Study on the Development of an Optimum Bus Route Search System

Supervised by Byeong-Deok YEA

August 2006

Graduate School of Korea Maritime University

Department of Marine Transportation

Information Engineering

Soon-Chun PARK

최적의 버스경로 탐색 시스템에 관한 연구

박 순 천

한국해양대학교 대학원
해상교통정보공학과
지도교수 예 병 덕

요 약 문

현재 자가용의 대중적인 보급으로 인하여 버스의 이용이 급격이 저하되어 있는 상황이다. 버스와 철도와 같은 공공교통수단보다 자가용을 선호하게 되면서 에너지 소비의 증가와 공해증가 등의 문제가 발생하고 있다. 또 지역의 버스회사가 재정난으로 인해 어려움을 겪게 되고, 그로 인해 서민들의 발인 버스운행에 어려움이 발생하게 된다. 따라서 우리는 버스이용이 줄어드는 이유를 파악하고 이를 개선할 필요가 있다.

버스이용이 불편한 가장 큰 원인은 노선정보를 얻기가 매우 까다롭다는 것이다. 철도나 항공의 경우에는 역과 공항에서의 자료를 얻거나 안내원들에게 물어볼 수 있고 인터넷 상으로도 이용에 필요한 정보를 얻을 수 있지만 버스의 경우는 다르다. 이것이 곧 노선버스의 이용을 불편하게 하는 원인이 되고 있다.

이 논문에서는, 여행객과 같이 그 지역의 지리에 대해 잘 알지 못하는 사람들도 공공교통을 이용하고자 한다면, 이러한 버스의 노선정보를 손쉽고 편리하게 제공받음으로써 버스 이용객의 편의를 도모함은 물론 이

용객을 증가시키는 것을 목적으로, 버스경로탐색시스템을 개발하는 것에 대하여 기술하고 있다.

그리하여 우리는 웹상에서 버스네트워크를 위한 노선탐색 시스템을 개발하였다. 이 시스템은 GPS로부터 얻은 출발지와 도착지 또는 랜드마크 데이터베이스 위치정보와 현재시간을 이용하여 적당한 경로를 찾는다. 우리의 노선탐색 방법은 버스의 갈아타기 뿐만 아니라 도보이동도 고려한 탐색방법이다. 그러나 다익스트라의 최단경로 알고리즘만을 이용한 노선탐색방법은 실용적이지 못하다. 따라서 버스 특유의 매우 유동적인 성격을 소화할 수 있는 알고리즘을 제시하고자 하였다. 그래서 전체탐색을 위해 탐색방법을 크게 2단계로 구분하였다. 이 시스템은 1단계에서 다익스트라 알고리즘으로부터 얻은 최단경로 추정치를 이용하여 경로를 탐색하고, 2단계에서 몇 가지 조건을 이용한 가지치기 방법과 탐색순서, 탐색조건 등을 채택하여 가장 적절하고 실용적인 경로를 선택한다. 실험에 의해서 그 유효성이 증명되었다.

이와 같이 하여 개발된 시스템은 현재 인터넷을 통하여 일반에 공개되어 있으며(http://www.ikisaki.jp/), 이용자들로부터 호평을 받고 있다.

Contents

Chapter 1

Introduction

With the economic development motorization has been accelerated in developing countries as well as developed countries, which has caused not only the pollution and global warming by the exhaust gas but also the increase of public and private expenses for physical distribution due to congestions. To cope with these problems the public transportation systems must be prepared and utilized by as many persons as possible. For this purpose mass transfer systems such as high speed trains, subways, buses, etc. have been developed and provided by the central and local governments all over the world.

Mass transfer by buses is still a major transportation system in many small cities even though the subway systems have become the main transportation system in the big cities in many countries recently. However, passengers, especially the tourists and strangers for the city, can not get the information easily about the buses and their routes due to the peculiar characteristics of the bus transportation systems. Excellent Route Search Systems for Bus Transportation are necessary to reduce the pollution and expenses due to private cars and to increase the utilization of public transportation systems.

The development of a Bus Route Search System in Tottori City, Japan that is a result of Government-Industry-University Cooperation will be described precisely following chapter.

1.1 Current Situation of Bus Transportation System in Tottori City

The number of people who use bus transportation system is decreasing year by year in every province of Japan. Especially, almost 70% of all bus routes operated in Tottori Prefecture was red bottom line in 1999 and, therefore, it became a great burden to the local government to maintain the bus transportation system. To increase the number of people who will use bus transportation it is necessary to enhance the convenience in several points. The first example of inconvenience in bus system is that it is difficult to get the information relating to the bus routes. In case of railroad transportation system, passengers can get easily the route related information from route map, timetable, tariff, or by making inquires of staffs at the station. In case of bus stops, however, there is not enough space to notify detailed route information nor any staff to ask generally and, as a result, route information available from the bus stop is limited considerably. In general, railroad stations are indicated very well on the signpost; however, bus stops are not clearly indicated on the guidepost and, therefore, it is very difficult for tourists or strangers to find the location of bus stops. Furthermore, there are well equipped route search systems at railroad stations in general which can show detailed route information by simply inputting departure and arrival stations to the system; on the other hand, there is not any good system available for bus transportation system.

The bus companies in Tottori City have recognized the urgent need of bus route search system in order to provide convenience to passengers and, consequently, to enhance the utilization of bus transportation system. They have cooperated to develop a route search system with Tottori University which has professors and experts in the field. The Tottori Chamber of Commerce and Industry[1] has participated in this cooperation later, which resulted in the Government-Industry-University Cooperation, and promoted the development of the system.

1.2 Route Search System for Bus Transportation

The number of people has been increased who uses the Internet Service or Package Software that can show the course information such as estimated time required, fare, transfer guide, etc. of the public transportation systems by inputting the departure and arrival stations to the software. This kind of service has been started in the railroad transportation system because the route and time of the railroad are almost fixed and accurate. Although the service is expanded in part to the public bus transportation system recently, there is no fully available route search system in case of public bus transportation service in a city due to the peculiar characteristics of the bus route.

The currently available route search system for bus transportation in a city utilizes the search system for railroad without any change; however, the bus routes are very much different from those of railroad and, therefore, the search results for bus route by the system are not sufficient to be utilized by the passengers satisfactorily. One of such differences is that the distance between bus stops in bus transportation system is much shorter generally than that in railroad system. Because of this reason passengers prefer transfer by walk to by bus in some cases, and this situation is one of the main reasons to make the realization of the system difficult.

This thesis deals with the development of Route Search System for Bus Transportation. Users of this system can get the optimum route that minimizes the time and cost between two points taking all available means of transportation such as transfer by walk, bus, railroad etc. into consideration. Furthermore, this system enables the users to find optimum route not only with the names of the departure and arrival bus stops but also the names of landmarks such as office buildings, famous sightseeing spots, etc. This function is very much helpful for tourists and strangers who do not know the City well.

The route search algorithm has been designed to find optimum route with 2 steps. The first step is to find the shortest path in time between two points. The second step is to find optimum route from the shortest path in time by considering other circumstances.

1.3 Support System for Bus Route in Tottori City

The developed system has the following functions:

1) Route Search Function
2) Data Management Function
3) Landmark search Function
4) Timetable Display Function for Each Bus Stop
5) Map Display Function for Transfer by Walk
6) Timetable Display Function for Each Bus Route

The function 1) is implemented with the search algorithm proposed in the former section. The design, loading and assessment of the function will be explained in the later chapters. 2) is a function of data management for public transportation system in Tottori City. The data is revised two times a year generally and, therefore, the database shall be amended two times a year as well. This function was designed elaborately to achieve smooth revision of the database. The function 3) is a sort of input assistant function that can be used to select and input departure and arrival places in the function 1) not with the names but with the landmarks such as nearest building, memorial monument, public office, etc. The function 4) shows timetable of route buses based on the bus stops. With this function passengers can confirm the time when they use the bus at the specific bus stop. In the function 5) users can cut out and display arbitrary size of map by selecting 2 points on the original map. In addition this function can show the direction from 1 point to another, which can be utilized to offer area maps from the starting point to departure bus stop and/or from arrival bus stop to the destination. The function 6) shows timetable of buses based on the bus route. With this function passengers can find the time required between stops and can understand the time zone when they can use the bus route.

The functions 3) and 5) are support functions for the function 1). And the function 2) is a management system for public transportation information. Figure 1.1 shows the taxonomy for functions of the support system.

1.4 Composition of the Thesis

This thesis is composed of 7 chapters.

After this introduction chapter, the characteristics of the bus route are explained in Chapter 2. Specific features of bus route such as transfer by walk between bus stops, direct routes, bus stops with same names, etc. are defined and precisely described.

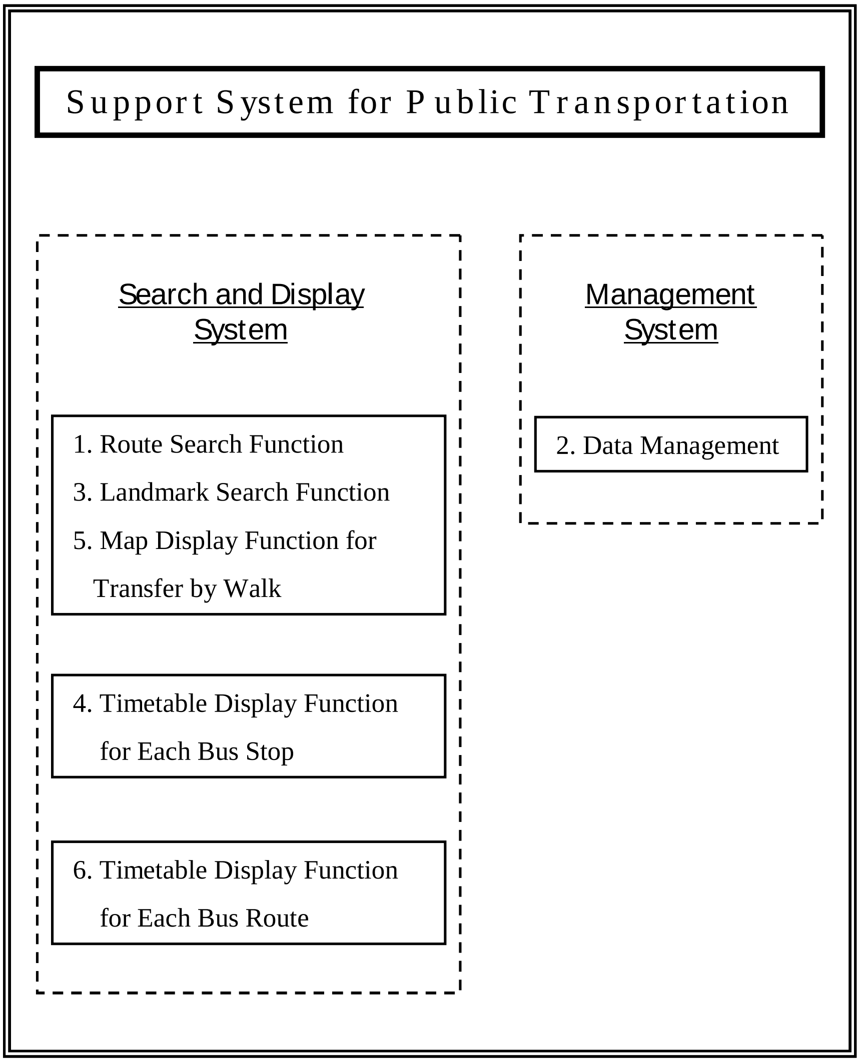

Figure 1.1 Taxonomy for functions of the support system

In Chapter 3 database for bus course search system is explained. All kinds of data for this system are introduced. And the table regarding special day and relationship of tables are mentioned. In Chapter 4 the course search technique used in this system is explained. Network, xml file and zone which is new concept for this system are introduced. The function which is the best practical parts of this study is explained in Chapter 5. These are map service function, time table service function as well as course search function. Experimentation and conclusion is described in Chapter 6 and 7.

Chapter 2

Characteristics of Bus Route

2.1 Movement by Walk

The bus routes are not designed to get the shortest path between two bus stops but to pass the places where as many passengers as possible can use. As a result, the routes are very much complicated and the distances between bus stops are rather short in general and, consequently, a passenger can get off a bus at a bus stop and walk to another bus stop of a different route to take a bus.

