INTERACTIVE TIMETABLING SYSTEM WITH FUNCTION OF ESTIMATING TRAIN CONGESTION ON TIMETABLE PLANNED

DAISUKE TATSUI[1], TAKETOSHI KUNIMATSU[1] & TAKASHI SAKAGUCHI[2]
[1]Railway Technical Research Institute, Japan.
[2]West Japan Railway, Japan.

ABSTRACT

It is important for railway operators to make suitable timetables. On the assumption that passengers' train paths are invariant even if the timetable changes, the timetables are determined according to the various statistical data under the current timetable. In planning the timetable, it is difficult to grasp the change of passengers' paths due to timetable modifications because it is too complex. In this paper, we propose a framework of timetabling with due consideration of the path change and propose a practical timetabling system which can quickly estimate the congestion of each train by using passengers' origin–destination (OD) data collected by automatic ticket checkers. With the system, timetable planners can interactively make and modify timetables by trial and error while confirming the congestion, and finally, they can reach the most preferable one. In order to realize such an interactive system, it is important to develop a fast estimation algorithm of train congestion. We developed a new shortest path search algorithm to determine which trains each passenger gets on. The algorithm devised based on the Dijkstra method has two features. First, the shortest path search from each node to all the destination stations in the composed graph network is executed only once. Second, overlapping the path searches are omitted using stored information of the shortest paths which have already been searched. By applying this algorithm, it took only about 10 s to estimate the train congestion under a timetable of the whole day on an urban commuter line.

Keywords: Dijkstra method, estimation of congestion, interactive system, passengers' path, shortest path search, timetabling.

1 INTRODUCTION

For railway companies, it is necessary to prepare timetables so that they provide enough transportation capacity to satisfy passengers' demands. Currently, a software dedicated for timetabling is used, but timetabling itself is not automated. Therefore, the quality of timetables relies on the timetable planners' experience and skill. The planners conduct lots of investigations under current timetables and determine a modified timetable based on the results, their experiences and so on. But there are two problems with respect to this method. One is that timetables they prepare are not reflecting passengers' real demands. The other problem is that the timetable planners assume that passengers' train paths will not change in case the timetable changes.

Especially during the rush time when many people use the railway for commuting, there are many people who give up boarding on the heavily congested trains and get on other trains

This paper is part of the proceedings of the 15th International Conference on Railway Engineering Design and Operation (COMPRAIL)
www.witconferences.com

ISSN: 2058-8305 (paper format), ISSN: 2058-8313 (online), http://www.witpress.com/journals
DOI: 10.2495/TDI-V1-N3-382-391

which have space for them to get on. It means that the passengers choose trains against their wills. Therefore, it is difficult to grasp all the passengers' real demands by the investigations under current timetable.

Also, passengers choose trains based on the given timetable in accordance with the purposes of trips. If they need to reach the destination station as soon as possible, they choose the earliest trains regardless of train congestion. If the timetable changes, train paths of these passengers may change because the earliest train will change according to the train reordering. But the changes of passengers' train path are so complex that the timetable planner has difficulty in grasping them by the investigation under current timetable.

So, as shown in the article, we have built a new timetabling system which can estimate train congestion by using passengers' OD data collected by the automatic ticket checkers. In order to reflect passengers' real demands, the system supposes that all the passengers get on the trains with the aim to reach the destination station as soon as possible. The system calculates passengers' train path under planned timetables, based on passengers' real demand.

The system has three features as follows. First, the timetable planners can estimate congestion of each train at sight by fast calculation algorithm of train congestion. Second, they can grasp congestion of each train visually and intuitively by the coloured train lines on train diagram. The colour is determined based on the estimated train congestion. Third, on the diagram which is coloured according to train congestion, they can modify timetable properly by using mouse operation such as dragging, dropping, copying and pasting.

At the same time, in this article, we propose a workflow which helps the timetable planners make the timetables more efficiently. In the workflow, the timetable planners can prepare an appropriate timetable interactively by repeating modification of the timetable and estimation of the train congestion.

2 PROBLEMS RELATED WITH PREPARATION OF TIMETABLE

2.1 Workflow of timetabling and problems associated with timetabling

Timetables are determined based on a comprehensive view of management policy, passengers' demands on the lines, passengers' volume under current timetable, and the constraints of resources required for operating trains such as the number of the rolling stocks and crews.

