Crew and Vehicle Rescheduling Based on a Network Flow Model and Its Application to a Railway Train Operation

Tatsuhiro Sato * Shuichiro Sakikawa* Toyohisa Morita* Nobutaka Ueki*
Tomohiro Murata†

Abstract —**For the crew rescheduling problem (CRP) and the vehicle rescheduling problem (VRP), we propose a 0-1 integer programming formulation based on a network flow model and a solution method that uses a heuristic flow modification and a local search technique. The proposed formulation is able to represent the "differences between the new and original schedules", which is a significant criterion for the CRP/VRP, though taking these differences into account is difficult for other related formulations based on the set partitioning/covering models. Implementation of a prototype system for vehicle rescheduling showed that the model is also effective for designing interactive operations between users and the computer system. The results of numerical experiments with real-world vehicle rescheduling data showed that the proposed method generated feasible solutions within a practical amount of time, and on the basis of a two-phase solution approach, the proposed method improved the evaluation values of the solution. This work could lead to practical computer systems which would effectively support train recovery operations under strict time limitations.**

Keywords: ***crew/vehicle rescheduling, network flow model, heuristics, local search, railway, train operation***

1 Introduction

The *crew rescheduling problem* (CRP) and the *vehicle rescheduling problem* (VRP) involve assigning crews (drivers and conductors) or vehicles to trains while they are running whenever disruption occurs and timetables are changed.

For instance, most Japanese railway lines are known for congested train networks. In addition, they form a huge and complicated traffic network because of their interconnectedness. Therefore, once some minor disruption occurs to one train schedule, it spreads over such a wide area of the network that other trains are delayed and thus significantly lowers passenger transportation efficiency. To prevent delays from escalating, train operators, who are typically veteran experts, try to immediately change the timetable. They make time alterations, change departure orders, cancel train services, and set up extra trains. To ensure the availability of these changes, they also have to coordinate the schedules for vehicles and crews according to the changed timetable under strict time limitations. It is increasingly difficult to secure skilled operators because of the complicatedness of the task and a shortage of human resources.

The CRP/VRP can be viewed as special cases of the *crew scheduling problem* (CSP) and the *vehicle scheduling problem* (VSP) that must be solved under dynamic circumstances. These problems have been widely investigated[1]-[6]. For example, Cacchiani et al.[3] have proposed an integer linear programming (ILP) formulation for VSP with seat constraints, which means deciding on the combination of vehicles required for each train to satisfy passenger seating, and developed a heuristic solution method. Caprara et al.[4] have proposed an approximation method for solving the CSP by modeling it as a set covering problem (SCP) and using an enumeration algorithm with Lagrangian relaxation. Fischetti et al.[5] have considered a simplified but still NP-hard case in which several depots are specified, and they proposed a 0-1 linear programming formulation that can be applied to both crew and vehicle scheduling. They devised an exact method based on a polyhedral approach.

Almost all the studies for the CSP/VSP have focused on making a schedule from *scratch*, with the assumption that there is enough time for scheduling because the schedule will not be used right away but in the future. However, these approaches appear to be inappropriate for the CRP/VRP because of the strict time limitations.

In this study, we developed a 0-1 integer programming formulation for the CRP/VRP on the basis of a network flow model. and a solution method with the concept of two-phase modification of temporary solutions using a

*Hitachi, Ltd. 292, Yoshida-cho, Totsuka-ku, Yokohama-shi, 244-0817, Japan Email:tatsuhiro.sato.uq@hitachi.com
†Graduate School of Information, Production and Systems, Waseda University 2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, 808-0135, Japan

(Advanced on line publication:1 August 2009)

heuristic flow modification and a local search technique. Implementation of a prototype system and computational results of real-world data from a Japanese railway line for vehicle rescheduling are described. They indicate that our network-oriented modeling and solution approach is suitable for handling several significant requirements for developing a computer system for the CRP/VRP, such as processing time, differences between the new and original schedules, and interactive operations between users and the system.

2 Problem Description and Model

2.1 Multi-Commodity Flow Network

A timetable for one train can be partitioned into several trips. Each trip starts and ends at stations at which crews can be changed or transferred. A crew schedule can be represented as a sequence of several trips, which is called *pairing*. This is similar to the one described above for a vehicle. Trips are segments of a timetable partitioned by stations where entering a depot or turning back is possible. A sequence of several trips represents a schedule for a vehicle, which is also called *pairing* in this paper.

Hereafter, the word *resource* refers to a vehicle or crew. By associating a node with each trip, and a directed link with each possible trip transition, we can represent schedules as flows of the resources on a network with the vertices and the links described above (Figure 1). In Figure 1, a solid arrow represents a resource flow, namely, the allocation of resources to both the nodes from which the arrow departs and arrives, in this order. Therefore, a series of solid arrows is a schedule, or pairing, for one resource.