Accordingly, cases that movement by walk becomes effective exist.

At the Fig.2.1, O shows bus stop, a solid line shows route by bus and a dotted line shows course by walk. The case is that a person moves from bus stop A to bus stop H in the example of Fig. 2.1(a). There are 3 routes in this example that bus stop A, B, C, D is one route, bus stop E, F, G, H is other route and bus stop D, F is the other route. If a passenger moves from bus stop A to H only by bus, the only available path is A->B->C->D->H. However, if he can move by walk between bus stops, the path A->B->F->G->H may be available because of short distance between B and F. It becomes possible to include the way that a person moves from A to B by bus, moves from B to F by walk and moves from F to H by bus in search targets. At this point, the course including movement by walk isn't always the best way. But we can choose which of these two ways. Distances between rail road stations are too far for passenger to move by walk generally and, therefore, this kind of problem can be regarded as a unique problem in searching bus routes[2].

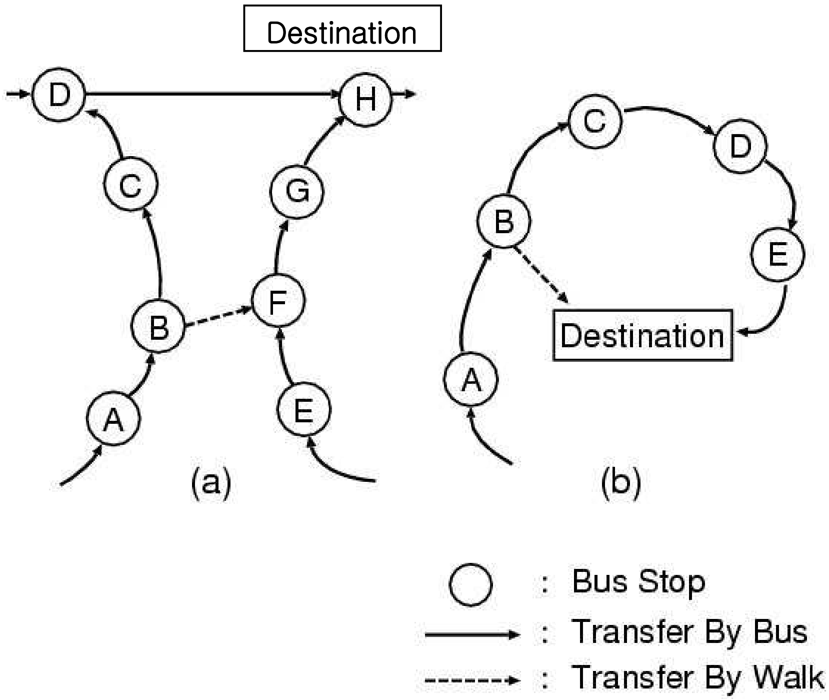

Figure 2.1 The example of the movement by walk

Besides, in some cases, it can be considered that the fastest way to a destination is not necessarily the nearest bus stop to the destination. In other words, it may be faster to get off at a bus stop, which is not the nearest one, and move by walk to the destination. As illustrated in Fig. 2.1(b), the nearest bus stop to the destination is E. If we do not consider the movement by walk, the only available route will be A->B->C->D->E->destination. However, if the movement by walk is considered, the route A->B->destination can be included in the target routes for search.

In the same manner, it must be considered that a passenger moves by walk to a bus stop other than the nearest one and takes a bus at the starting point.

On the contrary to the railroad route search problem, in which only the paths between the stations are considered, routes between the real starting point and the real destination must be considered in the bus route search problem.

Starting bus stop and arriving bus stop are not certainly the nearest one from starting point and destination because there are cases like Fig. 2.1(b).

2.2 Direct Bus Service

The numbers of bus stops may be reduced to the direct line even in the same bus route due to the difference of starting time or the difference of the day of a week. In such a case, the direct line will be treated as a new route which differs from the original bus route. For example, if a bus in a certain bus route **α** stops at the bus stops A and C generally except the starting time zone around 8:00 in which it stops at the bus stops A, B, and C, it's supposed that two different bus routes **α-1** and **α-2** exist.

2.3 Plural Bus Stops Having the Same Name

The examples of bus stops arrangement are shown at the Fig.2.2. Almost bus stops are established two bus stops both sides of the street like as Fig. 2.2(b). But rarely, bus stop is established one side of the street like as Fig. 2.2(a) or, three or four bus stops are established near the large intersection like as Fig. 2.2(c) and (d).

Therefore bus stops are indistinguishable from their name.

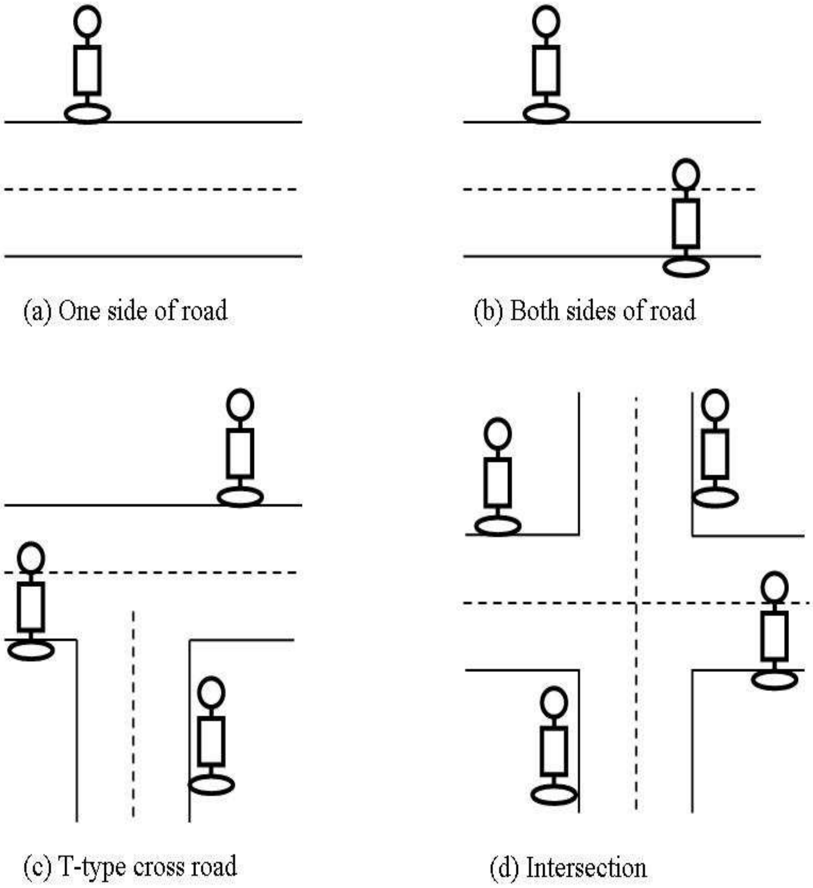

Figure 2.2 The example of bus stops arrangement

Chapter 3

Database for Optimum Bus Route Search System

3.1 Table for Course Search

3.1.1 Bus stop table

Bus stop table manages the table of bus stops. This table has made by using the data provided from Iconsamato Company. The record of table is composed of “ID”, “name” indicating name of bus stop, “north” indicating the north latitude of position and “east” indicating the east longitude of position.

The example of bus stop table is shown in Fig. 3.1

ID	name	north	east
201	郷原	35.15.30.02	134.16.29.82
202	局前	35.24.36.58	134.09.37.78
203	玉鉾公民館前	35.27.34.12	134.17.37.62
204	金屋	35.19.19.70	134.12.00.01
205	金原	35.26.44.61	134.08.32.76
206	金沢	35.29.45.81	134.07.59.44

Figure 3.1 The example of bus stop table

3.1.2 Connect information table

Connect information table manages the connect information between bus stops. Connect information expresses the contact between bus stops to compose bus routes. The record of table is composed of “ID”, “bus_stop_ID” indicating the bus stop to stop, “connect_ID$next” indicating the next connect information ID, “time” indicating the required time from former bus stop, “dest” indicating the destination of separate bus route and “via” indicating the place passed through. “connect ID$next” is ID of this table record and [-1] is regarded the last stop of one separate route.
In Fig. 3.2, a part of connect information table is shown.

ID	bus_stop_ID	connect_ID$next	time	dest	via
156	53	157	0	-	-
157	52	158	0	-	-
158	581	159	0	-	-
159	580	160	0	-	-
160	579	161	0	-	-
161	175	162	0	-	-
162	175	163	0	-	-
163	746	164	0	-	-
164	828	165	0	-	-
165	546	166	0	-	-
166	387	167	0	-	-
167	676	-1	0	-	-
7556	436	7557	1	鳥取駅(バス停)	
7557	81	7558	1	鳥取駅(バス停)	
7558	741	7559	2	鳥取駅(バス停)	
7559	363	7560	1	鳥取駅(バス停)	
7560	174	7561	1	鳥取駅(バス停)	
7561	387	7562	2	鳥取駅(バス停)	
7562	676	-1	2	鳥取駅(バス停)	
12377	675	12378	0	智頭駅	
12378	215	12379	8	智頭駅	
12379	215	12380	1	智頭駅	
12380	629	-1	17	智頭駅	

Figure 3.2 The example of connect information table

3.1.3 Route table

Route table manages the connection of running route. The record of table is composed of “ID”, “name” indicating the name of route, “company” indicating the name of bus company and “connect_ID$first_bus_stop” indicating the ID of connect information table being the starting point. In route table, to know the running sequence of all the bus stops in one running route, connection information of running route is managed.
In Fig. 3.3, a part of route table is shown.

3.1.4 Separate route table

Separate route table manages data that running routes subdivide to separate routes. The record of table is composed of “ID”, “connect_ID$first_bus_stop” indicating the ID of connect information table being starting point, “school_closed” indicating the workday in closure of a school, “school_open” indicating the workday in opening of a school, “sun” indicating the Sunday, “odd_sat” indicating the odd Saturday, “even_sat” indicating the even Saturday and “rosen_ID” indicating the ID of running route table including this separate route. The field indicating the running day is [1] in case of running and [0] in case of no running.
In Fig. 3.4, a part of separate route table is shown.

ID	name	company	connect_ID$first_bus_stop
1	桜谷・面影循環(上)	日本交通	1
2	市立病院(上)	日本交通	38
3	桜谷団地(下)	日本交通	48
36	神戸・横枕(下)	日ノ丸	1064
37	西郷・散岐(下)	日ノ丸	1112
38	逢坂(下)	日ノ丸	1190
92	智頭急行(上)	智頭急行	3162
93	智頭急行(下)	智頭急行	3165
94	若桜鉄道(上)	若桜鉄道	3168
95	若桜鉄道(下)	若桜鉄道	3181
98	因美(上)	JR西日本	3220
99	因美(下)	JR西日本	3234

Figure 3.3 The example of route table

ID	connect_ID$first_bus_stop	school_closed	school_open	sun	odd_sat	even_sat	rosen_ID
232	8246	1	1	1	1	1	52
233	15051	1	1	0	0	0	52
235	15064	1	1	0	0	0	52
237	8321	1	1	1	1	1	53
238	8357	1	1	1	1	1	53
240	8357	1	1	0	0	0	53
241	8401	1	1	1	1	1	54

Figure 3.4 The example of separate route table

3.1.5 Departure time table of starting bus stop in a route

Departure time table of starting bus stop manages the starting bus stop and the departure time. The record of table is composed of “rosen_ID” indicating ID of separate route table and “time” indicating the departure time. “time” is the value that hours are changed to minutes. In the change, the time is $\mathbf{x}$ hours $\mathbf{y}$ minutes and the value changed is $\mathbf{z}$.

$$\mathbf{z} = \mathbf{x} \times 60 + \mathbf{y}$$

In Fig. 3.5, a part of Departure time table of starting bus stop is shown.

route_ID	time
10	728
11	508
12	470
12	490
12	510
12	570
12	830
12	870
12	930
13	452
13	810
13	940

Figure 3.5 The example of departure time table of starting bus stop

3.2 Landmark

3.2.1 Table design of landmark

First of all, landmark has sorted in large group, middle group and small group. And landmark name and position information are inputted in the landmark table, classification name is inputted in classification information table, the code corresponding to classification is inputted in classification code table and the landmark corresponding to code in landmark classification table.
The tables about landmark made newly are explained below.