For timetabling, the timetable planners use the dedicated software, which enables them to conduct their task very efficiently. When they input departure time at the starting station, train line is drawn automatically by calculating the arrival time and departure time at intermediate stations. Moreover, it has a function that allows them to make cyclic pattern timetables by copying the existing train lines. However, it is impossible to propose appropriate number of trains, the number of vehicles of each train or the operating sections of each train so as to meet passengers' demands.

In order to help the timetable planners make decisions, many investigations about transportation situation are conducted. In these investigations, investigators on platforms count and record the number of passengers on each train. Also, in order to grasp the transportation situation, train operating companies have begun to estimate the number of passengers on each car based on its weight and passengers' trip data stored at automatic ticket gate (Kubota et al. [1]).

However, even if the investigation results clarify the problems of the current timetables such as existence of the heavily congested trains, it will not be obvious whether the modified timetable becomes more convenient for passengers, because it is difficult to grasp real passengers' demand from the results of executed investigations under current timetable, and it is also difficult to grasp all the passengers' train choice change caused by timetable modification. For example, suppose that 4 rapid trains and 8 local trains are operated every hour. Timetable planners cannot grasp train congestion change in case where the number of rapid train changes from 4 to 6, and that of local train changes from 8 to 6, or in the case where the places where the rapid trains overtake the local trains change. If the congestion of rapid trains drops due to increase in the number of rapid trains, there is possibility that passengers getting on a local train because of heavy congestion under current timetable may shift to a rapid train.

2.2 Related research

Many researches on estimating train congestion under planned timetable have been executed, making use of passengers' OD data obtained from the automatic ticket gate. For example, we built a prototype 'Multi-Agent Train Simulator with Intelligent Information Exchange' which has a function by which passengers' flow and delay time of each train are estimated simultaneously by using the data and timetable data (Kunimatsu et al. [2]). Also, Nagasaki et al. [3] proposed a fast-estimating algorithm of passengers' flow under a given timetable by preliminary search in a reverse direction (a direction going back in time).

However, in these researches, the timetable planners have to make a timetable for estimating the train congestion every time. So it is beyond the scope of their researches to consider their actual work flow with which they prepare timetables by trial and error based on the estimated train congestion in the stage of planning and making timetables. They conduct various modifications to the current timetable one by one and finally reach target timetable. Especially, when they prepare dozens of candidate timetables because of large-scale modification of current timetable, it is not realistic to prepare relevant data and estimate train congestion for all the candidate timetables. Therefore, it is necessary to unite the function of preparing timetable with that of estimating train congestion so that they can make lots of timetables and examine which one is the best. Then, the timetable planners can estimate train congestion and grasp them promptly every time they modify the timetable.

On the other hand, as for research on visualization, Inagawa et al. [4] proposed a coloured diagram in order to grasp the result of evaluation of timetables easily. In the research, at first, actual result operation data of some selected days are collected from the programmed train control system, and statistics of delay of departure and arrival time at each station are obtained. Then, train lines on the diagram are coloured according to the delay results. This method enables the timetable planners to grasp and specify easily where delays occur, but it is impossible to grasp the change of train congestion when the timetable changes because the method is based on the actual result.

2.3 Requirements for timetabling system

Based on the discussion above, in this research, we propose an interactive timetabling system which fulfils three requirements as follows. First, the system can estimate train congestion quickly under modified timetable. Second, the timetable planners can grasp promptly train congestion easily and visually. Third, the timetable planners can modify the timetable instantly with confirmation of the estimated train congestion.

3 INTERACTIVE TIMETABLING SYSTEM WITH FUNCTION OF ESTIMATING TRAIN CONGESTION

3.1 Proposed work flow of preparing timetables

Considering the requirements mentioned above, we propose a work flow of preparing timetables with the use of the system we built, as shown in Fig. 1.

First, the timetable planner inputs the initial draft timetable data into the system (e.g. the current timetable data). After that, estimation of train congestion is conducted under the input timetable according to user command. The calculation is performed based on the input timetable data, passengers' demands of the target line (origin stations, destination stations, time zones when passengers appear at origin stations, the number of passengers. These data are created by converting those obtained by automatic ticket gate.), and infrastructure data (order of stations, track layout, platform layout at each station). Then, the result is shown as a coloured diagram in which each train is coloured according to the train congestion so that the timetable planners can grasp train congestion easily (see Fig. 2). Therefore, in case the timetable planners can find some congested trains, they modify the timetable on the coloured diagram.

