next user decreases, and the average waiting time of the whole users increases. On the other hand, if the optimal matching time is shorter than the average moving time of vehicles, the vehicle would arrive later than the desired pick-up time and the user's waiting time would increase.

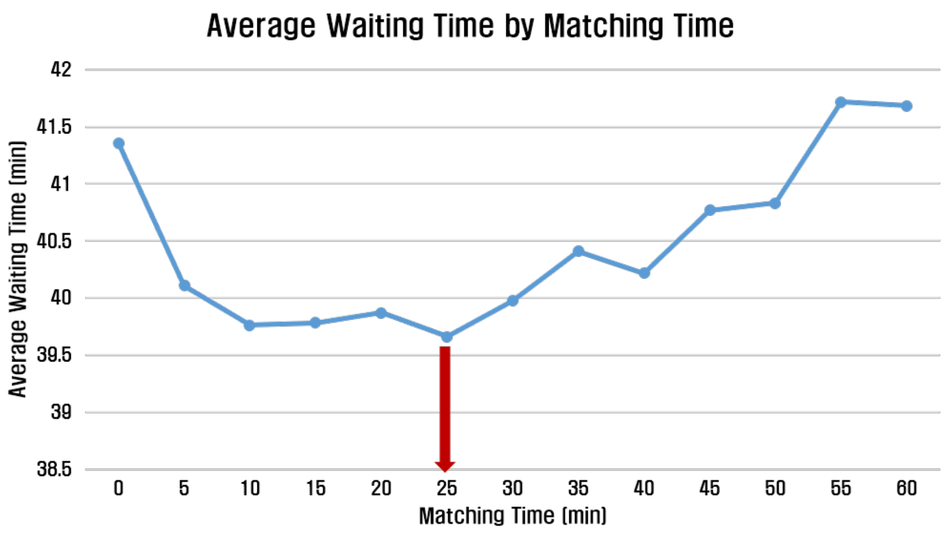
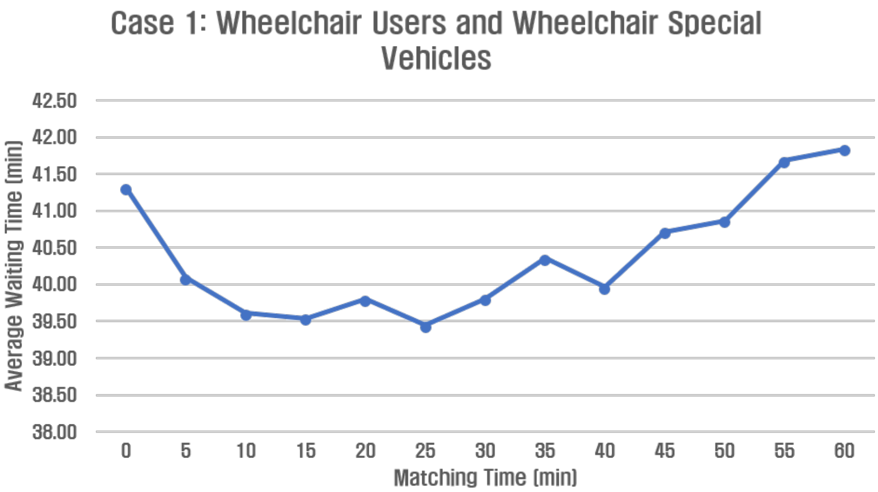


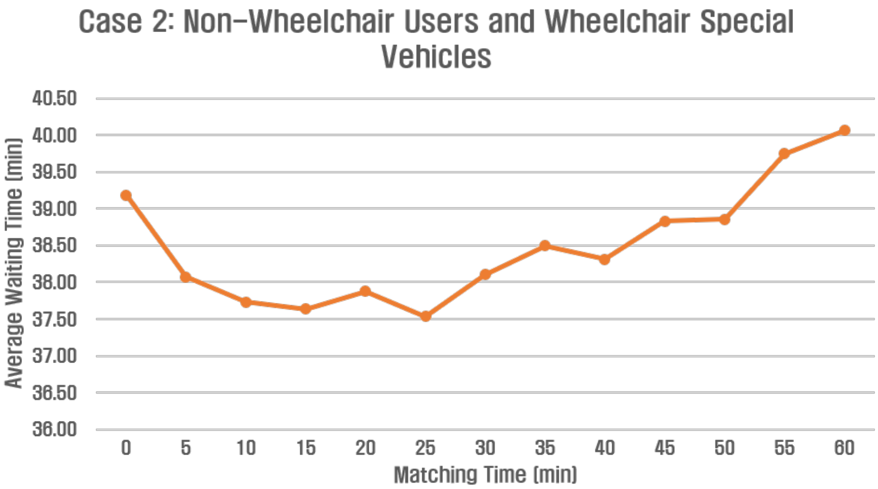
Figure 12 Plot of Average Waiting Time by Matching Time

Table 5 Results of Average Waiting time by Matching Time

Matching Time	Average Waiting Time
0	41.4
5	40.1
10	39.8
15	39.8
20	39.9
25	39.7
30	40.0
35	40.4
40	40.2
45	40.8
50	40.8
55	41.7
60	41.7

The types of the match according to the type of vehicle and the user were wheelchair users and wheelchair special vehicles (Case 1), non-wheelchair users and wheelchair special vehicles (Case 2), and non-wheelchair users and taxis for the disabled (Case 3). For each case, the average waiting time according to the matching time was minimized when the matching time was 25 minutes. This was because the average moving time of vehicles was also about 26 minutes for each type of match, so the average waiting time of all cases was minimized at the matching time of 25 minutes which was closest to the average moving time of vehicles.