3.2.2 Landmark table

Landmark table manages the data of each landmark. The record of table is composed of “ID”, “name” indicating the name of landmark, “yomi” indicating the pronunciation of the landmark name, “address” indicating the address, “north” indicating the north latitude, “east” indicating the east longitude, “frequency” indicating the frequency of used in search and “visible” indicating whether to output the landmark at the search result or not.
In Fig. 3.6, a part of landmark table is shown.

3.2.3 Classification information table

Classification information table manages the classification sorting landmark. The landmark is divided in [large group], [middle group] and [small group]. The record of table is composed of “ID”, “level” indicating the classification level, “name” indicating the classification name and “yomi” indicating the pronunciation of classification name. “level” has one value in the [1] indicating large group, [2] indicating middle group and [3] indicating small group.
In Fig. 3.7, a part of classification information table is shown.

ID	name	yomi	address	north	east	frequency	visible
462	ローソン白兎海岸店	NULL	鳥取県鳥取市白兎６９３-５５	35.31.20.05	134.07.10.37	0	1
716	鳥取大学	NULL	鳥取県鳥取市	35.30.48.10	134.10.30.50	0	1
722	ローソン鳥取大前店	NULL	鳥取県鳥取市湖山町北２-１１０	35.30.55.31	134.10.35.31	0	1
1791	鳥取駅(バス停)	NULL	鳥取県鳥取市東品治町	35.29.29.79	134.13.42.11	0	1
1822	鳥取駅(ＪＲ 山陰本線)	NULL	鳥取県鳥取市東品治町	35.29.27.11	134.13.43.09	0	1

Figure 3.6 The example of landmark table

ID	level	name	yomi
152	1	ショッピング	NULL
154	2	アウトレット・ショッピングモール	NULL
155	2	ファッション・美容	NULL
156	3	メガネ	NULL
157	3	靴・衣料品	NULL
160	3	アクセサリー	NULL
162	2	生活	NULL
163	3	ディスカウントショップ	NULL
164	3	コンビニ	NULL

Figure 3.7 The example of classification information table

3.2.4 Classification code table

Classification code table manages the number of classification code. All the positions have one classification code from large group to small group. The record of table is composed of “sector_ID”, “subsector_ID”, “group_ID”.
In Fig. 3.8, a part of classification code table is shown.

3.2.5 Landmark classification table

Landmark classification table makes connections each landmark and the number of classification code. The record of table is composed of “code” indicating the number of classification code and “landmark_ID” indicating landmark. The ID of classification number table is saved in “code” and the ID of landmark table is saved in “landmark_ID”.
In Fig. 3.9, a part of landmark classification table is shown.

3.2.6 No-registration landmark table

No-registration landmark table manages the data which is not in landmark table but which user has searched. The record of table is composed of “name” field indicating the keyword. In search of landmark, if user searches the keyword not to be in landmark, the keyword can be added to landmark later.
In Fig. 3.10, a part of no-registration landmark table is shown.

ID	sector_ID	subsector_ID	group_ID
11002	152	154	0
11003	152	155	156
11004	152	155	157
11007	152	155	160
11009	152	162	163
11010	152	162	164

Figure 3.8 The example of classification code table

CODE	landmark_ID
11003	997
11004	899
11004	1017
11007	1690
11007	1954
11010	162
11010	462
11010	722
11010	3459

Figure 3.9 The example of landmark classification table

name
工学部
農学部
地域教育学部

Figure 3.10 The example of no-registration landmark table

3.3 Table of Special Running Day

3.3.1 Special running day

Special running day is out of the ordinary day like New Year's Day, the end of the year, Thanks-giving Day etc. The data of special running day is very tiny part of all transport data. But we cannot ignore the data of special running day to regard the practical use.

3.3.2 Special running information table

Special running information table manages a day of the week doing special running and route information. The record of table is composed of "ID", "bus_rosen" indicating the name of running bus route, "train_rosen" indicating the name of running railroad line, "term_start" indicating the start of term, "term_end" indicating the end of term, "start_bs" indicating the starting bus stop, "bus_base" indicating the time of starting bus stop or the time of special running , "tarin_base"

ID	bus_rosen	train_rosen	term_start	term_end	bus_base	train_base	start_bs	start_time	end_time	erase
1	NULL	98	1	1	NULL	0	675	448	NULL	1
2	NULL	99	1	1	NULL	0	866	501	NULL	1
3	0	-1	2	2	3	NULL	NULL	NULL	1080	1
4	NULL	101	3	3	NULL	2	629	503	NULL	1
6	NULL	100	3	3	NULL	2	736	459	NULL	1

Figure 3.11 The example of special running information table

indicating the day of the week being train base and “erase” indicating whether add or deletion. The ID of route table which is changed is saved in “”bus_rosen” and “train_rosen”.
In Fig. 3.11, a part of special running information table is shown.

3.3.3 Period table

Period table manages the period of special running day. The record of table is composed of “ID”, “type” indicating whether terms or special day, “month” indicating what month, “day” indicating what day, “week” indicating what week and “wday” indicating a day of the week.

In “type”, [0] is saved in the case of term and [1] is saved in the case of specially fixed day. If “type” is [0], the values are saved in “month” and “day”. If “type” is [1], the values are saved in “week” and “wday”. That Sunday is [0], Monday is [1], Tuesday is [2], Wednesday is [3], Thursday is [4], Friday is [5] and Saturday is [6] are saved in “wday”.

In Fig.3.12, a part of period table is shown.

3.4 User Management Table

User management table manages the information of user to be able to change the data. The record of table is composed of “ID” indicating the name of user, “pass” indicating the password and “company” indicating the company where the user works.
In Fig.3.13, a part of user management table is shown.

ID	type	month	day	week	wday
1	0	12	30	NULL	NULL
2	0	12	31	NULL	NULL
3	1	0	NULL	3	0

Figure 3.12 The example of period table

id	pass	company
nikkou	bus	日本交通
hinomaru	bus	日ノ丸
jr	train	JR西日本
wakasatetudou	train	若桜鉄道
tizukyuukou	train	智頭急行
keisana	mysql	all

Figure 3.13 The example of user management table

3.5 Relationship of Each Table

The relationship of tables about course search is displayed in Figure 3.14, relationship of tables about landmark is displayed in Figure 3.15 and the relationship of tables about special running period is displayed in Figure 3.16[3].