When disruption occurs and the timetable is thus changed, the network should also be changed. This may cause problems with schedule feasibility. Suppose there are two trips, **a** and **b**, **a** is going to end before **b** according to the schedule, then there is a link from the node of **a** to the node of **b**, and there is also a flow of a resource on the link. Under these circumstances, if **a**'s end time (i.e., the station arrival time) comes after **b**'s start time because of some delay or a change of **a**'s end time, the link from **a** to **b** vanishes because the sequence of **a** ending before **b** is no longer available. Consequently, the flow from **a** to **b** with the meaning of *executing* **a** *first and then* **b** is no longer physically available.

It is important in the CRP/VRP that we should not only make a feasible schedule, but also satisfy various criteria derived from union contracts and/or company regulations. Examples of the criteria for the vehicle are *duration of waiting at stations* and *number of vehicles with the final destination changed* and for the crew, *number of transfers*, *number of rides without driving or conducting*, *amount of overtime work*, and *of meal or break time*.

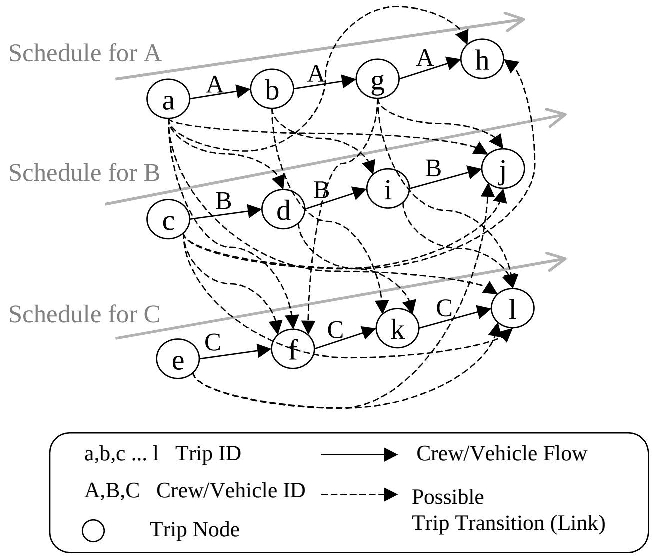

Figure 1: Multi-commodity Flow Network for Crew/Vehicle Schedules

In addition to the criteria derived from specific problem instances, there is also a criterion common to the CRP/VRP, which is the *differences between the new and original schedules*. If there are only a few differences, train operators are able to rapidly understand and confirm the new schedule. Additionally, a schedule with only a few differences is helpful for the field staff at train operation areas such as stations or depots, because the fewer differences there are, the more easily the staff can make necessary arrangements for the new schedule, such as notifications to crews or preparations for reserved vehicles, within strict time limitations. As static scheduling like the CSP/VSP is done from scratch with no time restrictions, minimizing differences between the new and original schedules is an inherent criterion for dynamic scheduling like the CRP/VRP.

2.2 Mathematical Formulation

2.2.1 Notation

V : Set of nodes $\{1, \cdots, n\}$. It consists of trips and dummy nodes that are described later.

E : Set of directed links $\{a_{ij} \equiv (i,j) \mid i,j \in V\}$. If ($i$'s end time) < ($j$'s start time) and ($i$'s end location) = ($j$'s start location), then the directed link (i,j) is an element of E.

R : Set of resources $\{1, \cdots, m\}$.

s, t : Dummy nodes denoting start (s) and end (t) of the schedule. All the directed links from s to the other nodes are elements of E. Similarly, all the directed links from nodes except t to t are elements of E. The links that end at s are not included in E. The links that start from t are also not included in E. For notational convenience, node $1(\in V)$ and node $n(\in V)$ represent s and t, respectively.

x_{ij}^k : 0-1 variable denoting the amount of resource r_k on the directed link $(i,j) \in E$. It is 1 when resource r_k flows from i to j, in other words r_k is scheduled to be allocated to i and j in this order.

$\boldsymbol{X^k}$: $n \times n$ matrix of which (i,j) element is a variable x_{ij}^k.

$$\boldsymbol{X^k} = \begin{pmatrix} x_{11}^k & x_{12}^k & \cdots & x_{1n}^k \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1}^k & x_{n2}^k & \cdots & x_{nn}^k \end{pmatrix}$$

But, if $(i,j) \notin E$, then (i,j) element of $\boldsymbol{X^k}$ is 0.

The schedule for resource k is represented as a sequence of nodes that can be constructed by tracing directed links with $x_{ij}^k = 1$ from nodes 1 to n.