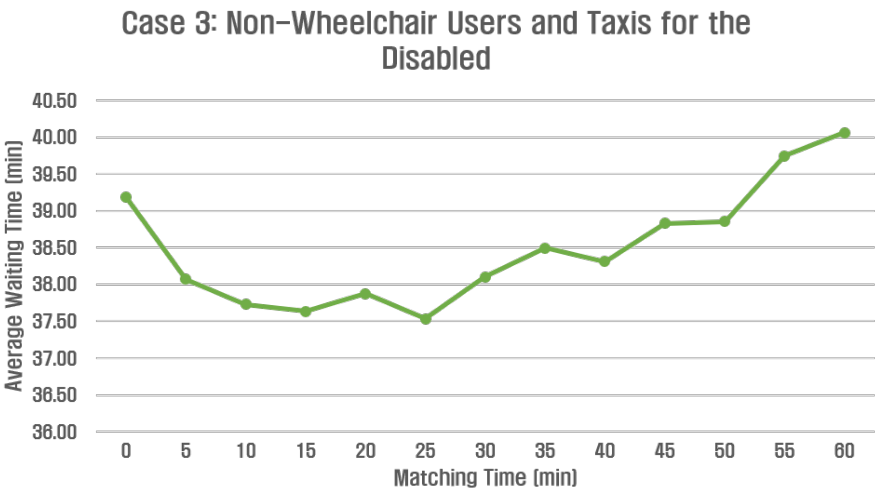


Figure 13 Results of Each Type for Match

The simulation model of this study was tested on real data from DRT for the disabled in Seoul. These results were based on real data except some data for Privacy Policy and operating system security reasons, and the results of real data including private data could be partially changed. However, this study aims at comparing and evaluating the level of service of DRT for the disabled according to various matching times. Therefore, the simulation model developed in this study can be used to develop various operation strategies.

Chapter 5. Conclusion

This study developed a simulation model to improve the satisfaction of DRT for the disabled which is operated for the disabled and to find the matching time minimizing the waiting time as a way to reduce the waiting time to improve the service level. DRT is categorized as "Dial-a-ride Problem" and agent-based simulation was developed in this study for the performance evaluation of the operation strategy for the limit of the size of the problem and various scenarios.

Simulation model consisted of 3 modules. In case of event module, it was developed according to the procedure of DRT for the disabled. The simulation model coded in the Java language environment was tested for a total of 105,462 requests during March 2016 of the DRT for the disabled in Seoul. A simulation model was performed by calibrating the real data in advance and comparing the average waiting time of users according to various matching times and estimating the optimal matching time for 25 minutes. Compared with the current DRT for the disabled in Seoul, the waiting time is reduced and the number of users waiting for more than 60 minutes is also reduced.

The data obtained in this study was not adequately described in describing the real system in that it analyzed the real data by assuming some data for Privacy Policy and operational security reasons. However, the simulation model developed in this study was based on real data for one month, and real data for one year was analyzed to estimate the parameters to enhance the reality of the simulation model. This study has an academic contribution in that it can evaluate the performance of DRT service for the disabled and develop efficient real-time operation strategies for various operation strategies in the future by simulating a large-scale network of demand responsive transport for the disabled.

The dispatch method using the optimal matching time estimated in this study has the following social contributions: i) Improve mobility and satisfaction of demand responsive transport service for the disabled. ii) Increase efficiency of demand responsive transport service for the disabled. iii) Support decision making of the operator by evaluating various strategies.

In future research, user classification dispatch strategy considering user types and vehicle types can be developed, and waiting time can be predicted using various APIs and applied to dispatch methods.

References

Psaraftis, H. N. (1980). A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science, 14*(2), 130-154.

Psaraftis, H. N. (1983). An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. *Transportation science*, *17*(3), 351-357.

Bodin, L. D., & Sexton, T. (1986). The multi-vehicle subscriber dial-a-ride problem. *TIMS studies in Management Science*, *26*, 73-86.

Desrosiers, J., Dumas, Y., & Soumis, F. (1986). A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *American Journal of Mathematical and Management Sciences*, *6*(3-4), 301-325.

Jaw, J. J., Odoni, A. R., Psaraftis, H. N., & Wilson, N. H. (1986). A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological*, *20*(3), 243-257.

Borndörfer, R., Grötschel, M., Klostermeier, F., & Küttner, C. (1997). *Telebus Berlin: Vehicle scheduling in a dial-a-ride system*. Konrad-Zuse-Zentrum für Informationstechnik.