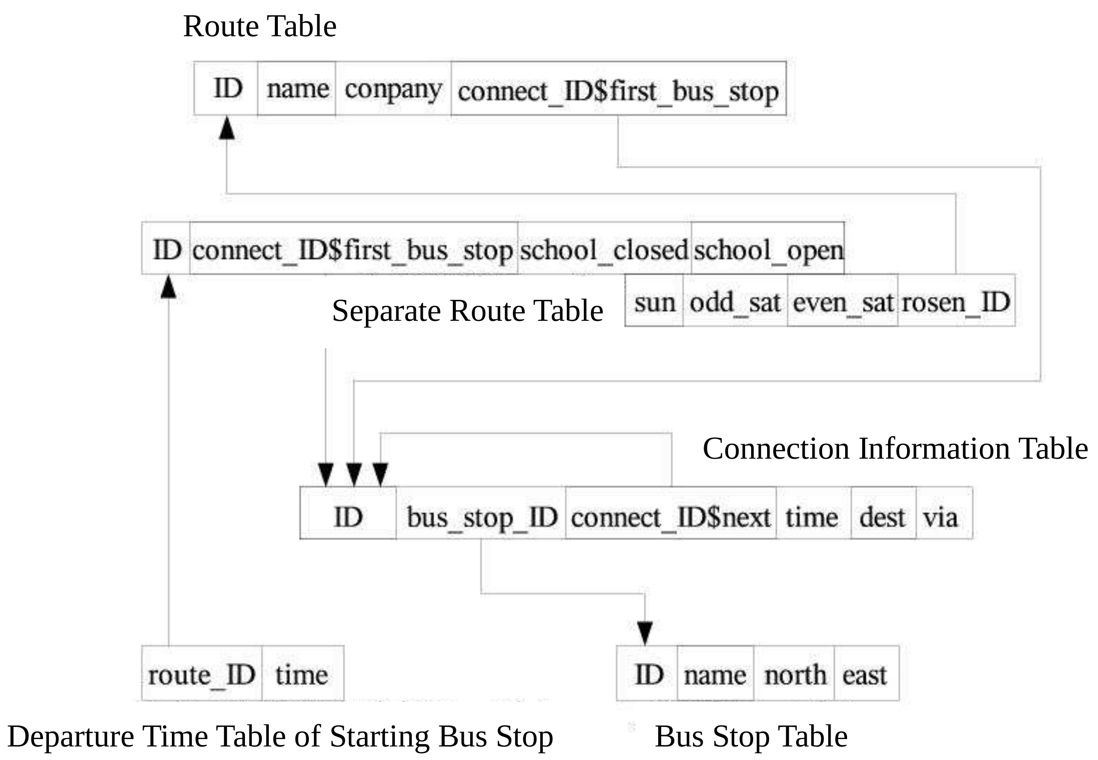

Figure 3.14 The relationship of tables about course search

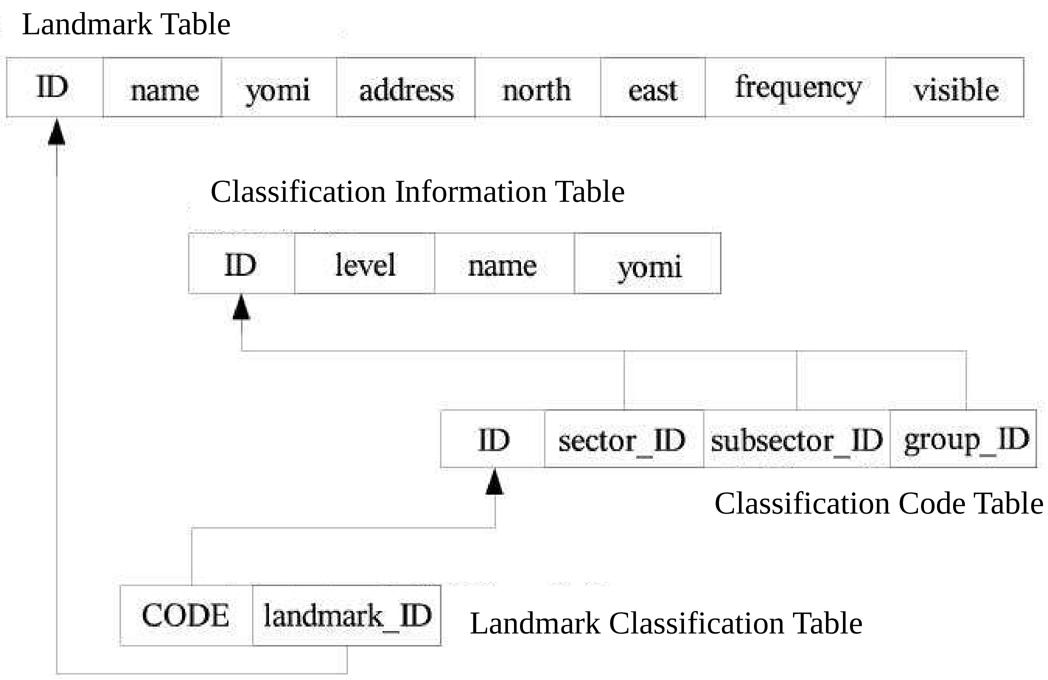

Figure 3.15 Relationship of tables about landmark

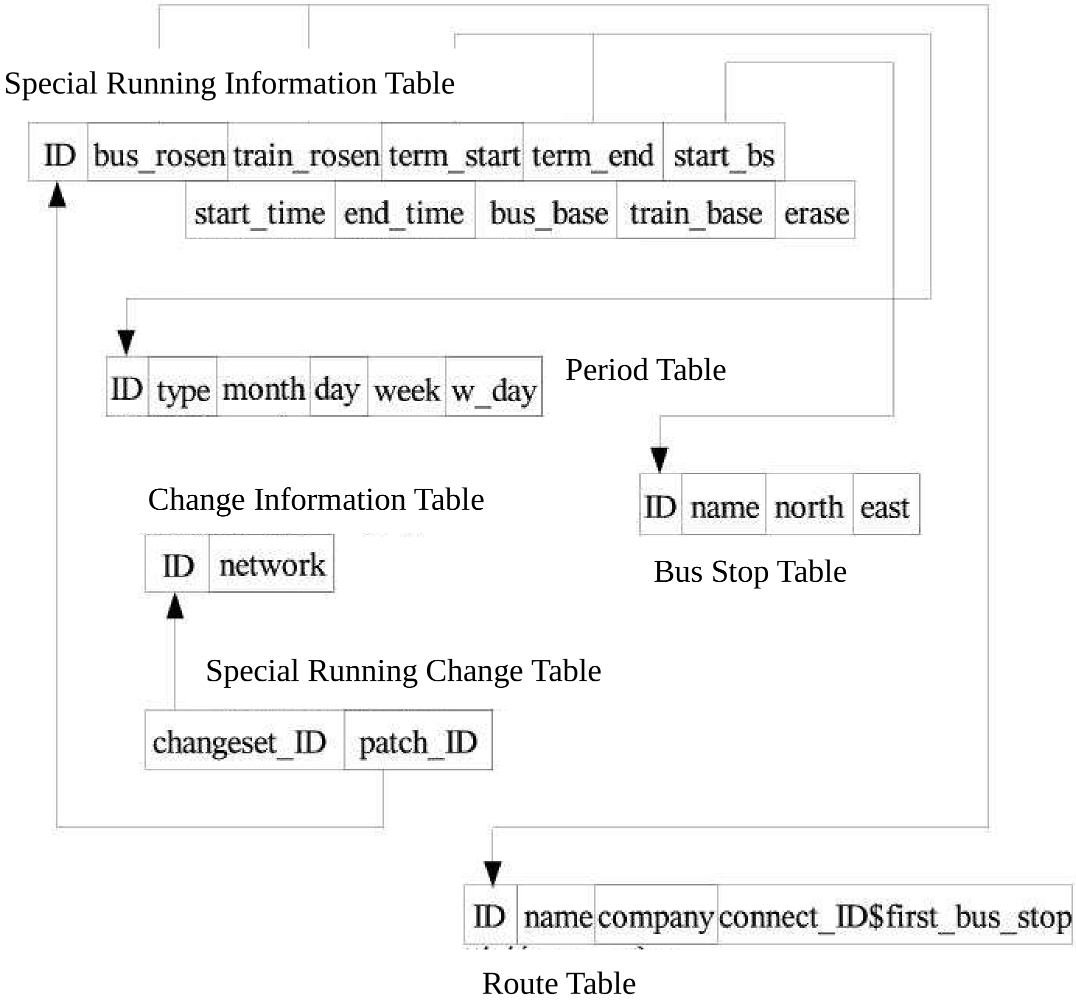

Figure 3.16 The relationship of tables about special running period

Chapter 4

Course Search Technique

4.1 Making out the Network

4.1.1 The order of making network

The network is made for carrying out course search and the way is shown in Figure 4.1. First, xml files are made from MySQL database. Then, the files displaying network and time schedule of all bus stops are made by using xml files.

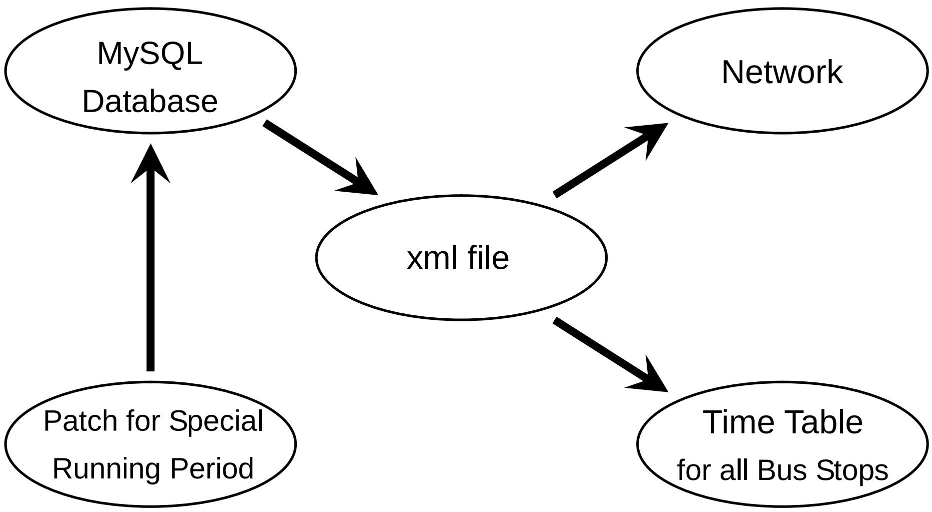

Figure 4.1 The order of making network

4.1.2 Making xml files of all separate routes

The files that indicate network and time table of all bus stops are made by using xml files after xml files have made. Namely, xml files are made to construct the network and to get time table of all bus stops. In xml files, there are bus company name of all separate bus routes, bus route names, the name of bus stop where the separate bus route stops, the time required between bus stops, destination and the place of passing through.

In Fig. 4.2, the example of xml file is shown.

```xml
<?xml version="1.0" encoding="euc-jp"?>
<路線>
<header>
<バス会社名>日本交通</バス会社名>
<路線名>市立病院(上)</路線名>
</header>
<?xml version="1.0" encoding="euc-jp"?>
<路線>
<header>
<バス会社名>日本交通</バス会社名>
<路線名>市立病院(上)</路線名>
</header>
<body>
<バス停><バス停名>市立病院</バス停名><所要時間>0</所要時間><行先>鳥取駅(バス停)</行先><経由></経由></バス停>
<バス停><バス停名>的場</バス停名><所要時間>1</所要時間><行先>鳥取駅(バス停)</行先><経由></経由></バス停>
<バス停><バス停名>宮長</バス停名><所要時間>2</所要時間><行先>鳥取駅(バス停)</行先><経由></経由></バス停>
<バス停><バス停名>大覚寺</バス停名><所要時間>1</所要時間><行先>鳥取駅(バス停)</行先><経由></経由></バス停>
<バス停><バス停名>吉成二丁目</バス停名><所要時間>1</所要時間><行先>鳥取駅(バス停)</行先><経由></経由></バス停>
<バス停><バス停名>南中前</バス停名><所要時間>1</所要時間><行先>鳥取駅(バス停)</行先><経由></経由></バス停>
<バス停><バス停名>東富安</バス停名><所要時間>1</所要時間><行先>鳥取駅(バス停)</行先><経由></経由></バス停>
<バス停><バス停名>末広日交前</バス停名><所要時間>2</所要時間><行先>鳥取駅(バス停)</行先><経由></経由></バス停>
<バス停><バス停名>農協会館前</バス停名><所要時間>1</所要時間><行先>鳥取駅(バス停)</行先><経由></経由></バス停>
<バス停><バス停名>鳥取駅(バス停)</バス停名><所要時間>2</所要時間><行先>鳥取駅(バス停)</行先><経由></経由></バス停>
</body>
</路線>
```

Figure 4.2 The example of xml file

4.1.3 The network to carry out the course search

To carry through the course search, the network is constructed with nodes which denote bus stops and train stations, arcs which denote separate routes from xml files. If then plural arcs come out from a node, the conditions of lopping off branch should not be used in search for all routes. So, bus stops are treated as different routes in all separate routes.

The weight of arc is the required time from one node to the other node. When we search the route, the waiting time is added to the required time dynamically.

4.1.4 About zone

A zone is made to find an virtual bus stop which a movement by walk is possible from a certain point. It is constructed with the horizon which denotes an latitude and the vertical which denotes longitude and Tottori prefecture is divided by it as a space of 5 minutes walk. When constructing the network, the positions of a certain imaginary bus stop and an accessible imaginary bus stop by walk in 5 minutes in its near 8 zones are connected by zone. In the case of searching the network, the search expands as 8 near zones (level 1), 16 near zones (level 2), 32 near zones (level3) until when an imaginary bus stop is found from the coordinates of a staring point and a destination. If a virtual bus stop is found, the system searches for one more level and memorizes all the bus stops in the zone. If movement time by walk to the nearest bus stop puts as ‘n’, only the accessible bus stops within 5*(n/5 +1) are connected with a starting point or a destination. Therefore, it becomes easy to find the reachable bus stop by walk by making “the zone”

4.1.5 Connection between bus stops movable by walk

To build the network, the zone mentioned in former chapter is used in finding out the bus stop accessible from a certain bus stop in 5 minutes by walk. It is enough by only finding 8 near zones of the zone including a certain bus stop.

Ta:b, required time(minutes) from bus stop A to bus stop B by walk is as follow.

In this calculation, x is difference of latitude, y is difference of longitude, m is the integer that latitude and longitude are altered to distance[meter], r is the integer that straight distance are altered to road distance and v is the integer of speed by walk. We apply that m is (14.0/45), r is 1.25, v is 50.

$$x = A_{lon} - B_{lon} \tag{4.1}$$

$$y = A_{lat} - B_{lat} \tag{4.2}$$

$$T_{A:B} = \left| \sqrt{x^2 + y^2} \times m \times r / v \right| \tag{4.3}$$

4.2 The Algorithm of Course Search

4.2.1 The outline of course search algorithm

In course search,

1. Starting position and destination are added to the network prepared.
2. By using the Dijkstra’s Algorithm, we can get the shortest time and the number of switching to destination.
3. The values got from the Dijkstra’s Algorithm method is used as conditions of lopping off branch. So, search for all routes is done by using the conditions.

4.2.2 Connection between starting position and destination

At the time of route search, starting position and destination must be added dynamically to the network prepared. This system conducts steps by zone as follow.

1. After calculating the zone including a starting position and a destination, from the zone until finding the bus stop, the search expands as 8 near zones(level 1), 16 near zones(level 2), 32 near zones(level3).
2. If bus stop is found, the system searches for one more level and memorizes all the bus stops in the zone.
3. If movement time by walk to the nearest bus stop puts as ‘n’, only the accessible bus stops within 5*(n/5 +1) are connected with a starting position or a destination.

4.2.3 Dijkstra method

The calculation time in Dijkstra Algorithm[4] depend on the calculation time for scanning the node of the shortest distance the ones which are connected to a certain node (Extract-Min operation, hereinafter).

Currently, the Priority Queue utilizing Fibonacci heaps is the fastest one among the data structures that have been utilized in this operation. In the priority queue, the calculation amount for one Extract-Min operation is O(log n) when there are n nodes. On the other hand, in this system, we do not utilize the Fibonacci Heaps but the data structure called bucket method[5]. In this method, queues are prepared for every value which is expected to be the shortest path, and the calculation amount is O(1) for one Extract-Min operation. However, the memory consumption by this method is increased with the number of expected shortest paths; it is useful only for cases where the expected shortest paths are limited under a certain number.

The bus route search problem is adequate problem for the bucket method because the expected shortest path is given in the unit of minutes and, further more, the maximum time in minutes is limited under the longest transfer time by bus.

Input data into the system is composed with 3 data of departure time, starting point, destination as the fig.3. The starting point and destination are appointed by land mark. Important 2,085 facilities in Tottori prefecture are managed as land marks by MySQL. And staring point can be inputted by GPS when using the cell phone with GPS.

The operations corresponding to bucket are,

1. an addition of node
2. a movement of node
3. an elimination of node

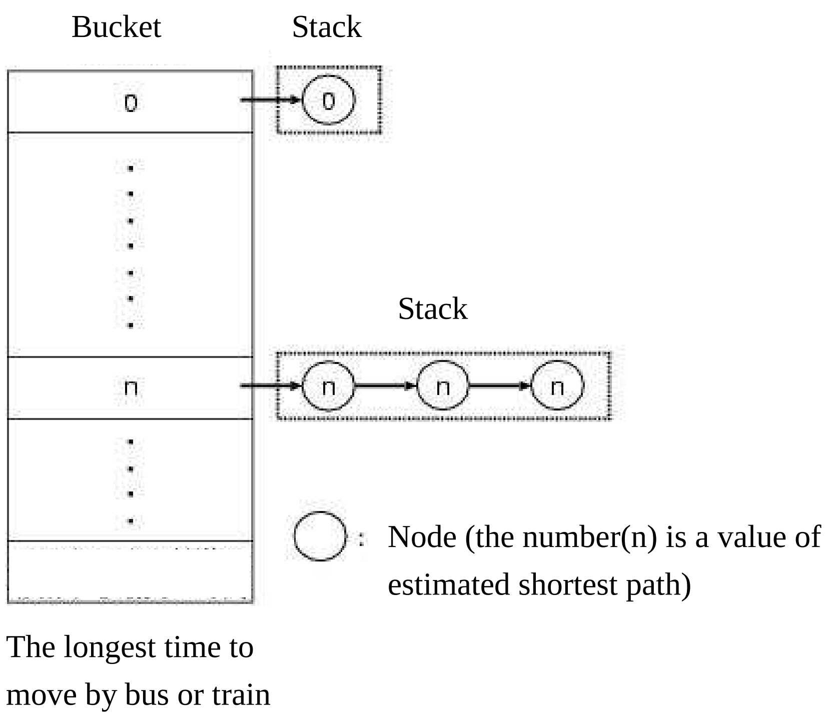

Figure 4.4 Bucket Method

Search is done by the Dijkstra Algorithm improved as follow.

