Model and Solution Method for Railroad Crew and Vehicle Rescheduling

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

Abstract —**The crew rescheduling problem (CRP) and the vehicle rescheduling problem (VRP) are problems of reactively making schedules for a set of crews or vehicles in response to transport disorder and changes in train operations. We propose a 0-1 integer programming model for CRP/VRP based on the multicommodity flow network, and approximation method with heuristics and local search. The proposed model is used to formalize a "difference from the original schedule", which is a significant criterion in solving CRP/VRP. Experimental results to real-world data of vehicle rescheduling are also described.**
Keywords: crew/vehicle scheduling, rescheduling, multicommodity flow network, local search, train operation

1 Introduction

The *Crew Rescheduling Problem* (CRP) and the *Vehicle Rescheduling Problem* (VRP) are problems of reactively making schedules for a set of crews or vehicles, both are resources needed for train operations, in response to transport disorder and changes in train operations.

Most Japanese railroad lines are known for congested train networks. In addition, they form a huge and complicated traffic network because of the interconnectedness. Therefore, once small disorder occurs in one train line, it spreads over such a wide area of the network that other trains are delayed and thus significantly lowering passenger transportation efficiency. To prevent delays from escalating, train operators, who are typically veteran experts, try to immediately change the timetable. For instance, they make time alterations, change departure orders, cancel train services, and set up extra trains. To ensure the availability of these changes, they have to coordinate the schedules of vehicles and crews needed for changes under strict time limitations. It has been increasingly difficult to prepare and train skilled operators because of a lack of training time and human resources.

The CRP/VRP can be viewed as a problem for treating *the Crew Scheduling Problem* (CSP) and *the Vehicle Scheduling Problem* (VSP) especially under dynamic circumstances. CSP and VSP have been widely investigated[1]-[6]. For example, Cacchiani et al.[3] have proposed an integer linear programming (ILP) model 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 CSP by modeling it as a set covering problem (SCP) and using an enumeration algorithm with Lagrangian relaxation. Fischetti et al.[5] have considered simplified but still NP-hard case in which several depots are specified, and proposed a 0-1 linear programming formulation that can be applied to both crew and vehicle scheduling, and devised an exact method based on a polyhedral approach.

Most researches for CSP/VSP have focused on making a schedule from *scratch*, assuming there is enough time for scheduling because it will not be used right away but in the future. However, these approaches would appear

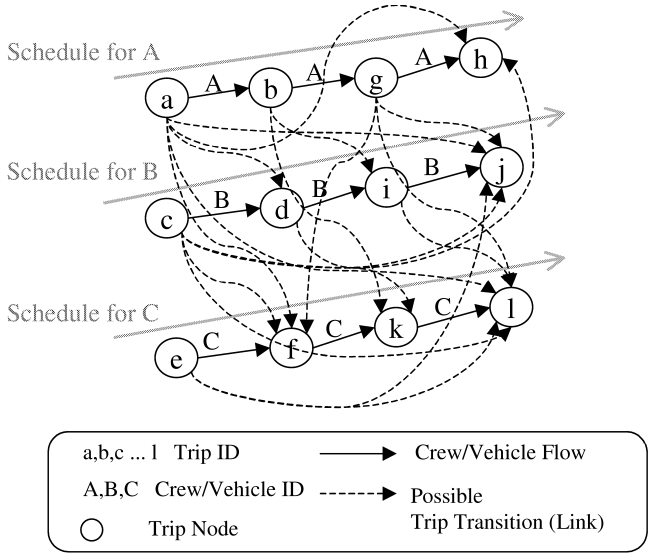

Figure 1: Multicommodity Flow Network for the Crew/Vehicle Schedule

*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

to be inappropriate for CRP/VRP because of the strict time limitations.

We propose a 0-1 integer programming model for CRP/VRP based on the multicommodity flow network, and an approximation method with heuristics and local search. Experimental results from real-world data of vehicle rescheduling are also described.

2 Problem Description and Model

2.1 Multicommodity Flow Network

A timetable for one train can be partitioned into several trips, each one starts and ends at stations in which crew change or transfer is possible. Then, a schedule for a crew can be represented as a group of several trips, which is called *pairing*. It is similar to the one described above for a vehicle. Trips are segments of a timetable partitioned by stations, where entering a depot or turn-back is possible, and then a group of several trips represents a schedule for a vehicle.

In the following sections, the word *resource* refers to vehicles or crew. By associating a vertex with each trip and a directed link with each possible trip transition, we can represent the schedule as a multicommodity flow network on a graph with the vertices and the links described above (Fig.1). In Fig.1, a solid arrow represents a resource flow, therefore a series of solid arrows is a schedule for one resource, which is sometimes called a *duty*.

