Automatic generation of car shunting scheduling in railway car depots

Y. Nagasaki & S. Takahashi
Advanced Technology R&D Center, Mitsubishi Electric, Japan

Abstract

A train schedule defines each train's arrival/departure time at a car depot. Between the arrival and departure times, every train has to have its designated works, such as inspection and cleaning of the cars, completed, which are predefined by a car operation schedule. The planner of car shuntings at a car depot must create a schedule that satisfies the above-mentioned conditions, while also taking care of other constraints, such as work places, length of tracks, crossover blockings of shunting routes, and workers' schedules. In this article, we describe a system that automatically generates car shunting schedules. First, the initial solution phase roughly produces the order of the shuntings and works, the places where the works are to be conducted, and the routes of the shuntings by using a rule-based algorithm. In the second phase, the schedule is modelled by using the Programming Evaluation and Review Technique (PERT) to detect any condition violations and applied to the other rule-based algorithm to resolve these violations. This system changes the schedule by using different methods, such as changing a track to hold, changing a route to shunt, and swapping the order of works. Depending on the conditions of the violations, the system searches for the appropriate method to resolve theses violations step by step. The degree of the condition violations is evaluated by using the total amount of the delay of train departures. We also present the search results that are generated by the system in this paper.

Keywords: depot, shunting, scheduling, PERT, rule-based algorithm.

1 Introduction

There are four major types of railway scheduling, i.e., train, crew shift/roster, car-operation, and depot-shunting scheduling. The problem of shunting

WIT Transactions on The Built Environment, Vol 103, © 2008 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CR080571

scheduling in a depot is affected by the conditions designated by the train and car-operation scheduling. A train schedule defines each train's arrival/departure time at a car depot, and the car-operation scheduling designates the schedule of inspection and cleaning for each car, which must be done in the depot. The depot's shunting scheduler must create a schedule that satisfies the above-mentioned conditions, while taking into account other constraints, such as work places, lengths of tracks, crossover blockings of shunting routes, and workers' schedules.

Schedulers currently create shunting schedules mainly by hand. Even though a computerized system has been introduced, the system only displays a created schedule in the form of a diagram and points out the constraint violations. Although this type of system supports schedulers in creating schedules, it does not automatically create schedules. Therefore, the schedulers still do all of the decision making.

There have been several previous works on the automatic scheduling of shunting. Tomii and Zhou [1] used a combination of a genetic algorithm (GA) and PERT for depot shunting scheduling problems. Sato et al. [2] used constraint programming (CP) and took rescheduling into consideration in case of condition changes. Freling et al. [3] used mixed integer programming (MIP) and column generation heuristics, and solved the car assignment and shunting scheduling problems simultaneously.

We propose a system that can automatically generate a depot-shunting schedule. We use rule-based algorithms and PERT in this system. We think that rule-based algorithms are suitable for reflecting on the schedulers' experiences, and PERT enables the system to easily and quickly manipulate a shunting schedule. The system takes each train's arrival/departure time at the depot and the designated works for each train as input, and produces a shunting and working schedule as output. The system also uses some constraint information related to the target depot.

At first the system creates an initial solution using a rule-based algorithm. Only the order of works and the holding tracks are considered in the initial solution. Then, the PERT structure that represents the initial solution is created and calculates a strict amount of time for each shunt and required work. Another rule-based algorithm repeatedly modifies the PERT structure and tries to satisfy all the conditions.

In this paper, the word "work" means a task that should be applied to each train, for example an inspection or a cleaning. The word "worker" means a person or a team who are in charge of accomplishing the works. The word "condition" means the designated arrival/departure time at the depot, and the word "constraint" means the other restrictions, such as the crossover blockings of the shunting routes and the workers' schedules. Therefore, if a constraint violation remains, a solution is physically infeasible. However, even if a condition violation remains, a solution is physically feasible although it is not actually performable.

At first in Section 2, we enumerate the conditions and the constraints of the depot shunting problems. In Section 3, the first phase to generate an initial

solution is discussed. Section 4 shows the PERT structure that represents the depot shunting problems and the methods for modifying the solutions by manipulating the PERT structure. The second phase to search for a practicable solution and to escape the local optimum solutions is proposed in Section 5. Finally, the conclusion of this paper is given in Section 6.

2 Depot shunting scheduling problem

2.1 Track layout and a shunting schedule diagram

Figure 1 shows an example of a track layout in a depot. This layout is very simplified and there are many more tracks that have a complicated relation of connections in a real depot. There are many tracks for holding, where trains arrive and depart. There are also many tracks where works, train cleanings, and inspections are conducted. In this example, holding tracks and work tracks are serially connected, but in some depots trains have to change direction during shunting between holding tracks and work tracks.