1. Search is ended at the time that the estimated shortest course of node indicating a destination is determined.
2. Search in this system is regarding the dynamic weight of arc

4.2.4 Search for all routes

Search for all routes is an investigation of all routes from a starting position to a destination. The shortest course can be obtained by searching course of bus routes network by the Dijkstra Algorithm, but the search time may be too long due to huge amount of data. And there are many cases that courses having unnecessary transfer are found. Therefore the min-time, which is the shortest time from a starting position to a destination, is calculated by the Dijkstra Algorithm regarding a required time as a cost. And then search for all routes is done after regarding the transfer times of the obtained course as the max-transfer. It is possible of finding more practical course having less transfer times.
We developed a search method which can find the optimum route by search for all routes of bus again after limiting the search range with the route obtained from the preliminary search considering only the shortest course in time. The system stops when it finds the limiting conditions during the search operation. The limiting conditions (conditions of lopping off branch[6]) in this system are as follows;

1. The case that the number of bus transfer exceeds a setting number should be excluded.
2. Movement by walk from bus stop to bus stop in same separate route is not allowed.
3. It should be excluded if the sum of elapsed times and the numbers of transfer from a departure to a bus stop of a certain separate route exceeds the value of the false case in the past search.
4. It should be excluded if the elapsed time from the departure to the current point exceeds the estimated shortest time to current bus stop of the separate

route obtained by Dijkstra algorithm.

5. It should be excluded if the elapsed time from the departure to the current point exceeds the min-time.

4.2.5 Procedures of course search

Course search is conducted by the algorithm of Dijkstra and search for all courses. The procedures of course search are explained here. The principle of the procedures is that the optimum course is found as fast as possible.

In this point, let's describe transfer. Suppose that minimum 2 minutes are needed in a transfer. And transfer has conditions as follow.

1. The movement by walk between bus stops in same bus route is not allowed.
2. The movement by walk toward running direction of route is not allowed.
3. The time required in the movement by walk is limited in 5 minutes. And in case of movement toward train station, it is limited in 20 minutes.

Procedure A

1. A min_transfer, which is the minimum number of transfer from the starting position to the destination, is calculated by the Dijkstra algorithm.
2. Assume that a bus is delayed 2 minutes from the schedule.

Procedure B

Procedures of course search is,

1. The courses of the case that movement time by walk from a starting position and a destination to a bus stop is limited in 20 minutes, are acquired by procedure A.
2. If the courses that movement time by walk is over 5 minutes are included in from a starting position to a bus stop and from a destination to a bus stop in

the courses obtained at 1, the courses of the case that movement time by walk from a starting position and a destination to a bus stop is allowed until 5 minutes, are acquired by procedure A.

3. The rout set of the route search algorithm can be obtained from the rout sets which are resulted from the above operations 1 and 2, which has the minimum numbers of transfer and elapsed time.

According to course search algorithm, the practical course that required time is not the smallest one but people want as well as the course that required time is the smallest one and transfer time is the smallest one too, both of all are outputted.

Chapter 5

Function of Bus Course Search System

5.1 Course Search Function

5.1.1 Acquirable course

Starting position is designated as landmark or current position through the mobile phone having GPS function. The system outputs the courses as follow.

1. The course that the required time from a starting position to a destination is the shortest one, is outputted. If the number of the shortest course is plural, the course that the times of transfer is the smallest one in the shortest courses is outputted.
2. If there are the courses reducing transfer times though the required time lengthens to maximum 30 minutes,
3. At the case of courses gotten from 1 including transfer within 5minutes, the courses having a surplus time over 5 minutes in transfer, are outputted together even though the required time lengthens to maximum 30minutes.
4. At the case of courses gotten from 1 including movement over 5minutes by walk from a starting position to the first bus stop and from the last bus stop to a destination, the courses that movement by walk is within 5 minutes, are outputted together even though the required time lengthens to maximum 30minutes.
5. If to do walk from that position to a destination without the last transfer is faster than doing transfer, courses including the course are outputted.

Search time is very shortened and any course search can be finished within 1 second.

What the courses of 1~5 are acquirable practically is shown at the experimentation in chapter 6.

5.1.2 Departure time search & Arrival time search

Departure time is appointed as any day, any time in 3 months or any minutes after from present time. The example case of appointing optional day and time is shown in Figure 5.1 and the example case of 10minutes after now is shown in Figure 5.2.

If user is connected with the bus course search system, very first, the screen that user appoint day and time is shown as Fig. 5.1. Here, the next screen is shown that user click '設定' (setting) after inputting the data of day and time. And if '○までに到着する' (arrive by the time) is chosen and '経路探索' (course search) is clicked, use can get the course searched. Additional courses are presented in the below of screen of course searched. These are the course having sufficient time to transfer or having time shortened to move by walk etc. So, user can choose the course he wants.

If people appoint the day and time to arrive by the time, it is possible to search the arrival time. In Figure 5.3, the example of arrival time search is shown. Fig. 5.2 is same to Fig. 5.1. Arrive time is indicated in Fig. 5.1 and departure time is indicated in Fig. 5.2. If user decide '分後' (what minutes after) and '〇に出発する' (departure), the course he wants is searched.

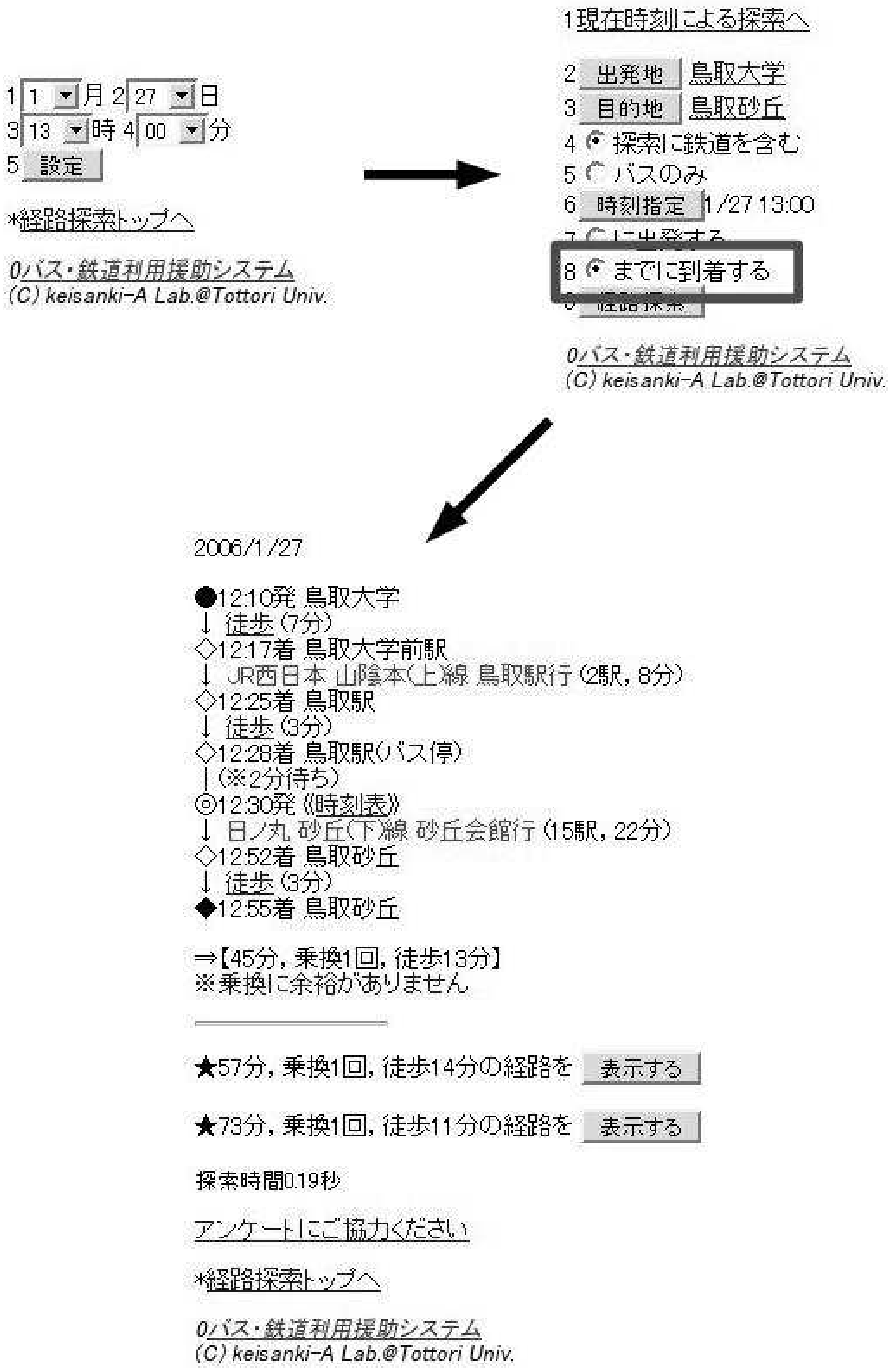

Figure 5.1 The example of departure time search (a designated day and time)

Figure 5.2 The example of departure time search (10minutes after now)

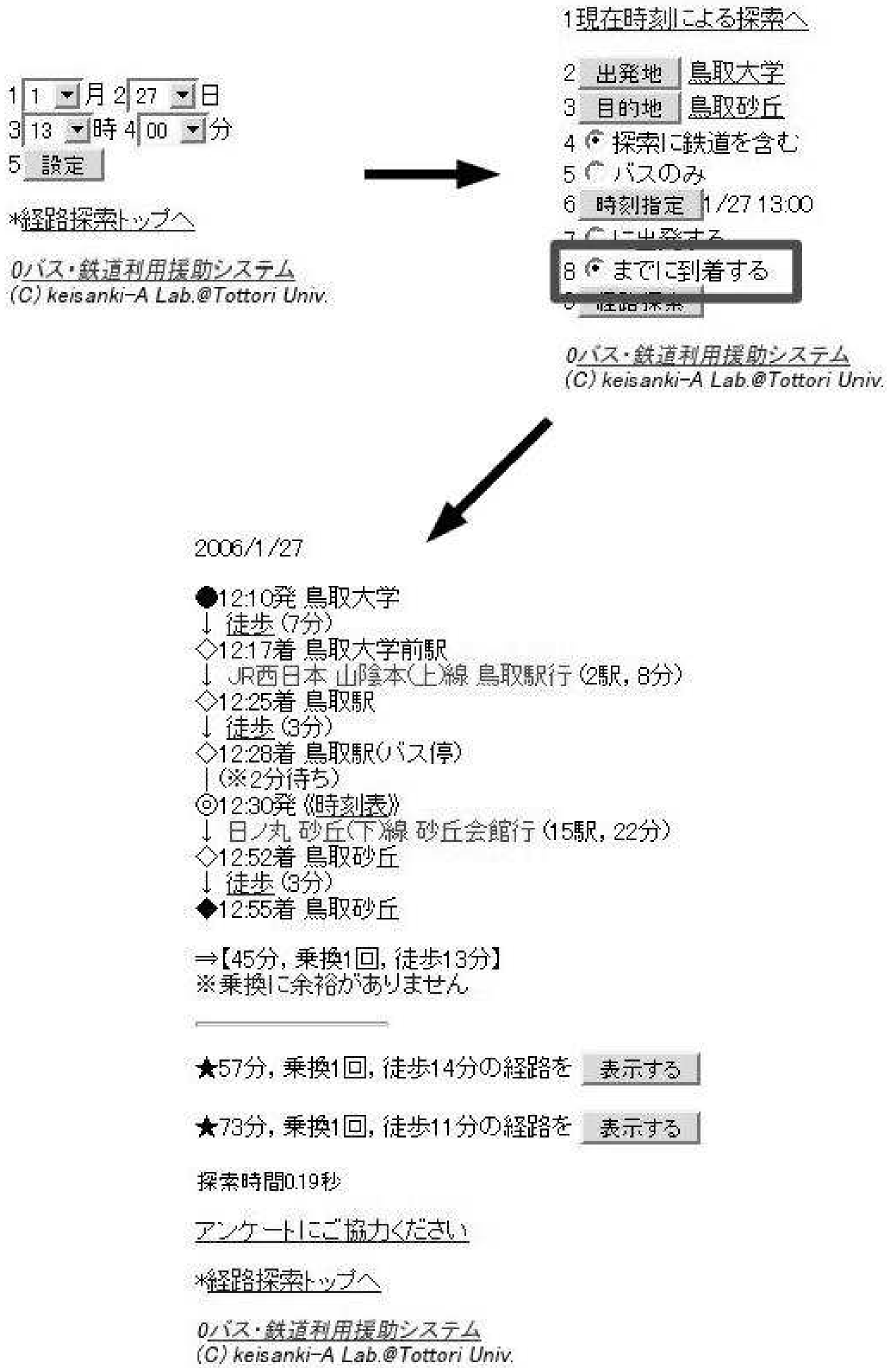

Figure 5.3 The example of arrival time search

5.2 Data Management Function

Data management function is what renews the data for the system. Data renewal is largely divided 2 kinds as running diagram renewal and landmark data change. MySQL database is used.