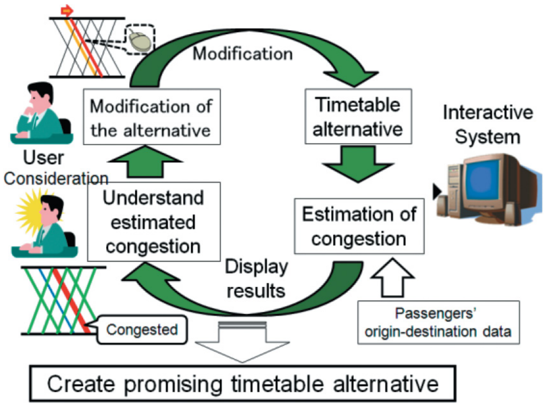
Modification
Modification of the alternative
Timetable alternative
Interactive System
User
Consideration
Understand estimated congestion
Estimation of congestion
Display results
Congested
Passengers' origin-destination data
Create promising timetable alternative

Figure 1: Process of interactive timetabling incorporating estimation of train congestion.

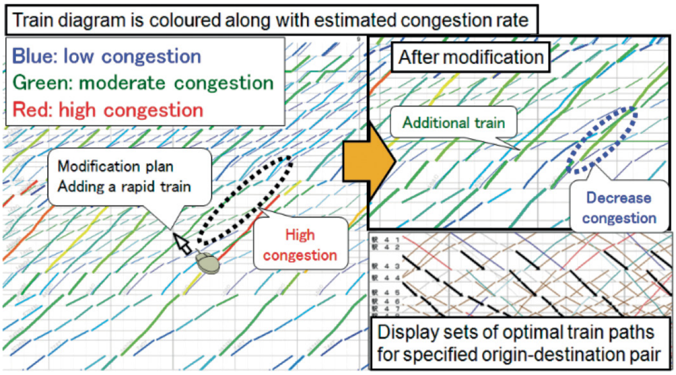
Train diagram is coloured along with estimated congestion rate
Blue: low congestion
Green: moderate congestion
Red: high congestion
Modification plan
Adding a rapid train
High congestion
After modification
Additional train
Decrease congestion
Display sets of optimal train paths for specified origin-destination pair

Figure 2: An example of system interfaces for timetabling.

After modification of the timetable, if the timetable planner executes estimation again, then the result of re-estimated train congestion under the modified timetable is shown as the coloured diagram. As a next step, the timetable planners compare train congestion before and after the modification of the timetable, and they can confirm whether the modification is appropriate. In this way, by repeating modification of the timetables and confirmation of the train congestion under the modified timetable, the timetable planners can find proper modification with trial and error by which the train congestion can be adjusted. Then, they can finally prepare the most suitable timetable.

If this work flow is executed, it is possible both to improve timetable quality based on the estimation of the train congestion and to make timetabling more efficient. In order to achieve this, the interactive system we built needs to have fast algorithm for estimating train congestion.

3.2 Presupposition of this research

Considering the purpose of the system (i.e. prior evaluation of the revised timetable), estimation of train congestion is conducted on the following three presuppositions.

First, passengers' demand is invariant even if a timetable changes. Passengers' demand will change if a new line is constructed, the mutual direct operation starts or the required time is shortened by vehicle speed-up. However, such a large-scale modification is not conducted frequently, so it is out of consideration in this research.

Second, delay caused by troubles or getting on or off by passengers at each station is not considered. It is difficult to consider train delay caused by getting on or off trains by passengers at each station, because it is necessary to analyse relationship between train congestion and dwelling time which varies by station or time. Therefore, in this research, it is out of consideration, and the timetable planners are supposed to set enough dwelling time so as not to cause delay.

Third, passengers choose trains so as to reach their destination station as soon as possible. It is based on the thought of timetabling that the planners have to grasp passengers' needs for rapid train which they really want to get on and to make timetables to have enough transportation capacity. For example, as shown in Fig. 3, if the ratio of the number of the rapid trains to that of the local trains is 1 to 1, train congestion of the local trains is found to be 120% and that of the rapid trains is found to be 150%. But the train congestion is calculated assuming the inclusion of the passengers who avoid getting on rapid train because of heavy congestion, which amounts to 30% of the total amount of the passengers. The problem is that these passengers are invisible to the investigators. Therefore, even if train congestion of the rapid trains is higher than those of the local trains, there is possibility that passengers' demand for rapid trains is not so huge as to operate rapid trains and local trains with the ratio of 2 to 1. Under the same situation, if passengers can get on any train they like without avoiding rapid trains of heavy congestion, train congestion of the rapid trains and the local trains is estimated to be 180% and 90%, respectively. Accordingly, it is better to operate rapid trains and local trains with the ratio of 2 to 1 in order to meet passengers' actual demands. To obtain an idea to conduct such a modification, it is necessary to modify the timetables according to passengers' actual demands for each train without considering passengers who avoid heavily congested trains.