Toth, P., & Vigo, D. (1997). Heuristic algorithms for the handicapped persons transportation problem. Transportation Science, 31(1), 60-71.

Cordeau, J. F., & Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological, 37*(6), 579-594.

Aldaihani, M., & Dessouky, M. M. (2003). Hybrid scheduling methods for paratransit operations. *Computers & Industrial Engineering, 45*(1), 75-96.

Diana, M., & Dessouky, M. M. (2004). A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B: Methodological*, *38*(6), 539-557.

Rekiek, B., Delchambre, A., & Saleh, H. A. (2006). Handicapped person transportation: An application of the grouping genetic algorithm. *Engineering Applications of Artificial Intelligence*, *19*(5), 511-520.

Cordeau, J.-F. (2006). A Branch-and-cut algorithm for the dial-a-ride problem. Operations Research, 54, 573–586.

Cordeau, J. F., & Laporte, G. (2007). The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, *153*(1), 29-46.

Jørgensen, R. M., Larsen, J., & Bergvinsdottir, K. B. (2007). Solving the dial-a-ride problem using genetic algorithms. Journal of the Operational Research Society

Melachrinoudis, E., Ilhan, A. B., & Min, H. (2007). A dial-a-ride problem for client transportation in a healthcare organization. Computers & Operations Research, 34, 742–759.

Ropke, S., Cordeau, J.-F., & Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. Networks, 49, 258–272.

Wolfler Calvo, R., & Colorni, A. (2007). An effective and fast heuristic for the dial-a-ride problem. 4OR: A Quarterly Journal of Operations Research, 5, 61–73.

Colorni, A., & Righini, G. (2001). Modeling and Optimizing Dynamic Dial-a-Ride Problems. International transactions in operational research, 8(2), 155-166.

Coslovich, L., Pesenti, R., & Ukovich, W. (2006). A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. European Journal of

Operational Research, 175(3), 1605-1615.

Madsen, O. B., Ravn, H. F., & Rygaard, J. M. (1995). A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. Annals of operations Research, 60(1), 193-208.

Teodorovic, D., & Radivojevic, G. (2000). A fuzzy logic approach to dynamic dial-a-ride problem. Fuzzy sets and systems, 116(1), 23-33.

Mageean, J. and J.D. Nelson. 2003. The evaluation of demand responsive transport services in Europe. Journal of Transport Geography 11: 255-270.

Cornillier, F., Boctor, F., & Renaud, J. (2012). Heuristics for the multi-depot petrol station replenishment problem with time windows. *European Journal of Operational Research*, *220*(2), 361-369.

Beaudry, A., Laporte, G., Melo, T., & Nickel, S. (2010). Dynamic transportation of patients in hospitals. *OR spectrum*, *32*(1), 77-107.

Seoul Metropolitan Facilities Management Corporation. (2016). Improvement of Citizen Satisfaction by Call to Disabled Persons.

Seoul Metropolitan Facilities Management Corportation (2017). Status for the Handicapped in Seoul call taxis.

Gottfried, B. S. (1984). *Elements of stochastic process simulation* (No. 04; QA76. 9. C65, G67.).

Law, A. M., Kelton, W. D., & Kelton, W. D. (2007). *Simulation modeling and analysis* (Vol. 3). New York: McGraw-Hill.

Leemis, L. M., & Park, S. K. (2006). *Discrete-event simulation: A first course*. Upper Saddle River, NJ: Pearson Prentice Hall.

요약(국문 초록)

장애인콜택시 이용자 대기시간 최소화를 위한 시뮬레이션 연구

Demand Responsive Transport (DRT)는 교통약자 (장애인, 고령자, 임산부 등)를 위한 Door-to-Door 서비스로 널리 활용되는 교통수단이다. 교통약자는 이동하는데 몸이 불편한 사람들로 대중교통을 이용하는데 제한이 있거나 큰 어려움을 겪는다. DRT는 이용자가 원하는 시각, 원하는 장소를 요청하면 이용자의 요구에 맞춰 이동서비스를 지원할 수 있어, 교통약자를 위한 맞춤 서비스를 제공하는데 적합한 교통수단이다. 본 연구는 교통약자의 이동편의 증진을 위해 DRT의 서비스수준을 높이기 위한 방안으로 서비스수준의 지표 중 하나인 이용자 대기시간을 최소화시킬 수 있는 방안을 연구하고자 시뮬레이션을 개발하였다. 시뮬레이션은 다양한 운영방안을 평가하는데 유용하게 활용되며, DRT의 경우 Problem size가 크기 때문에, 수리적해법보다 시뮬레이션을 개발하였다. 서울시에서 운영하는 서울시 장애인콜택시 2016년의 자료를 활용하여 본 연구는 시뮬레이션을 개발하고, calibration하여 최적 Matching Time을 찾고, 기존 운영방안과 비교하여 효과를 평가하였다.

주요어: Demand Responsive Transport (DRT), The Disabled, Simulation, Dial-a-Ride Problem, Performance Evaluation

학번: 2016-21238