5.3 Landmark Search Function

In this system, currently, rapidity of position information is high and a name of position is almost accurate. So, practical value is very high. If people does input nothing, the system display all the search word (position name) in the order form search time is the largest one to the last one. In Fig. 5.4, the example of landmark search is shown.

5.4 Map Service Function for Movement by Walk

This system offers map service for movement from starting point to the first bus stop, from the last bus stop to destination by walk. And it also offers map service for movement by walk between bus stops during total course. If Fig. 5.5, the example of map service for movement by walk is shown.

5.5 Time Table Service Function of All the Bus Stops

People can select time table search at the top of the search system screen and can get the time table of riding bus. In Fig. 5.6, the example is shown. Furthermore, time table is displayed to divide each hour as 7 o’clock zone, 8 o’clock zone for convenience to see.

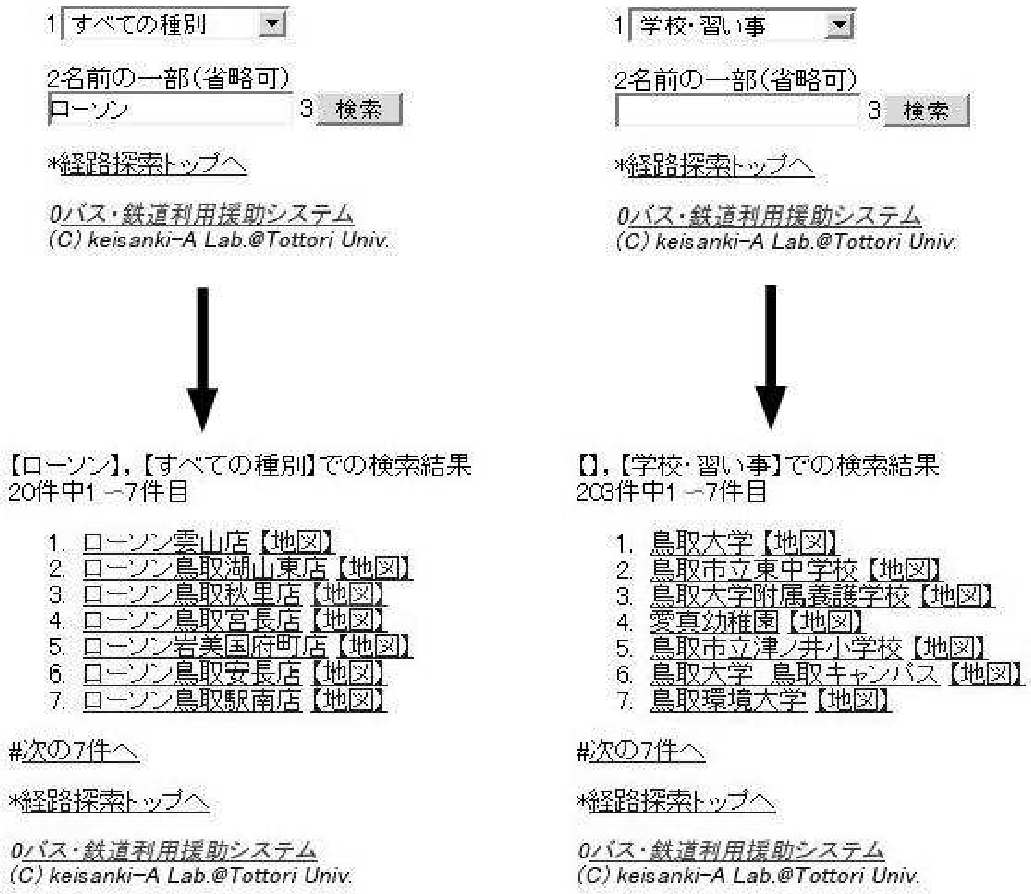

Figure 5.4 The example of landmark search

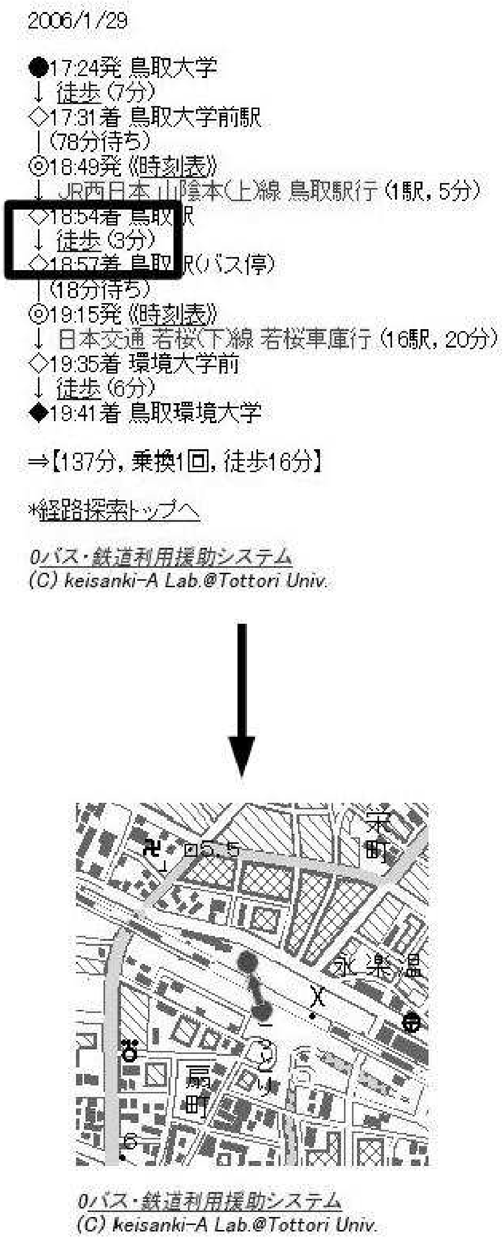

Figure 5.5 The example of map service for movement by walk

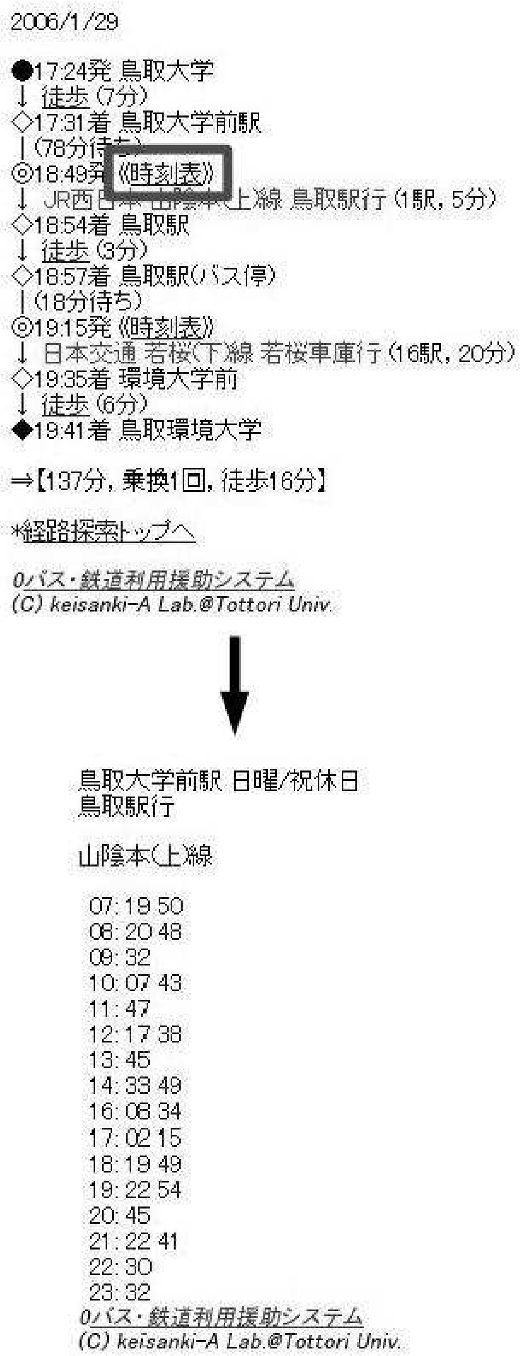

Figure 5.6 The example of time table service

Chapter 6

Experimentation

6.1 Environment

The calculator in this experimentation is that CPU is Pentium 4, 3.0GHz of Intel company, memory is 1GB and OS is turbolinux10 Desktop.

6.2 Validity of Lopping off Branches Conditions in All the Courses Search of Algorithm

Whether lopping off branches conditions is valid or not has experimented. A party of the time required at the time of arriving at the bus stop of any separate route and transfer times excesses the value of failed in other search.
We compare the search time of the case having no conditions with the case having lopping off branches conditions. In Table 6.1, the result is shown.

Setting	Without Conditions	With Conditions
2/3 10:00 dep. Tottori city hall ⇒ Tottori city AOYA	149.49[s]	0.67[s]
2/3 13:00 dep. Tottori city hall ⇒ Tottori city AOYA	34.90[s]	0.19[s]
2/3 13:00 arr. Tottori city hall ⇒ Tottori city AOYA	277.53[s]	0.53[s]
2/3 1:00 dep. Tottori University ⇒ Chizu police station	3.99[s]	0.40[s]
2/3 1:00 dep. Tottori University ⇒ Chizu police station	2.72[s]	0.41[s]
2/3 13:00 arr. Tottori University ⇒ Chizu police station	13.49[s]	0.35[s]

Table 6.1 Search time comparison “without conditions” with “with conditions”

The result of experimentation shows that the cases with conditions are 7~500 times faster than the cases without conditions. And search time of the cases without conditions is not regular. But search time of the case with conditions is nearly regular within 1 second.
According to this experimentation, the validity of lopping off branch conditions has confirmed.

6.3 Validity of Transfer Conditions

In here, the transfer conditions are :

1. It is not allowed to walk to bus stops of the same bus route.
2. It is not allowed to walk forward the advance direction of bus route.
3. The time of movement by walk is allowed to 5 minutes. But, in case of movement forward railroad station, it is allowed to 20 minutes.

6.3.1 Validity of the condition that people can be allowed to walk only to 20minutes to the railroad station

In Fig. 6.1 and 6.2, the example of comparing the case having a condition that people can walk only to 20minutes to the railroad station with the case having no condition are shown.

Fig. 6.1 is the course gotten of the case that movement forward railroad station by walk is also limited within 5minutes and Fig. 6.2 is the course gotten of the case that movement forward railroad station by walk is limited within 20minutes in the same conditions. Te course of Fig. 6.1 includes the transfer and the course of Fig. 6.2 includes the movement by walk instead of transfer. So, the latter method is more efficient because there is no bothering of transfer.