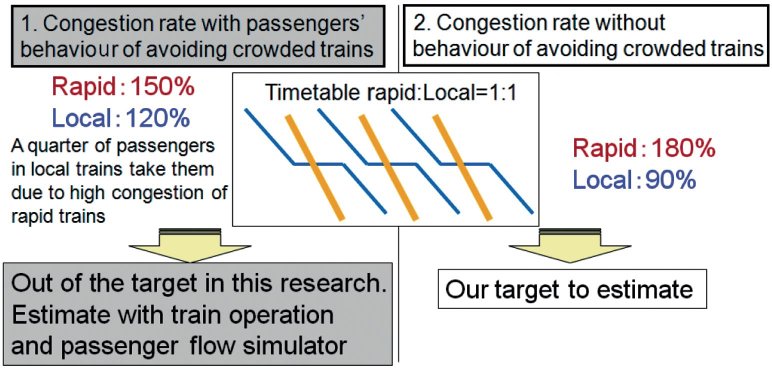
1. Congestion rate with passengers' behaviour of avoiding crowded trains
2. Congestion rate without behaviour of avoiding crowded trains
Rapid : 150%
Local : 120%
A quarter of passengers in local trains take them due to high congestion of rapid trains
Timetable rapid:Local=1:1
Rapid : 180%
Local : 90%
Out of the target in this research. Estimate with train operation and passenger flow simulator
Our target to estimate

Figure 3: Estimation reflecting passengers' real demands.

4 ALGORITHM FOR FAST ESTIMATION OF TRAIN CONGESTION

4.1 Method of estimating train congestion

In this system, estimation of train congestion is conducted by the following steps.

First, passengers' appearances at each station are created based on passengers' demands data. When passengers' data based on leaving time from the destination stations are available, they have to be converted to the format of passengers' demands data described in Section 3.2 according to the required time from the origin stations to the destination stations. Based on the data and Poisson distribution, appearance time of each passenger is determined by stochastic method. For further details, see Kunimatsu et al. [2].

Second, the graph network which represents the timetable is constructed. The graph network has nodes (including information of reference time and platform) and arcs as follows.

- Nodes

There are three types of nodes. The first is a train arrival node that represents each train arrival at each station. Here, the reference time is the arrival time, and the reference platform is where the train arrives and departs. The second is a train departure node that represents each train departure at each station. Here, the reference time is the departure time, and the reference platform is where the train arrives and departs. The last is a stay node that represents passengers stay on the platform in order to wait for the next train. Each departure node has one stay node. Also, the reference time is equal to the departure time plus Δ which means minute time smaller than the minimal time of timetable (e.g. 1 s). The reference platform is the same as that for the departure node.

- Arcs

There are three kinds of arc, and the cost of each kind of arc is equal to the difference between the reference times of the nodes regardless of their kinds. The first one is a train arc that represents a train running between stations or dwelling at stations. The second one is a transfer arc that represents passengers' change of train. A tail node of transfer arc is an arrival node, and a head node of it is a corresponding stay node which has the nearest reference time immediately after the arrival node, considering the transfer time. Here, the reference platform of the stay node is different from that of the arrival node, but the station of the stay node is the same as that of the arrival node. The last one is a stay arc that represents passengers' waiting for next train on the same platform. The tail node of the stay arc is the stay node, and the head of it is a corresponding stay node which has the nearest reference time immediately after the arrival node. The platform of the stay node is the same as that of the arrival node.

After creating a graph described above, the shortest path search from every arrival or departure node to every station is conducted according to the method based on Dijkstra method. In order to accelerate search speed of this search algorithm, the following are devised. First, each node has data about both the next node on the shortest path and the shortest required time in order to reach every station. Second, according to the arrival or the departure time of each node, the shortest path search is conducted in a reverse direction (a direction going back in time) in the same way as Nagasaki et al. [3] proposed. Third, both the shortest required time and an arc which should be traced on the shortest path from the node are recorded or updated as the data of the shortest path, according to the sum of the time data of another node linked by the arc and the cost of the arc. Moreover, using the stored shortest path data, the calculation of re-search is accelerated by omitting overlapping path search (see Section 4.2).