$\boldsymbol{X}$: $n \times n$ matrix of summation of $\boldsymbol{X^k}$

$$\boldsymbol{X} \equiv \boldsymbol{X^1} + \boldsymbol{X^2} \cdots + \boldsymbol{X^m} = \sum_{k=1}^{m} \boldsymbol{X^k}$$

Each element in $\boldsymbol{X}$ is the total sum of flows on each directed link. That is, let x_{ij} be (i,j) element of $\boldsymbol{X}$, then $x_{ij} = \sum_{k=1}^{m} x_{ij}^k$ for the directed links in E, and $x_{ij} = 0$ for the others.

e_{ij}^k : flow feasibility $(k \in R,\ i,j \in V)$

$$e_{ij}^k = \begin{cases} 1 & \text{k is able to flow from i to j} \\ 0 & \text{otherwise} \end{cases}$$

It should be mentioned that if $(i,j) \notin E$, then e_{ij}^k is always set to 0.

Because E is constructed so as not to include links that are inconsistent with time and location, tracing the links results in physically valid allocations to the trips for any resource. However, since there are certain resources which cannot be allocated to certain trips for some reason, such as vehicle type, we define the feasibility for each resource by e_{ij}^k.

d_i : the number of resources required for node i. Each d_i is greater than or equal to 1, that is, $d_i \geq 1$.

Though d_i generally takes 1, there are some cases in which d_i is greater than 1, such as when a train needs multiple vehicles joined together to meet passenger demands, or multiple crews to handle a lot of duties.

b_{ij}^k : flow of resource k in the original schedule. If resource k was scheduled to be allocated to trip i and j in this order, then it takes 1; otherwise it takes 0.

Where both i and j are elements of V, but the directed link (i,j) is not always in E because the network is changed due to disruption and timetable changes.

c_{ij}^k : cost of a resource k's flow on link (i,j). The definition is further explained in 2.2.2.

$c(\boldsymbol{X^k})$: cost function of the schedule for resource k.

$c(\boldsymbol{X})$: cost function of the whole schedule.

$c(\boldsymbol{X^k})$ corresponds to a criterion that is independent of the other resources' schedules. For instance, it could stand for the suitability of meal time, that is, whether the schedule for crew k has meal time with appropriate timing and length from the point of view of maintaining reasonable working conditions.

On the other hand, if there is a criterion over several resources, it should be represented as a form of $c(\boldsymbol{X})$. An example of $c(\boldsymbol{X})$ is the *deviation of duration of waiting at stations.*

Both $c(\boldsymbol{X^k})$ and $c(\boldsymbol{X})$ are weighted sums of these standardized costs if there are multiple criteria to be considered.

2.2.2 0-1 Integer Programming Formulation

The CRP/VRP can be defined as the following 0-1 integer programming problem.

Minimize

$$w_1 c(\boldsymbol{X}) \ + \ w_2 \sum_{k=1}^{m} c(\boldsymbol{X^k}) \ + \ w_3 \sum_{k=1}^{m} \sum_{(i,j)\in E} c_{ij}^k x_{ij}^k \quad (1)$$

Subject to

$$\sum_{(1,j)\in E} x_{1j}^k \ = \ 1 \qquad \forall k \in R \quad (2)$$

$$\sum_{(i,n)\in E} x_{in}^k \ = \ 1 \qquad \forall k \in R \quad (3)$$

$$x_{ij}^k \ \leq \ e_{ij}^k \quad \forall (i,j) \in E,\ \forall k \in R \quad (4)$$

$$\sum_{(q,i)\in E} x_{qi}^k \ = \ \sum_{(i,q)\in E} x_{iq}^k \quad \forall i \in V \setminus \{1,n\},\ \forall k \in R \quad (5)$$

$$\sum_{k=1}^{m} \sum_{(q,i)\in E} x_{qi}^k \ = \ d_i \quad \forall i \in V \setminus \{1,n\} \quad (6)$$

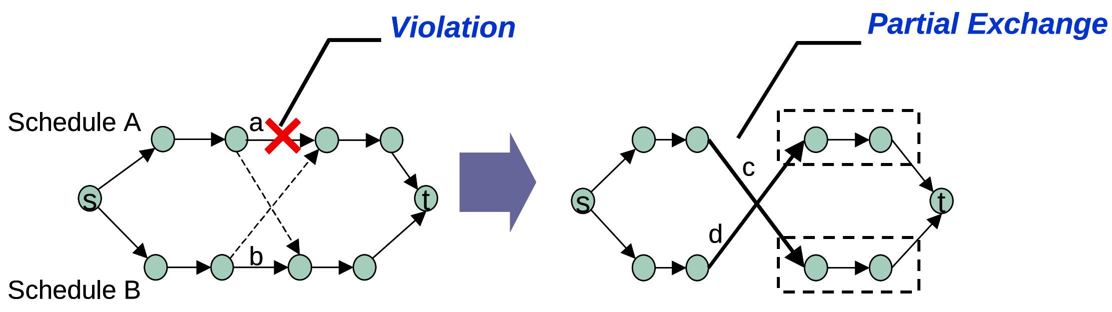

Figure 2: Partial Schedule Exchange

$$x_{ij}^k \in \{0,1\} \qquad \forall (i,j) \in E, \ \forall k \in R \tag{7}$$

In objective function (1), w_1, w_2 and w_3 are non-negative constants, and mean weights of the three different cost functions.