When transport disorder occurs and the timetable is thus changed, the graph structure should also be changed. This may cause problems with schedule feasibility. For example, there are two trips, a and b. Trip a is set to end before trip b, and accordingly there is a link from the vertex of trip a to the vertex of trip b. There is also a flow in the link. Under these circumstances, if the end time for trip a (i.e, the station arrival time) comes after the start time for trip b because of some delay or a change in the end time for trip a, the link from a to b vanishes as the order of trip a ending before trip b is no longer available. Consequently, the flow from trip a to trip b with the *execute trip a first, and then trip b* is no longer physically available.

It is important in CRP/VRP that we should not only make a feasible schedule, but also satisfy various evaluation criteria derived from union contracts and/or company regulations. Examples of the criteria for the vehicle are *transition time between trips* and *difference in the distance traveled by each vehicle*, and for the crew are *the number of different vehicles boarded*, *the number of rides without driving or conducting*, *the amount of overtime*, and *time and duration of meal or break time*.

In addition to such criteria, which derive from specific problem instances, there is also a common criterion to CRP/VRP, *the difference from the original schedule*. If there are only a few differences, the train operator is able to rapidly understand and confirm the changes. Besides, minimizing the differences also enables him or her to reduce the workload for train schedule alternations and thus suppresses confusion in the field staff that may arise from these schedule changes. The criterion is inherent in dynamic scheduling problems, which are different from the static ones for schedules from scratch.

2.2 Mathematical Formulation

2.2.1 Notation

V : Set of nodes $\{1, \cdots, n\}$. It consists of trips and dummy nodes, which 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 (node s) and end (node 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 for t to t, are elements of E. The links, which end at s, are not included in E. The links, which start from t, are also not included in E. For notational convenience, s and t represent nodes 1 and n, 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.

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

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, which 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 ($k \in R,\ i,j \in V$)

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

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 the links that are inconsistent with time and location, tracing the links results in physically valid allocations to the trips for any resources. However, since there are certain vehicles which cannot be allocated to certain trips due to a vehicle-specific requirement, for instance, we define the feasibility for each resource by e_{ij}^k.

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 0.

Where both i and j are elements of V, but the directed link (i,j) is not always in E because the graph structure is changed due to transport disorder and timetable changes

$\boldsymbol{B^k}$: $n \times n$ matrix of which (i,j) element is b_{ij}^k.

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

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

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

Each element in $\boldsymbol{B}$ is the total sum of flows on each directed link in the original schedule. That is, let b_{ij} be (i,j) element of $\boldsymbol{B}$, then $b_{ij} = \sum_{k=1}^{m} b_{ij}^k$.

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

$c(\boldsymbol{X^k})$ is independent of the other schedule. If there is a cost relating to several resources, it should be approximated by separating it into several cost functions of each resource. For the "difference in the distance traveled of each vehicle", by setting a base distance, the cost function $c(\boldsymbol{X^k})$ would be the difference between the distance of vehicle k and the above base distance. If there are several costs to be considered, $c(\boldsymbol{X^k})$ is the weighted sum of these standardized costs.

2.2.2 0-1 Integer Programming Formulation

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

$$\text{Minimize} \quad w_1 \sum_{k=1}^{m} c(\boldsymbol{X^k}) \; + \; w_2 \sum_{i=1}^{n} \sum_{j=1}^{n} |p_{ij}| \qquad (1)$$

subject to

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

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

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

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

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

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

Where $w_1, andw_2$ are weight constants and p_{ij} is the (i, j) element of the matrix $\boldsymbol{P}$ described as follows.

$$\begin{aligned} \boldsymbol{P} &\equiv \boldsymbol{B} - \boldsymbol{X} \\ &= \sum_{k=1}^{m} \boldsymbol{B^k} - \sum_{k=1}^{m} \boldsymbol{X^k} \\ &= \begin{pmatrix} \sum_{k=1}^{m}(b_{11}^k - x_{11}^k) & \cdots & \sum_{k=1}^{m}(b_{1n}^k - x_{1n}^k) \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^{m}(b_{n1}^k - x_{n1}^k) & \cdots & \sum_{k=1}^{m}(b_{nn}^k - x_{nn}^k) \end{pmatrix} \end{aligned}$$

Therefore, $|p_{ij}| = |\sum_{k=1}^{m}(b_{ij}^k - x_{ij}^k)|$ (strictly speaking, for $(i,j) \notin E$, $|p_{ij}| = |\sum_{k=1}^{m} b_{ij}^k|$ since the (i,j) element of $\boldsymbol{X^k}$ is 0). It represents the difference of the flow amount in the directed link (i,j) between the original schedule and the rescheduled result. Hence, the total sum of $|p_{ij}|$ at the 2nd term of objective function (1) corresponds to the criterion *difference from the original schedule.*

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 in 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 just one resource. That is, one of the resources must be allocated to all trips, constraint (7) : The amount of flow 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 vehicles without driving or conducting in the crew scheduling, (7) is replaced by the following inequality version, which allows multiple resource allocations to a trip.