Based on the result of the shortest path search, the number of passengers who appear at the origin station at the time zone between nodes is assigned to each node, classifying passengers according to their destination stations.

Finally, the number of passengers is assigned to each train and section according to both the number of passengers and the shortest path data. By accumulating them, the total number of passengers of each train in each section is calculated.

4.2 Algorithm for fast re-estimation

In order to speed up the re-estimating train congestion after the first estimation, the path search which has no possibility that train paths will change by timetable modification is omitted.

In this research, we propose a method by which re-search of the shortest path is conducted only for the node which may have changed the shortest path according to updated information of the structure change of the graph network. The following process is performed every time the path search is conducted and the timetable is modified.

After the path search described in Section 4.1 is conducted, the data of the shortest path at each node are retained. The data include the next node on the shortest path and the shortest required time in order to reach every station.

On the other hand, according to timetable modification, the structure of the graph network is updated, and changing the data of the shortest path of each node is recorded with the method described in Section 4.2.1. By means of this method, all the changes of the graph structure after the last path search are stored.

When research is carried out after timetable modification, data of the shortest path are updated with the method described in Section 4.2.2 based on the stored data described above.

4.2.1 Timetable modification and updated data of the network configuration

In each operation of modification of a timetable, data of network structure are updated, and list of nodes necessary for checking is updated by the following method. As for the operations which are not described in this section, they can be substituted by cancelling and adding train to the parts in need of correction.

Let N_o be a set of network nodes before modification; A_0, a set of arcs before modification; a_s, the tail of arc a; a_d, for the head of arc a; a_c, cost of arc before modification.

When the user cancels trains (including shortening of operating section) and changes the train operation from stopping to passing, all the nodes and arcs related to the train, the station and the section are deleted. Let N_- be a set of deleted nodes, and A_- be a set of deleted arcs.

In the case where the user adds trains (including extension of operating section) and changes the train operation from passing to stopping, all the nodes and arcs related to the train, the station and the section are added. Let N_+ be a set of added nodes.

If timetable is modified according to N' (a set of node which satisfies $N' = N_0 - N_- + N_+$), the graph structure after the modification is created by the method described in Section 4.1. Let A' be a set of arcs of the graph corresponding to the modified timetable.

As for recording or storing in the list of nodes necessary for checking, four kinds of the nodes are all recorded and stored in the list as described in Fig. 4. They are added nodes (N_+), data of the tail nodes of the added arcs which belong to not A_0 but A', data of the tail nodes of the deleted arcs which belong to not A' but A_0 in the case where the nodes are not deleted or belong to N', and data of the tail nodes of the arcs whose cost changed in the case where the arcs belong to both A_0 and A' and the a_c is different from a_c'. The data recorded and stored are a set of [node n, destination station s], and the recorded data are the one about every destination station from the nodes.

4.2.2 Updating the shortest path data using network updated data

For the graph network of the modified timetable, determining whether cost of the shortest path has changed and updating the data of the shortest path are conducted with the use of the list necessary for checking by the following steps until the list becomes empty.

First, the node which has the last reference time in the list is considered as a target. Then, the node and the destination stations are deleted from the list. Here, the order of destination stations is not considered.

In the case where there is neither the target node nor the data of the shortest path (the next node on the path, the shortest required time), the path search for the destination stations and data updating are conducted by the method described in Section 4.1. In addition, all tail nodes of all the arcs whose head are the target node and the destination station are added to the list; then the first step is conducted again.

If the node u and the destination station s have already had data of the shortest path, confirming whether it is optimal is conducted with the following method. Here, let C_{xy} be the cost from the target node x to the node y; C'_{us}, the cost recorded in the shortest path data; and v_0, the next node in it.

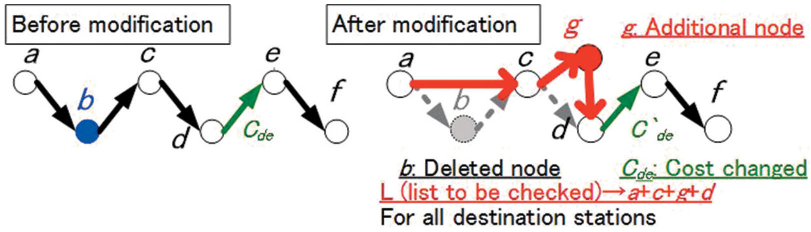

Figure 4: Process of specifying nodes to be checked.