The meanings of the above constraints are as follows. Constraint (2): each schedule starts from node 1. Constraint (3): each schedule ends at node n. Constraint (4): each resource flows only on the directed links that are feasible for the resource. Constraint (5): flow conservation constraint except for the start and end nodes. Constraint (6): each node except for nodes 1 and n is covered by d resources, that is, d resources must be allocated to trip i. Constraint (7): the amount of flow of each resource on each link is 0 or 1.

Each instance of constraints (2)-(5) is related to only one resource. On the other hand, each instance of constraint (6) is related to multiple resources.

When crews are able to ride on trains in the crew scheduling without driving or conducting, (6) is replaced by the following inequality version, which allows an arbitrary number of resources more than d_i to be allocated to a trip.

$$\sum_{k=1}^{m} \sum_{(q,i)\in E} x_{qi}^k \ \geq \ d_i \quad \forall i \in V \setminus \{1,n\}, \ k \in R \tag{8}$$

In objective function (1), the term $\sum_{k=1}^{m} \sum_{(i,j)\in E} c_{ij}^k x_{ij}^k$, which is called *cost function (A)* here, is the sum of all the flow costs, that is, the sum of c_{ij}^ks with the link (i,j) on which some resource flows. The value of each c_{ij}^k is set according to one of these two definitions.

Definition (1)

$$c_{ij}^k = \begin{cases} 1 & if \ b_{ij}^k = 1 \\ 0 & otherwise \end{cases}$$

Definition (2)

$$c_{ij}^k = \begin{cases} 1 & if \ b_{ij}^r = 1, \ \ \exists r \in R \\ 0 & otherwise \end{cases}$$

According to definition (1), the cost function (A) returns the sum of different parts of each resource k's solution from its original schedule. As for definition (2), the cost function (A) represents the sum of different parts of each resource k's solution from the original schedule for all resources.

Figure 2 shows an example of modifying schedules for two resources, in that two partial schedules, both of which have two trips, are exchanged, so the number of changed trips is four. In this case, the value of the cost function (A) is six and two, according to definitions (1) and (2), respectively. The value in definition (1) represents the number of trips that are not included in the original schedule for a resource (the number plus 2, strictly), whereas the value in definition (2) means the number of switches on the way to the original schedule for the other resource. The major differences between these two definitions is that, for one exchange, the value in (1) depends on the number of trips involved in the partial schedules exchanged, while the value in (2) is always constant, that is, two.

For the train operator, who usually changes a resource's schedule by switching it to another schedule in the middle of the flow, it is not the number of changed trips, but the number of switches that is more significant. Also, the field staff needs to specify when and where the schedule will be switched to another schedule in order to make necessary arrangements before the train reaches the point where the schedule is switched. Therefore, it seems more effective from both the viewpoints of decision support for operators and work support for field staff, to use definition (2) as the cost c_{ij}^k of the differences between the new and original schedules.

2.3 Discussion of the Model

It is well known that the CSP/VSP, which are similar problems to the CRP/VRP, can be modeled as a kind of set partitioning problem (SPP) or set covering problem (SCP). The CSP or VSP is formulated as a problem of deciding an optimal pairing combination that involves all trips and satisfies several predefined conditions, where *pairing* is a sequence of trips that means a schedule for a resource. An example of the SCP formulation is as follows.

[Minimize]

$$\sum_{j \in N} c_j x_j \tag{9}$$

[Subject to]

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad \forall i \in M \tag{10}$$

$$x_j \in \{0, 1\} \tag{11}$$

where N is a set of pairings, j is an element of N, M is a set of trips, i is an element of M, and c_j is the cost of pairing j. Also, a_{ij} takes 1 if pairing j involves trip i and 0 otherwise, and x_i is a decision variable that takes 1 if pairing j is selected and 0 otherwise.

A variety of methods have been proposed to find optimal or near-optimal solutions for the set-partitioning-or-covering-modeled CSP/VSP. Most of them can be classified into one of two approaches. One approach is preparing a subset of pairings in advance in which each element is reasonable as a schedule for a resource and deciding an optimal or near-optimal combination of pairings within the subset. For instance, Caprara et al. [4] have proposed a method for solving the crew scheduling problem that consists of an enumeration algorithm for constructing the subset of reasonable pairings and a heuristic search method based on *Lagrangian relaxation*[7].