Consequently, it is more valid that the case of movement forward railroad station by walk is limited within 20minutes is known.

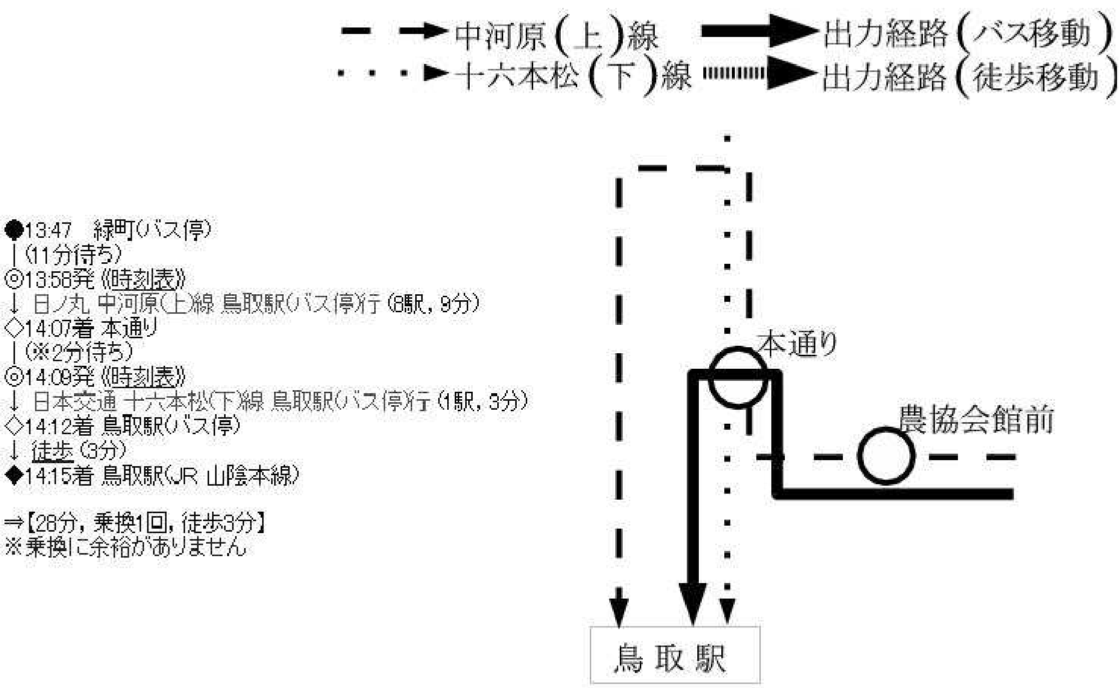

Figure 6.1 The example that movement forward railroad station by walk is also limited within 5minutes
(3rd February 13:47 緑町(バス亭) dep. 鳥取駅(JR) arr.)

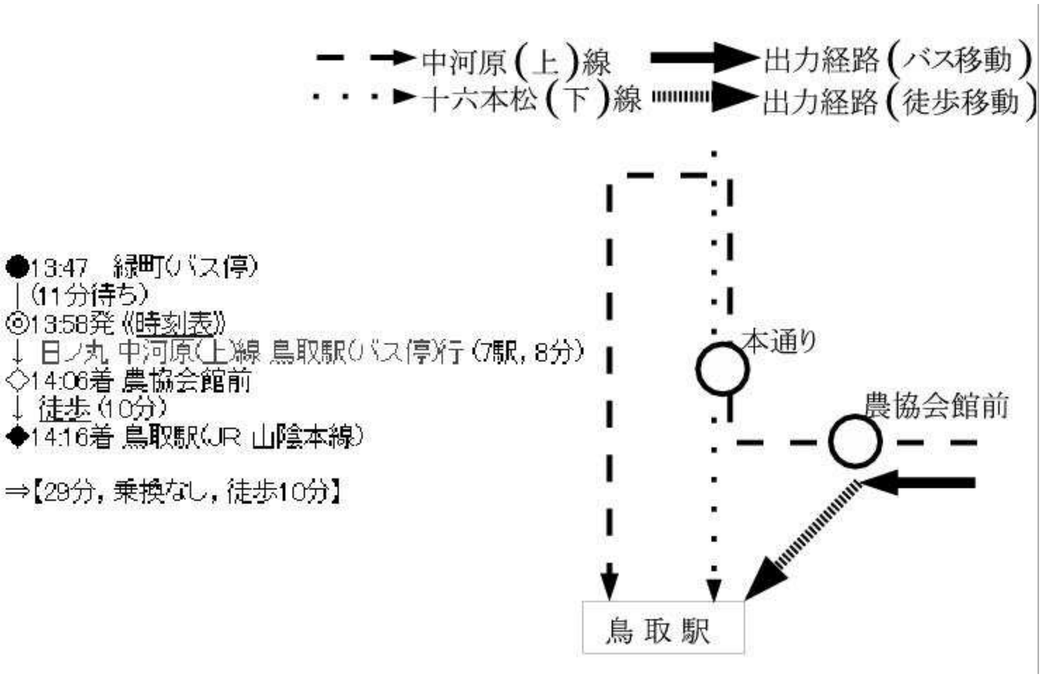

Figure 6.2 The example that movement forward railroad station by walk is limited within 20minutes
(3rd February 13:47 緑町(バス亭) dep. 鳥取駅(JR) arr.)

6.3.2 Validity of conditions that movement by walk toward the advance direction of bus is limited.

The results of that searches are done by using the second transfer condition that it is not allowed to walk forward the advance direction of bus route and by transfer by walk as possible are shown in Fig. 6.3 and Fig. 6.4. The course of Fig. 6.3 is practical. But the course of Fig. 6.4 is not practical. Because the course of Fig. 6.4 includes the unnecessary transfer at the bus stop of 太平線.

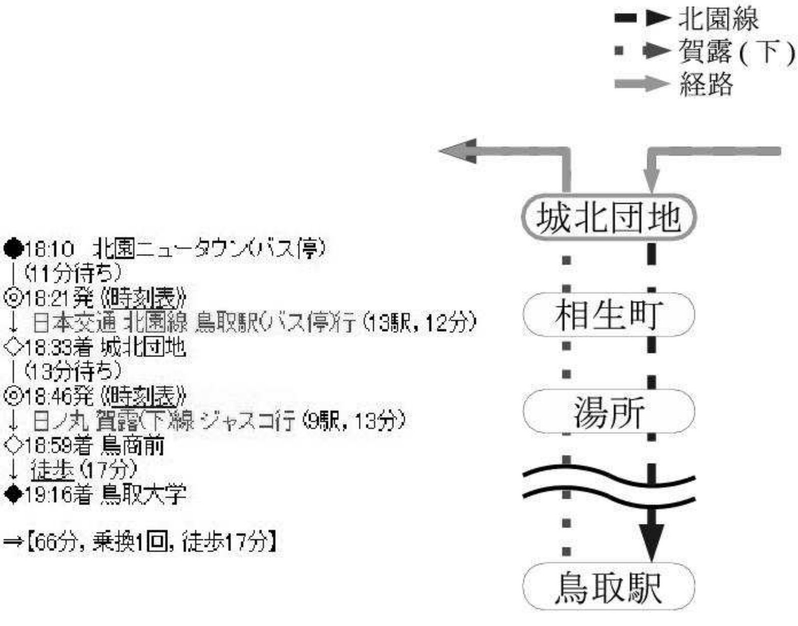

Figure 6.3 The example that it is not allowed to walk forward the advance direction of bus route and to do transfer by walk as possible (3rd February 18:10 北園ニュータウン(バス亭) dep. 鳥取大學 arr.).

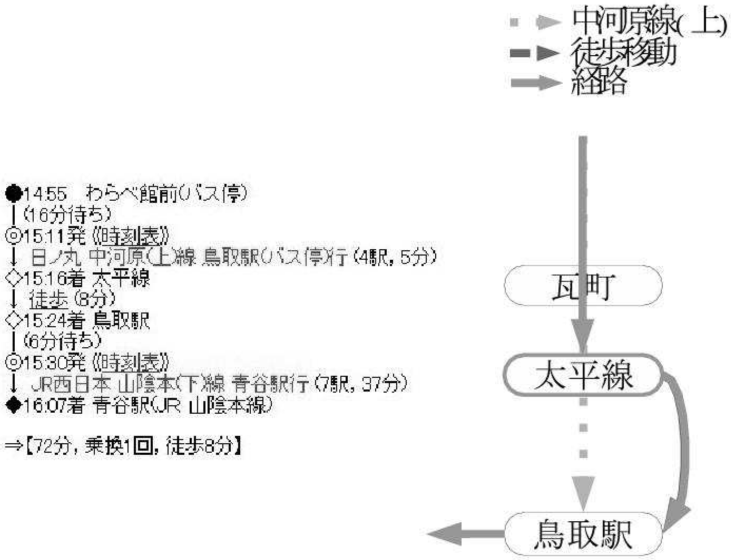

Figure 6.4 The example that it is not allowed to walk forward the advance direction of bus route and to do transfer by walk as possible (3rd February 14:55 わらべ館前 dep. 青谷駅 arr.).

The results of that searches are done by using the second transfer condition that it is not allowed to walk forward the advance direction of bus route and by transfer to bus as soon as possible are shown in Fig. 6.5 and Fig. 6.6. The course of Fig 6.6 is practical. But in the course of Fig. 6.5, we can transfer at any bus stop of 城北団地, 相生町 and 湯所. But doing transfer to the bus stop as soon as possible prevents to make detour and reduces the fare. The course of Fig. 6.6 is not practical owing to the above reasons.

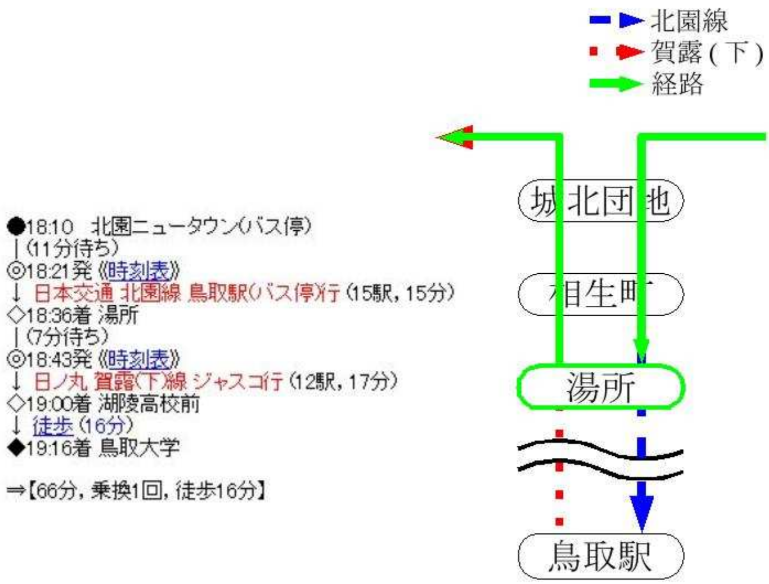

Figure 6.5 The example of a course gotten by the second transfer condition and transfer to bus as soon as possible
(3rd February 18:10 北園ニュータウン(バス亭) dep. 鳥取大學 arr.).

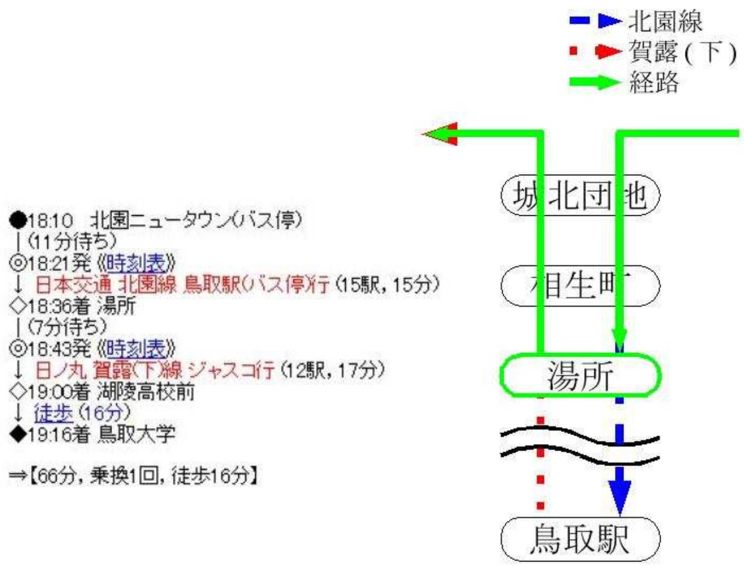

Figure 6.6 The example of a course gotten by the second transfer condition and transfer to bus as soon as possible
(3rd February 14:55 北園ニュータウン(バス亭) dep. 鳥取大學 arr.).