1. Minimum cost C^*_{us} and v^* which minimizes the cost are obtained by calculating $\mathrm{C}_{\mathrm{uv}} + C_{\mathrm{vs}}$ for every arc.
2. If $v_0 = v^*$ and $C_{\mathrm{uv0}} + C_{\mathrm{v0s}} = C^*_{\mathrm{us}} = C'_{\mathrm{us}}$, then the information is not changed (Fig. 5 (case 1)).
3. If v_0 is not in the changed graph and $C'_{\mathrm{us}} = C^*_{\mathrm{us}}$, or $v_0 \neq v^*$ and $C_{\mathrm{uv0}} + C_{\mathrm{v0s}} > C'_{\mathrm{us}} = C^*_{\mathrm{us}}$, then the next node is changed from v_0 to v^* because the minimum cost is the same but the next node is changed (Fig. 5 (case 2)).
4. If $v_0 = v^*$ and $C_{\mathrm{uv0}} + C_{\mathrm{vos}} = C^*_{\mathrm{us}} \neq C'_{\mathrm{us}}$, then the minimum cost is changed from C'_{us} to C^*_{us} because the path is the same but the cost is changed. In addition, each tail node of each arc whose head is the node u is added to the list necessary for checking with respect to the destination station s (Fig. 5 (case 3)).
5. If v_0 is not in the changed graph and $C'_{\mathrm{us}} \neq C^*_{\mathrm{us}}$, or $v_0 \neq v^*$ and $C'_{\mathrm{us}} \neq C^*_{\mathrm{us}}$, then the next node is changed from v_0 to v^*, and the minimum cost is changed from C'_{us} to C^*_{us}. In addition, the each tail node of each arc of which the head is node u is added to the list necessary for checking with respect to the destination station s (Fig. 5 (case 4)).
6. The first step of this section is conducted again. Target node is determined.

In this way, both confirmation of the possibility that the shortest path from each nodes is changed by timetable modification and path change operation in case where the shortest path is changed are conducted at the same time by propagation on the graph. This enables us to narrow the re-search width without giving any influence on the optimality of the shortest path. For example, if local change of the network structure such as reordering of one train is only considered, it is possible to shorten the calculation time drastically.

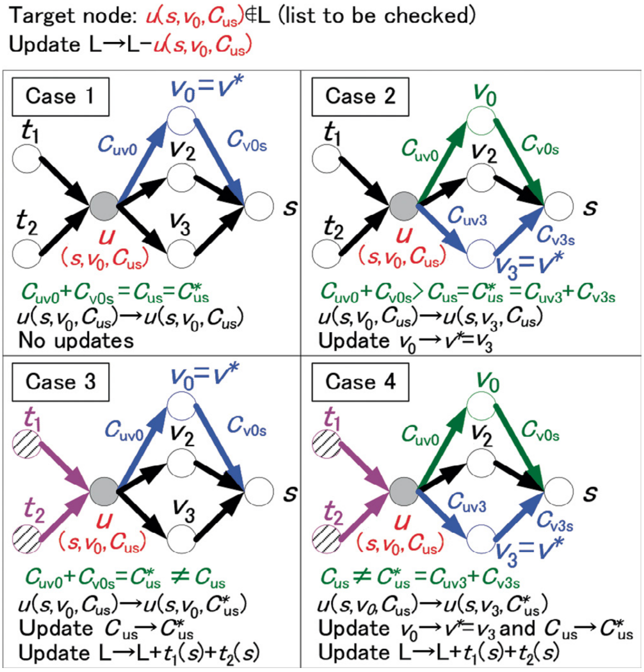
Target node: $u(s,v_0,C_{us}) \notin$ L (list to be checked)
Update L→L−$u(s,v_0,C_{us})$
Case 1
$v_0=v^*$
t_1
C_{uv0}
v_2
C_{v0s}
t_2
v_3
s
u
(s,v_0,C_{us})
$C_{uv0}+C_{v0s}=C_{us}=C^*_{us}$
$u(s,v_0,C_{us}) \rightarrow u(s,v_0,C_{us})$
No updates
Case 2
v_0
t_1
C_{uv0}
v_2
C_{v0s}
t_2
C_{uv3}
s
u
C_{v3s}
(s,v_0,C_{us})
$v_3=v^*$
$C_{uv0}+C_{v0s}>C_{us}=C^*_{us}=C_{uv3}+C_{v3s}$
$u(s,v_0,C_{us}) \rightarrow u(s,v_3,C_{us})$
Update $v_0 \rightarrow v^*=v_3$
Case 3
$v_0=v^*$
t_1
C_{uv0}
v_2
C_{v0s}
t_2
v_3
s
u
(s,v_0,C_{us})
$C_{uv0}+C_{v0s}=C^*_{us} \neq C_{us}$
$u(s,v_0,C_{us}) \rightarrow u(s,v_0,C^*_{us})$
Update $C_{us} \rightarrow C^*_{us}$
Update L→L+$t_1(s)$+$t_2(s)$
Case 4
v_0
t_1
C_{uv0}
v_2
C_{v0s}
t_2
C_{uv3}
s
u
C_{v3s}
(s,v_0,C_{us})
$v_3=v^*$
$C_{us} \neq C^*_{us}=C_{uv3}+C_{v3s}$
$u(s,v_0,C_{us}) \rightarrow u(s,v_3,C^*_{us})$
Update $v_0 \rightarrow v^*=v_3$ and $C_{us} \rightarrow C^*_{us}$
Update L→L+$t_1(s)$+$t_2(s)$