Another approach is more exact but even less suitable for larger instances, that is, searching an optimal or near-optimal pairing combination while generating additional pairings as needed. For instance, Haase et al.[6] have proposed an exact solution approach for solving a simultaneous vehicle and crew scheduling problem in urban mass transit systems. Their approach incorporates a column generation process[8] into a branch-and-bound scheme, a combination that is generally referred to as *branch and price*[9].

Table 1 shows a comparison between the SPP/SCP model for the related CSP/VSP and the proposed network flow model for the CRP/VSP.

There is a fundamental premise in the SPP/SCP model that a set of pairings is given from outside, which means, in terms of the implementation of computer scheduling systems, that we have to develop a process of generating the pairing set on the basis of a different framework outside the model. In other words, as the SPP/SCP model is unable to provide any guidelines or principles for generating adequate pairings, a quite different model is needed for it. Therefore, the proposed model, which has no preconditions about the given pairing set and thus can handle wider problems, seems to be more useful from the point of view of engineering the computer systems. For instance, we are able to naturally implement user interactive functions for operating the solution, as they can be regarded as operations of the network flow model. This is explained further in the next section.

The resources available for rescheduling in CRP/VRP are limited to those that are on hand at the moment, whereas in static scheduling from scratch, the number of resources available is not limited in theory. This is equivalent to the number of pairings selected as a solution in the SPP/SCP model. Suppose we regard each trip itself as a pairing and select all of the pairings as a solution. Then, the solution is one of the feasible solutions of the SPP/SCP model. Of course, such a solution is unpractical because a lot of resources are needed. However, there is no doubt that allowing a lot of resources in the SPP/SCP model makes it easier to generate a feasible solution. On the other hand, it is difficult to find a feasible solution with the proposed model because available resources are specified in advance as elements of the resource set R.

As resources are treated as *class* in the SPP/SCP model, the model is unable to identify individual resources. For example, in a case where q pairings are selected as a feasible solution of the SPP/SCP model, although the solution tells us there are q resources needed, it is unknown how each pairing in the solution is associated with each instance of the q resources. Therefore, the SPP/SCP model cannot represent the differences between the new and original schedules because it is impossible to identify the new and original schedules without associating them with individual resources.

The objective function of the proposed model is a form of the weighted sum of the three kinds of cost functions. Two of the three functions, which use $c(\boldsymbol{X})$ and $c(\boldsymbol{X^k})$, could become non-linear functions depending on the criteria treated. On the other hand, the objective function of the SPP/SCP model is linear, which means that it is relatively easy to apply *relaxation techniques*, such as a *continuous relaxation* and a *Lagrangian relaxation*. A solution of a relaxed problem serves as a lower bound of the optimal solution of the original problem. The SPP/SCP model has an advantage in that there are several standard approaches for designing the solution method based on the lower bound information. One is a branch-and-bound method with pruning by the lower bound and another is a heuristics approach that proceeds towards the direction for an estimated optimum value using the gap between the value of the temporal solution and the lower bound.

In the objective function of the proposed model, the most significant criterion for rescheduling, namely, the *differ-*

Table 1: Comparison of Two Models

	Set partitioning/covering model for the CSP/VSP	Network flow model for the CRP/VRP
Pairings	given	not given
Resources	not specified	specified
Differences between new and original	unable to treat	able to treat
Objective function	linear only	both non-linear and linear
Size	medium (depends on the pairing set)	large

ences between the new and original schedules, is separated as a linear function. Thus, if only considering this criterion, that is, setting the weights w_1 and w_2 in the model to 0, then the standards for designing the solution method based on the relaxation techniques also could be applied to the proposed model.

If an objective function has non-linear terms, it is practical for the design of the solution method to resort the native heuristics for the problem and/or the *generate and test method*, in which the evaluation and update of temporal solutions are repeated. The proposed solution method described in the next section is one instance based on these approaches. Moreover, the objective function of the SPP/SCP has a limitation of handling only the criterion associated with each pairing, which corresponds to the cost function $c(\mathbf{X^k})$ in the proposed model. If the cost function $c(\boldsymbol{X})$, meaning a criterion that has to be evaluated over multiple pairings at a time, is needed, the SPP/SCP model would also have the non-linear term in the objective function, which results in using the above mentioned solution approaches for finding the solution.

Even if it is a small problem to be solved, the size of the proposed model becomes relatively larger due to the large number of flow variables to all combinations of the directed links in E and the resources. On the other hand, as the decision variable x_j is given to each pairing, the size of the SPP/SCP model depends on the pairing subset. Considering both the large model size and the strict time limitations, it is essential to design an effective solution method for the proposed model, which finds an acceptable solution within a practical amount of time.