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

2.3 Discussion of the Model

If the original schedule has become infeasible due to transport disorder and timetable changes, the train operator generally tries to correct the violations by exchanging a part of the schedule with another one. An example of this partial schedule exchange is shown in Fig.2.

In this example, two partial schedules, both of which have two trips, are exchanged, then the number of changed trips is four. If we regard the change in the schedule to be identical to the change in the trips, the amount of changes by one exchange would vary according to the number of trips involved in the partial schedules to be exchanged. However, for the train operator, who usually changes the schedule by partial exchanging, the means of changing each trip does not seem significant. In the train operation field, the field staff needs to specify when and where the schedule will be changed from the original so as to make the necessary arrangements such as notification to the crew or the worker involved before the train reaches the schedule change point. Therefore, it seems more effective from both viewpoints of the decision support for operators and the work support for field staff, to adopt the number of partial schedule exchanges as a concrete indicator for evaluating the difference from the original schedule.

The 2nd term of objective function (1) enables us to count the number of partial schedule exchanges. That is, because the term takes four at one exchange regardless of the number of trips involved, the term becomes $4n$ for exchanges of n times.

It is well known that the CSP/VSP can be modeled as a set partitioning problem (SPP) or a set covering problem (SCP), which defines schedule candidates as a subset of trips, is formulated as a problem in deciding the optimal cost combination of the candidates with all the trips covered. Because this type of formulation simply treats a schedule as a group of trips, it cannot represent a connection between trips. Therefore, unlike the proposed model, it is difficult for related models to be used for explicitly representing changes in the partial schedule, which can be treated by using the connection relationship between trips.

On the other hand, for SPP/SCP formulation, a variety of mathematical methods have been proposed to find optimal or near-optimal solutions. For vehicle scheduling, there are several methods, such as the one mentioned above[4] with the enumeration algorithm and the Lagrangian relaxation method[7], and one[9] for bus drivers using branch and price[9], which is a combination of branch and bound and column generation[8].

However, our proposed model, is nonlinear since the objective function contains the absolute value on the 2nd term, therefore, it is likely to be difficult to design a relaxation problem by which good lower bounds of the optimal solution can be provided. Even if it is a small problem to be solved, the model size becomes relatively larger because of large number of flow variables to all combinations of the directed links in E and the resources. Therefore, for practical purposes, it seems difficult to design a mathematical solution method to the proposed model.

3 Solution Method

3.1 Concept

Because the CSP/VSP are dynamic problems that should 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. Phase 1 generates a feasible solution by modifying the original schedule, and then Phase 2 continues to search for alternatives for improving the evaluation value until the time limit expires while maintaining schedule feasibility. Based on this approach, we can perform scheduling activity that is adaptable to the time remaining for train recovery, which can vary according to the situation.

3.2 Partial Exchange Heuristics for Phase 1

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

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

Step 2 select a flow (**b**) of any other schedule except for the one with **a**.

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

Step 4 return to Step 1 if there are unselected invalid flows, otherwise, confirm whether the current solution still has some violated flows. If there are no violated flows, then finish since the current solution

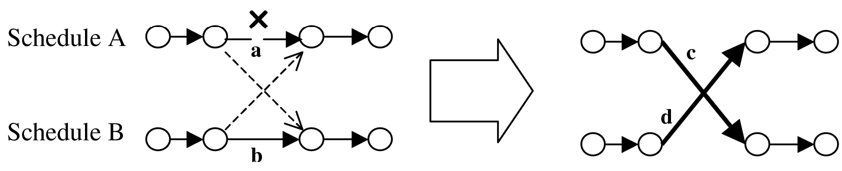

Figure 2: Partial Schedule Exchange

is feasible, otherwise, set all the violated flows 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 limitation, then finish. In this case, the current solution is not feasible because of the remaining violated flows.

In Step 2, flow **b** must be one of the flows by which the exchange results in correcting **a**'s violation and no new violation are generated. This type of exchange is called **normal exchange**. By 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 Fig.2). If there are several candidates for **b**, select the one by which the exchange generates a new flow with 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 the several candidates is almost the same as a normal exchange, but the candidates, by which the exchange results in going back to the old status with violations, are excluded by referring to the exchange history.