Figure 5: Process of updating the optimal path for the destination.

5 EVALUATION OF A PROTOTYPE BUILT FOR AN ACTUAL COMMUTER LINE

In order to evaluate the system, we built a prototype for an urban commuter line and measured estimating time for the current timetable and re-estimating time when the train order between one rapid train and a following train is changed at 20 stations.

The target line includes 103 stations, the number of passengers per day is 1,600,000 and the number of trains per day is 2,319. The measurement of the estimation time was executed on a PC equipped with a 2.4 GHz Pentium Core i7 processor, 8 GB memory and Windows 7 64 bit.

As a result, it took approximately 10 s for the estimation of train congestion under current timetable, and 7 s for the estimation under modified timetable. If the algorithm which was proposed by Kunimatsu et al. [2] is applied, the estimating time is approximately 180 s. So the estimating time of the algorithm we propose is considered sufficiently practical.

Moreover, comparing the estimating time under the current timetable and with that under the modified timetable, we confirmed the effect of the proposed fast algorithm for re-estimating train congestion in the case where the shortest path search for the nodes is not conducted if there is no possibility that the path of the node does not change even in case of modification of timetable.

However, it is early to conclude that the calculation speed is fast enough to use the system as an interactive system. As a future work, it is necessary to realize faster calculation (e.g. algorithm for summing up the passengers).

6 CONCLUSION

In this research, we built the prototype interactive timetabling system in order to realize raising both the quality of timetabling work and effectiveness of it. Specifically, we designed user-friendly interface and fast algorithm of estimating train congestion. As for fast-estimating algorithm of train congestion, it took 10 s in order to estimate current timetable on the actual train line, and it took only 7 s in order to estimate train congestion under a modified timetable when the train order between one rapid train and a following train is changed at 20 stations. As a result, effectiveness of fast re-estimation algorithm of train congestion is confirmed.

As for the future work, it is necessary to consider faster algorithm and to verify the usability of the system in case of applying it to various complex train lines.

REFERENCES

[1] Kubota, T., Morita, Y., Ota, M., Furuya, S. & Ieda, H., Analysis of rate of train overcrowding by stored data at ticket gates and load weight units, Infrastructure planning view. *Japan Society of Civil Engineers*, **25**, pp. 641–646, 2008. (in Japanese)

[2] Kunimatsu, T., Hirai, C. & Tomii, N., Evaluation of timetables by estimating passengers' personal disutility using micro-simulation. *Proceedings of the 3th International Seminar on Railway Operations Modelling and Analysis* (RailZurich2009), 2009.

[3] Nagasaki, Y., Asuka, M. & Komaya, K., A fast method for estimating railway passenger flow. *Computers in Railways* **X**, pp. 179–187, 2006. DOI: 10.2495/CR060181.

[4] Inagawa, M., Tomii, N. & Ushida, K., Visualization of train operation record data. *J-Rail*, **16**, pp. 745–748, 2009. (in Japanese)