3 Solution Method

Because the CRP/VRP are dynamic problems that are expected to be solved iteratively during train operation, the solution method should be highly responsive. Keeping this in mind, we propose the following two-phase solution approach (Figure 3). Phase 1 generates a feasible solution by modifying the original schedule, and then Phase 2 continues to search for alternatives that improve the evaluation value while maintaining schedule feasibility until the time limit expires. With this approach, it becomes possible to reschedule adapting to the time remaining for train recovery, which depends on the situation.

Phase 1: Partial Exchange Heuristics

We define a heuristic method for Phase 1, assuming that some instances of constraint (4) in the model have been violated because of the changes of the network. The violation of constraint (4) means that some flows have become infeasible because of transport disruption and timetable changes.

In such a situation, train operators generally try to correct the violations by exchanging a part of the schedule with another one. An example of this partial schedule exchange is shown in Figure 2. We use the partial exchange as a heuristic flow modification to make a feasible schedule in Phase 1. Phase 1 consists of the following steps.

Step 1 select the earliest violated flow (**a**).

Step 2 select a flow (**b**) from any other schedule that does not include **a**.

Step 3 exchange the partial schedule after **a** with the partial schedule after **b** (see Fig 2).

Step 4 return to Step 1 if there are unselected violated flows, otherwise, confirm whether the temporal solution still has some violated flows. If there are no violated flows, then finish because the temporal solution is feasible, otherwise, set all the violated flows as unselected and execute from Step 1 again as long as the total number of iterations does not exceed the regulated frequency. If the loop count exceeds the

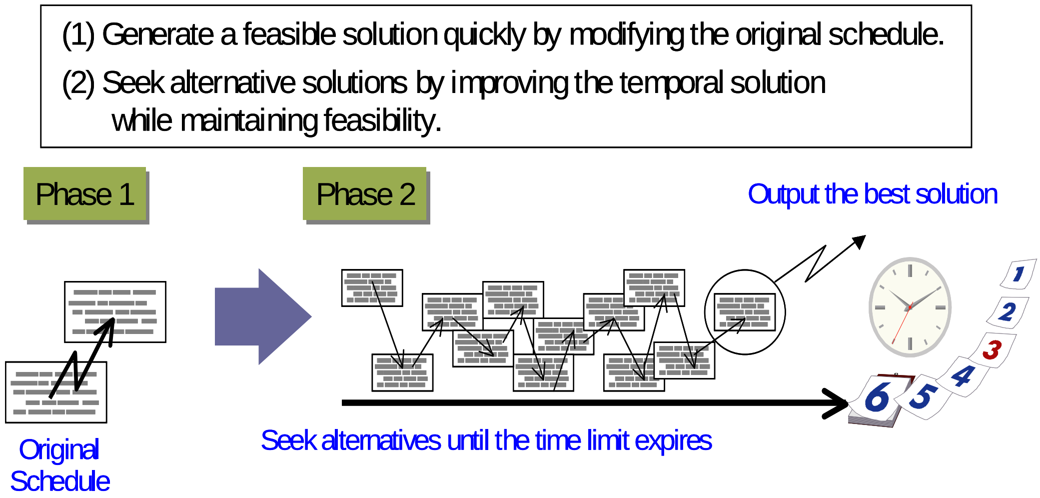

Figure 3: Two Phase Solution Approach

limitation, then finish. If this happens, the temporal solution is not feasible because violated flows are remained.

At Step 2, flow **b** is at first tried to be one of the flows with which the exchange corrects **a**'s violation and that generates no new violations. This type of exchange is called **normal exchange**. With this exchange, flow **a** is removed and instead a new flow starting from the same trip as **a** is generated (flow **c**, as shown in Figure 2). If there are several candidates for **b**, select the one with which the exchange generates a new flow that has the shortest time.

A normal exchange is not always possible. If so, select **b** under the relaxed condition that allows a new violation to occur to **b**. This type of exchange is called **forced exchange**. The basis for selecting **b** from several candidates is almost the same as a normal exchange, but the candidates with which an exchange results in some old status with violations, are excluded by referring to the exchange history.

If a forced exchange is also impossible, cut the violated link **a** and split the schedule with **a** into two partial schedules, then allocate a resource with no schedule (i.e. reserved crew or vehicle) to either of the two partial schedules. If there are no reserved resources, try to generate a new reserved resource by combining two schedules.

If the timetable has been changed to set up extra trains, in the preparation step before Phase 1, decompose the extra trains into trips and insert them into arbitrary positions of arbitrary resources' schedules. Though new violations of constraint (4) may occur in this step, it can be expected that they will be corrected through the above heuristic method.

Phase 2: Local Search for Alternative Solutions

After setting the solution generated in Phase 1 as an initial solution, the solution method searches for alternatives by a local search[10] in Phase 2. Local search is a kind of *generate-and-test method* in which a *neighborhood* of the temporal solution is generated at each iteration step. This gives a set of solutions similar to the temporal solution, and the solution with the best evaluation value is selected as the improved temporal solution.