6.4 Practicality of Acquirable Courses in This System

6.4.1 Experimentation outline

The acquirable courses described in chapter 5.1.1 are:

6. The course that the required time from a starting position to a destination is the shortest one, is outputted. If the number of the shortest course is plural, the course that the times of transfer is the smallest one in the shortest courses is outputted.
7. If there are the courses reducing transfer times though the required time lengthens to maximum 30 minutes,
8. At the case of courses gotten from 1 including transfer within 5minutes, the courses having a surplus time over 5 minutes in transfer, are outputted together even though the required time lengthens to maximum 30minutes.
9. At the case of courses gotten from 1 including movement over 5minutes by walk from a starting position to the first bus stop and from the last bus stop to a destination, the courses that movement by walk is within 5 minutes, are outputted together even though the required time lengthens to maximum 30minutes.
10. If to do walk from that position to a destination without the last transfer is faster than doing transfer, courses including the course are outputted.

The experimentation is done to see whether there is a practicality in the acquirable courses or not.

6.4.2 Practicality of the course that the required time is shortest and transfer times become reduced

●13:47　緑町(バス停)
|(11分待ち)
◎13:58発《時刻表》
↓ 日ノ丸 中河原(上)線 鳥取駅(バス停)行 (8駅，9分)
◇14:07着 本通り
|(※2分待ち)
◎14:09発《時刻表》
↓ 日本交通 十六本松(下)線 鳥取駅(バス停)行 (1駅，3分)
◇14:12着 鳥取駅(バス停)
↓ 徒歩 (3分)
◆14:15着 鳥取駅(JR 山陰本線)

⇒【28分，乗換1回，徒歩3分】
※乗換に余裕がありません
※本通りから鳥取駅(JR 山陰本線)まで14分歩けば14:21に到着します

Figure 6.7 The required time is the shortest course.
(3rd February 13:47 緑町(バス亭) dep. 鳥取駅(JR) arr.)

●13:47　緑町(バス停)
|(11分待ち)
◎13:58発《時刻表》
↓ 日ノ丸 中河原(上)線 鳥取駅(バス停)行 (7駅，8分)
◇14:06着 農協会館前
↓ 徒歩 (10分)
◆14:16着 鳥取駅(JR 山陰本線)

⇒【29分，乗換なし，徒歩10分】

★34分，乗換なし，徒歩3分の経路を 表示する

★28分，乗換1回，徒歩3分の経路を 表示する

探索時間0.16秒

Figure 6.8 The times of transfer is the smallest course.
(3rd February 13:47 緑町(バス亭) dep. 鳥取駅(JR) arr.)

6.4.3 Practicality of the course that the required time is shortest and the time required in transfer is sufficient

●10:00発 鳥取県庁
↓ 徒歩 (2分)
◇10:02着 鳥取県庁
↓ 日ノ丸 100円バス(青)線 鳥取駅(バス停)行 (7駅, 5分)
◇10:07着 農協会館生協病院入口
↓ 徒歩 (4分)
◇10:11着 末広日交前
｜(※2分待ち)
◎10:13発 《時刻表》
↓ 日ノ丸 中河原(下)線 高岡行 (6駅, 7分)
◇10:20着 緑町
↓ 徒歩 (7分)
◆10:27着 鳥取市立東中学校

⇒【27分, 乗換1回, 徒歩13分】
※乗換に余裕がありません

★37分, 乗換1回, 徒歩14分の経路を 表示する

探索時間0.25秒

Figure 6.9 The required time is the shortest course.
(3rd February 10:00 鳥取県庁 dep. 鳥取市立東中学校 arr.)

●10:00発 鳥取県庁
↓ 徒歩 (2分)
◇10:02着 鳥取県庁
↓ 日ノ丸 100円バス(青)線 鳥取駅(バス停)行 (8駅, 6分)
◇10:08着 永楽温泉町
↓ 徒歩 (5分)
◇10:13着 末広日交前
｜(10分待ち)
◎10:23発 《時刻表》
↓ 日ノ丸 市内岩倉(下)線 岩倉行 (6駅, 7分)
◇10:30着 緑町
↓ 徒歩 (7分)
◆10:37着 鳥取市立東中学校

⇒【37分, 乗換1回, 徒歩14分】

Figure 6.10 the time required in transfer is sufficient
(3rd February 10:00 鳥取県庁 dep. 鳥取市立東中学校 arr.)

6.4.4 Practicality of the course that the required time is shortest and the course to reduce movement time by walk

●12:00発 鳥取砂丘
↓ 徒歩 (13分)
◇12:13着 砂丘東口
| (21分待ち)
◎12:34発 《時刻表》
↓ 日本交通 福部·岩戸(上)線 鳥取駅(バス停)行 (5駅, 7分)
◇12:41着 城北団地
| (15分待ち)
◎12:56発 《時刻表》
↓ 日ノ丸 鹿野·青谷(下)線 鹿野行 (7駅, 10分)
◆13:06着 湖山駅前(バス停)

⇒【66分, 乗換1回, 徒歩13分】

★86分, 乗換1回, 徒歩3分の経路を 表示する

探索時間0.19秒

Figure 6.11 The required time is the shortest course.
(3rd February 12:00 鳥取砂丘 dep. 湖山駅前(バス亭) arr.)

●12:00発 鳥取砂丘
↓ 徒歩 (3分)
◇12:03着 鳥取砂丘
| (57分待ち)
◎13:00発 《時刻表》
↓ 日ノ丸 砂丘(上)線 鳥取駅(バス停)行 (7駅, 11分)
◇13:11着 城北団地
| (5分待ち)
◎13:16発 《時刻表》
↓ 日ノ丸 賀露(下)線 ジャスコ行 (7駅, 10分)
◆13:26着 湖山駅前(バス停)

⇒【86分, 乗換1回, 徒歩3分】

Figure 6.12 The course to reduce movement time by walk
(3rd February 12:00 鳥取砂丘 dep. 湖山駅前(バス亭) arr.)

6.4.5 Practicality of the course to move by walk instead of the last transfer

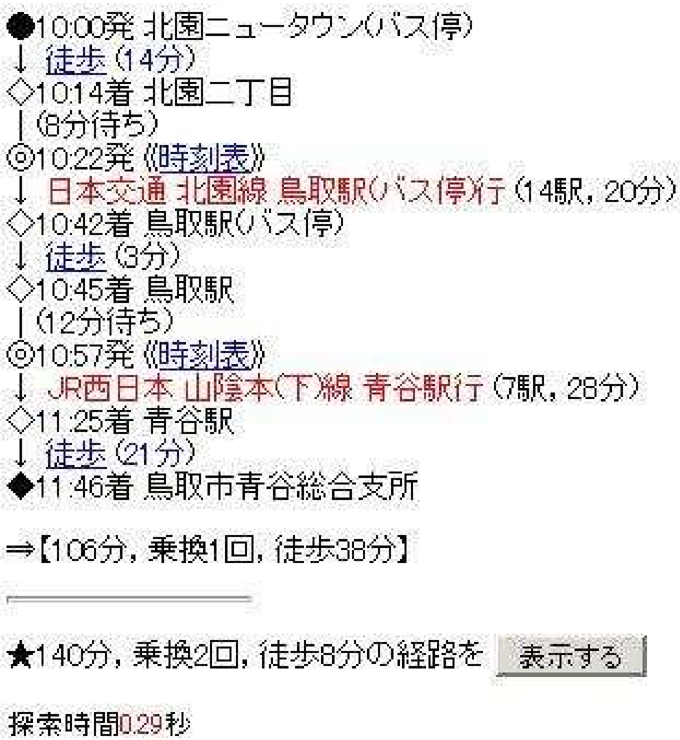

Figure 6.13 The course to move by walk instead of the last transfer.
(3rd February 10:00 北園ニュータウン(バス亭) dep. 鳥取市青谷総合支社 arr.)

●10:00　北園ニュータウン(バス停)
｜(17分待ち)
◎10:17発《時刻表》
↓ 日本交通 北園線 鳥取駅(バス停)行 (21駅，25分)
◇10:42着 鳥取駅(バス停)
↓ 徒歩 (3分)
◇10:45着 鳥取駅
｜(12分待ち)
◎10:57発《時刻表》
↓ JR西日本 山陰本(下)線 青谷駅行 (7駅，28分)
◇11:25着 青谷駅
↓ 徒歩 (1分)
◇11:26着 青谷駅(バス停)
｜(48分待ち)
◎12:14発《時刻表》
↓ 日ノ丸 勝部(桑原)(上)線 青谷町役場行 (3駅，2分)
◇12:16着 青谷町総合支所前
↓ 徒歩 (4分)
◆12:20着 鳥取市青谷総合支所

⇒【140分，乗換2回，徒歩8分】
※青谷駅(バス停)から鳥取市青谷総合支所まで20分歩けば11:46に到着します

Figure 6.14 The course to do the last transfer.

(3rd February 10:00 北園ニュータウン(バス亭) dep. 鳥取市青谷総会支社 arr.)

6.5 Search Time

The present system has accomplished through many times of revision work during the study period to develop the more efficient and practical algorithm.
When the first time, only the shortest course from Dijkstra’s Algorithm is pursued, the course gotten by Dijkstra’s Shortest Path Algorithm was incongruent in use actually. And what method of search is effective to deal with the huge data of bus routes and landmarks, etc. has studied. As a result of such effort, transfer conditions, conditions of lopping off branch and the concept to say ‘Zone’, etc. were introduced. Now, output the usable course actually and shortening of search time become possible.

All kinds of courses searched from this system can be outputted to computer screen or mobile phone screen in 1 second.

Chapter 7

Conclusion

A bus route search of practical speed becomes possible after that we improve Dijkstra’s Algorithm to calculate a weight of the arcs corresponding with arrival time at the arcs and to select high-speedily the next arc which has the value estimated as the shortest path.

Accordingly, we have developed the system having optimum bus route search f unction, data management function, landmark search function, map service for mov ement by walk and time table service for all the bus route. Possibility to print the m ore practical path that required time is shorter by considering movement by walk be tween bus stops in the search algorithm and that transfer time is much smaller by us ing lopping off branches conditions is known. That is to say, in this system we can g et the courses of several types which the user wants. We can choose any method am ong the shortest course, comfortable one with little transfer times and the course of small waiting time, etc. The user can choose the path he wants in accordance with g etting path not the shortest but intended for practical use including the shortest path.

Although the using the public transportation is necessary, bus routes are neglec ted because of the inconvenience of using bus. This bus route search system seeks t o the convenience of using bus and will provide the usefulness to the users. Further more the system is considered to be used in public and solve the problem of insuffic iency of public transportation.

After this, the research about optimum bus route system will be accomplished to the more detailed parts. For example, the place of passing through, the information about fare, etc may be reflected to this system. Currently, this system is opened to the public (http://www.ikisaki.jp/).

Reference

[1] Takao Kawamura, et.al: Path Planning System for Bus Network Including Walking Transfer, IPSJ Journal, Vol.46, No.5,pp.1207-1210 (2005).

[2] Genki, Kusugami ： 公共交通機關網の経路探索手法の開發とその鳥取市路線バス網への応用に関する研究, 鳥取大学大学院 工学研究科 知能情報工学専攻 (2005)

[3] Ryosuke, Kawasima ： 公共交通機關利用援助システムの開發に関する研究, 鳥取大学 工学部 知能情報工学科 (2006)

[4] Dijkstra, E.W.: A Note on Two Problems in Connection with Graphs, Numerische Math.1,pp.269-271 (1959).

[5] Goldberg, A.V. and Silverstein, C.: Implementations of Dijkstra's Algorithm Based on Multi-Level Buckets, Technical Report 95-187, NEC Research Institute, Inc. (1995).

[6] Soon-chun, Park, et.al: Path Planning for Tottori City Bus Network using Two-stage Shortest Path Algorithm, Report of the Faculty of Engineering Tottori University, Vol.36,No.1,pp1-7 (2005)