If 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, as a preparation step before Phase 1, decompose the extra trains into trips and insert them into arbitrary positions in arbitrary duties. 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.

3.3 Local Search Algorithm for Phase 2

By using a feasible solution from Phase 1 as the initial solution in Phase2, the solution method is used to search for alternatives using a local search[10]. Local search is a kind of improvement algorithm, in which a *neighborhood* of the current solution is generated at each iteration step, which is a set of solutions similar to the current one, and the solution with the best evaluation value is selected as an improved solution.

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

4 Computational Results

We developed a prototype system for vehicle rescheduling (Fig3), and conducted numerical experiments with real-world data of a Japanese railroad line. In the data, there were 786 trains and 185 vehicles. We used a PC with a Pentium 4 3.6 GHz CPU with a 2 GB memory.

A scenario was set up so that a train stopped on a line between stations for two hours because of a breakdown, resulting in large-scale disorder. We then executed rescheduling by using the proposed method after making several changes to the timetable, such as train service cancellations and setting up extra trains.

We defined the two following concrete evaluation functions; (1) the standard deviation of transition time between trips, and (2) the number of the partial schedule exchanges. (1) and (2) correspond to the 1st and 2nd terms of the objective function in our model, respectively. The total evaluation value of the solution is the weighted sum of these standardized scores.

As a result, we were able to devise a feasible solution with no violations. The total processing time was 716.3 sec, in which the time for Phase 1 was 9.4 sec and the time for Phase 2 was 706.9 sec. There were 105 local search iterations in Phase 2, and the average size of a neighborhood in each iteration was 2099.

Table 1: Evaluation Results of Solution

	Phase 1	Phase 2
std. dev. of transition time between trips (min.)	71.6	40.8
number of exchanges	126	108

Table 1 shows the evaluation values for the solution. From these results, we confirmed that by executing a local search with the initial solution from Phase 1, the proposed method improved upon the solution found in Phase 1 and generated a better solution in Phase 2.

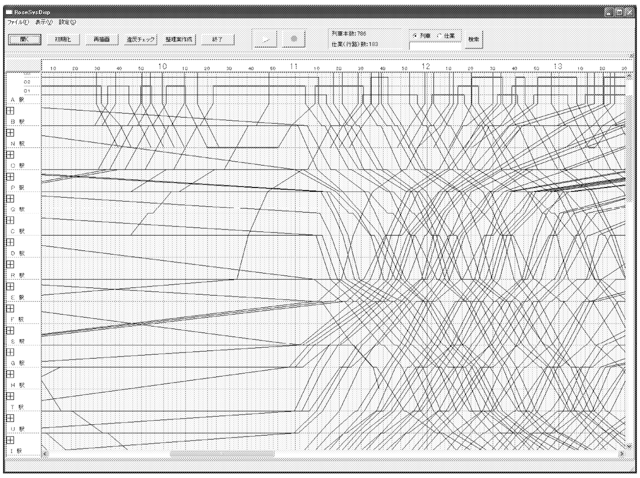

Figure 3: Prototype System

5 Conclusion

For the CRP/VRP, we defined a 0-1 integer programming model based on the multicommodity flow network and designed a solution method using heuristics and local search. The proposed model is able to represent the "difference from the original schedule" based on partial schedule exchange, which is a significant criterion to CRP/VRP, though it is difficult for other related models using set partitioning/covering formulations.

The results of numerical experiments with real-world data of vehicle rescheduling showed that the proposed method generated a feasible solution within a practical amount of time, and based on a two-phase solution approach, the proposed method gradually improved the evaluation values of the solution.

For future work, as there are additional conditions to be considered according to the individual problems, such as the rides without driving or conducting, and union contracts for crew, and also the track limitations at stations for the vehicles, we will improve the proposed solution method to cover them. It is also important to design a mathematical solution method based on several techniques such as branch and bound, which provide a good lower bound of the optimal solution and therefore enable us to numerically evaluate the capabilities of the proposed method.

References

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

[2] T. Erlebach, M. Gantenbein, D. Hürlimann, and G. Neyer : "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] V. Cacchiani, A. Caprara, and P. Toth : "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] A. Caprara, M. Fischetti, P. L. Guida, P. Toth, and D. Vigo : "Solution of Large-Scale Railway Crew Planning Problems: the Italian Experience", Lecture Notes in Economics and Mathematical Systems 471, pp.1-18 (1999)

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

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

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

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

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

[10] E. Aarts, and J. K. Lenstra : "Local Search in Combinatorial Optimization", John Wiley & Sons, Chichester (1997)