We define the neighborhood as a set of solutions generated by normal exchanges, that is, we generate a set of solutions by executing normal exchanges for all available flow combinations.

The terminate condition of the iteration is as follows. Either the number of iterations reaches a predefined upper limit, or there exist no solutions better than the temporal solution in the neighborhood.

4 Application to a Train Operation

4.1 Prototype System

We developed a prototype system of vehicle rescheduling for a Japanese railway line. The line has about 200 vehicles, and approximately 800 trains are operated on the line everyday.

Figure 4 is a display screen of the prototype system, showing a solution as a diagram with the time scale horizontally and station labels vertically. Each trip is represented as an oblique line and each connection between trips is represented as a horizontal line.

The system consists of the following three main software components: (a) a graphical user interface (GUI), (b) a database and network, and (c) a rescheduling engine that implements the proposed solution method. The GUI provides several interactive functions: while the system shows solutions and evaluation values to the user, the user can change the scenario or settings for rescheduling, such as delay of trains, train service cancellations, extra trains, and time windows available for rescheduling. The user can also modify or fix parts of the on-screen schedule simply by pointing and clicking with a mouse.

Most of the above interactive functions can be derived naturally from the proposed model, that is, they can be regarded as changing the network flow model itself. Thus they were relatively easily designed and implemented as operations of the model in the prototype system. Examples of the correspondence between those functions and operations are as follows. (a)Delay of trains corresponds to changing the starting and/or arriving time of the related trips and deleting invalid links if violations occur. (b)Train service cancellations correspond to deleting the related trips and links. (c)Modifications of the schedule correspond to changing flows of the schedule, where the flows are displayed as lines between trips. (d)Fixing parts of the schedule corresponds to setting flow feasibility e_{ij}^k of all links connected with the nodes of the fixed parts to 0, except those links with fixed flows, where k is a resource of the schedule. If using the set partitioning/covering model instead of the proposed model, we should design those functions thoroughly outside the model, which would leads to increasing complexity of the system, and thus decreasing flexibility for adding or improving the interactive functions.

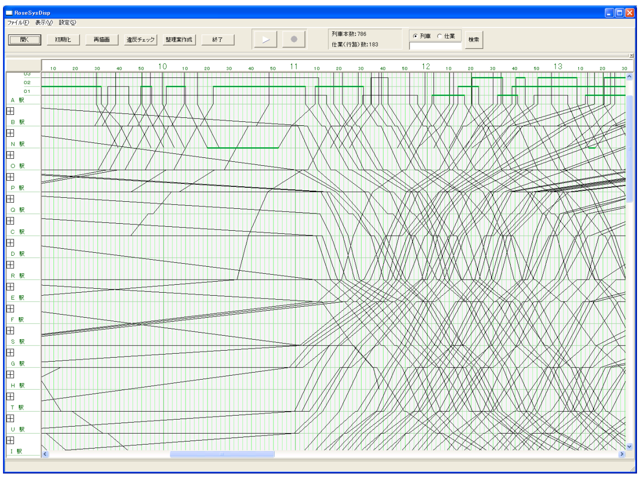

Figure 4: Prototype System

4.2 Computational Results of Real-World Data

Using real-world data from the Japanese railway line, we conducted numerical experiments for vehicle rescheduling. In the data, there were 786 trains and 185 vehicles. We set up a scenario in which a vehicle assigned to a train broke down between stations for two hours, resulting in large-scale disruption and several timetable changes, such as canceled train services and extra trains.

For the evaluation, we defined three criteria: (1) the standard deviation of the waiting duration at stations, (2)the number of vehicles with changed final destinations, and (3) differences between the new and original schedules. Items (1), (2), and (3) correspond to the first, second, and third terms of the objective function in our model. In the third term, definition (2) was selected as the cost c_{ij}^k. The total evaluation value is the sum of these three values, with all the weights w_1, w_2 and w_3 set to 1. The PC used for the experiments was a Pentium 4 3.6 GHz CPU with a 2 GB memory.

As a result, we were able to find a feasible solution with no violations. The total processing time was 422.5 sec, of which the time for Phase 1 was 9.4 sec and the time for Phase 2 was 413.1 sec. There were 83 local search iterations in Phase 2, and the average size of the neighborhood in each iteration was 2099 (Units).

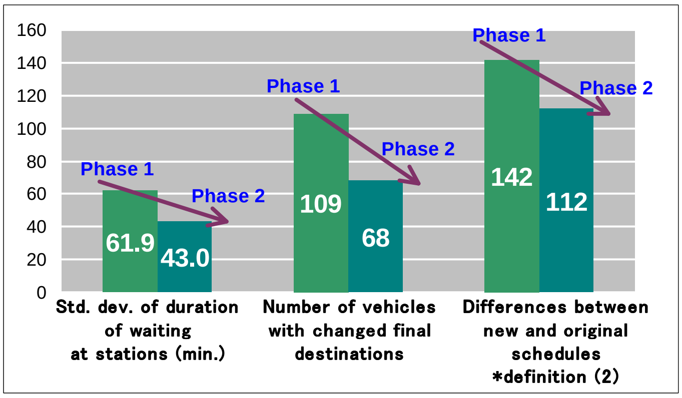

Figure 5: Evaluation Values of the Solution

Figure 5 shows the evaluation values of the solution. Though there are differences in degree of the improvement, all the evaluation values were improved in Phase 2 compared with Phase 1. Thus, we confirmed that by executing a local search using the initial solution from Phase 1, the proposed method clearly improved upon the solution found in Phase 1 and generated a better solution in Phase 2.

5 Conclusion

A basic study of modeling and solution frameworks for the crew rescheduling problem (CRP) and the vehicle rescheduling problem (VRP) was conducted. We defined a 0-1 integer programming formulation on the basis of a network flow model and designed a solution method with the concept of a two-phase modification of temporary solutions using a heuristic flow modification and local search technique.

The proposed formulation is able to represent the *differences between the new and original schedules*, which is a significant criterion for the CRP/VRP, though taking these differences into account is difficult for other related formulations based on the set partitioning/covering models. Implementation of a prototype system of vehicle rescheduling for a Japanese railway line showed that the proposed model had an advantage for designing interactive operations between users and the system, as those operations can be associated naturally to operations of the network model. Also, computational results of real-world vehicle rescheduling data from the railway line indicated that the proposed method generated a feasible solution within a practical amount of time, and on the basis of a two-phase solution approach, the proposed method improved the evaluation values of the solution.

We believe that our network-oriented modeling and solution approach is promising for developing a practical computer system for the CRP/VRP that would effectively support train recovery operations under strict time limitations. Incidentally, a software package based on the proposed formulation and method is being developed and tested for a railway line.

For future work, there are several conditions to be considered according to each situation, which have not been covered by the proposed solution method. These include rides that crews take without driving or conducting, and track limitations at stations for vehicles. We will incorporate these factors into the proposed solution method while improving the search efficiency to handle larger disruptions. It is also important to design a mathematical solution method based on the relaxation techniques. Because the mathematical method provides such a good lower bound of the optimal solution, we are able to numerically evaluate the capabilities of the proposed method, and the mathematical method itself has the potential to be more suitable for the crew/vehicle rescheduling.

References

[1] Freling, R., Paixão, J.M.P., Wagelmans, A.P.M., "Models and algorithms for vehicle scheduling", *Report 9562/A, Econometric Institute*, Erasmus University Rotterdam, 1999.

[2] Erlebach, T., Gantenbein, M., Hürlimann, D., Neyer, G., "On the Complexity of Train Assignment Problems", *Proc. 12th International Symposium on Algorithm and Computation (ISAAC 2001)*, Lecture Notes in Computer Science 2223, pp.390-402, 2001.

[3] Cacchiani, V., Caprara, A., Toth, P., "Solving a Real-World Train Unit Assignment", *Proc. 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATOMS 2007)*, pp.79-95, 2007.

[4] Caprara, A., Fischetti, M., Guida, P.L., Toth, P., Vigo, D., "Solution of Large-Scale Railway Crew Planning Problems: the Italian Experience", *Lecture Notes in Economics and Mathematical Systems 471*, pp.1-18, 1999.

[5] Fischetti, M., Lodi, A., Martello, S., Toth, P., "A Polyhedral Approach to Simplified Crew Scheduling and Vehicle Scheduling Problems", *Management Science*, Vol.47, Issue 6, pp.833-850, 2001.

[6] Haase, K., Desaulniers, G., Desrosiers, J., "Simultaneous Vehicle and Crew Scheduling in Urban Mass Transit Systems", *Transportation Science*, Vol.35, No.3, pp.286-303, 2001.

[7] Fisher, M.L., "The Lagrangian Relaxation Method for Solving Integer Programming Problems", *Management Science*, Vol.27, No.1, pp.1-18, 1981.

[8] Lübbecke, M.E., Desrosiers, J., "Selected Topics in Column Generation", *Operations Research*, Vol.53, No.6, pp.1007-1023, 2005.

[9] Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H., "Branch-and-Price: Column Generation for Solving Huge Integer Programs", *Operations Research*, Vol.46, pp.316-329, 1998.

[10] Aarts, E., Lenstra, J. K., *Local Search in Combinatorial Optimization*, John Wiley & Sons, Chichester, 1997.