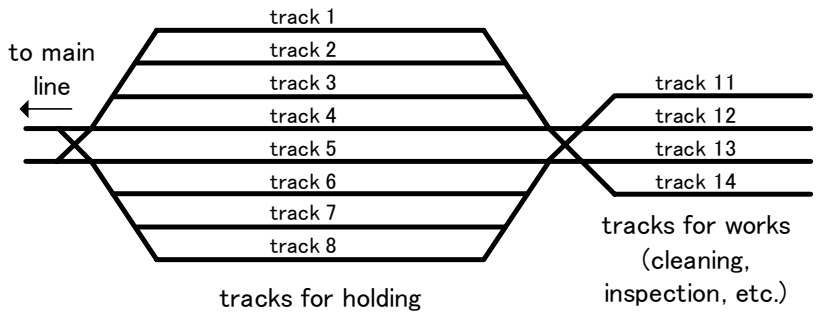

Figure 1: An example of track layout in a depot.

Figure 2 is an example of a diagram that represents a schedule of the shunts and works in a depot. In this figure, the horizontal axis corresponds to time and the vertical axis corresponds to tracks/workers. A rectangle on a track line represents a hold of a train on the specific track or a work by a worker. A slanted line connecting the corners of rectangles represents a train shunt. This figure is a kind of Gantt chart.

2.2 Conditions and constraints of the depot shunting scheduling problem

The conditions concerning the target trains and their assigned works are inputted into the system. Each train has a designated arrival and departure time. During hold in a depot, all of the designated works for a train must be completed. Some trains may not depart the depot on the same day, so they are held overnight.

The following constraints of the depot shunting problem are taken into account in this paper.

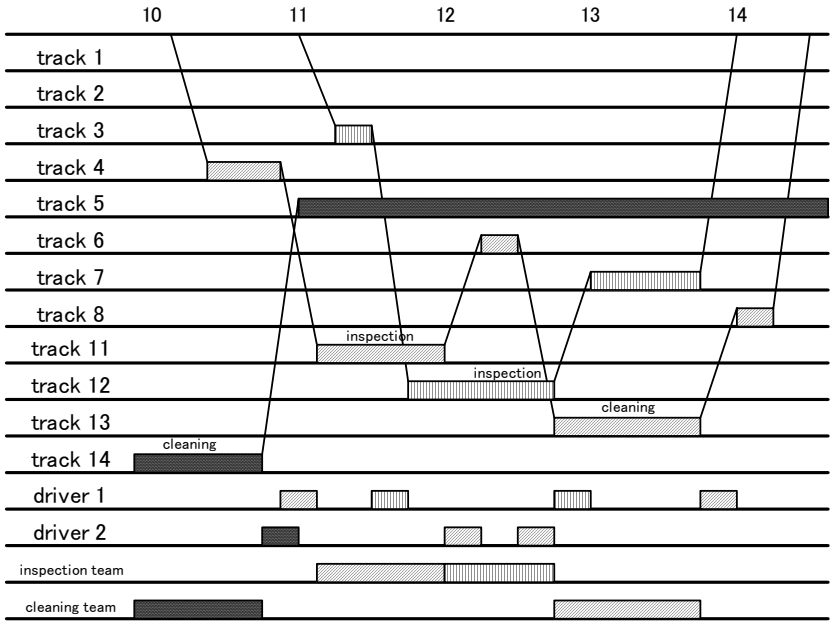

Figure 2: A diagram that represents a schedule of shunts and works.

(1) Track layout

The track layout of the depot is given. A train must hold on any track in the layout, and must use any route in the layout when it is shunting. A train must not crossover another train nor conflict with another train's route when it is shunting. The required time for the shunting of each route is also given. Depending on its train type, some trains may not use some tracks and routes due to restrictions (track length, electrified or not etc.).

(2) Work constraints

Every work must be done on a track that is equipped with the appropriate facilities for the work. Every work must have the appropriate workers assigned to it. In addition, every shunting must be assigned an appropriate driver. Special care is also needed when assigning the duty hours of the workers. The required time of each work depends on the work type.

3 Initial solution generation

A rule-based algorithm that is based on the input conditions is used to generate the initial solution. In the initial solution, the orders of the works and tracks for holding are already decided.

In the sequential order of the train's arrival time to the depot, each train's work order and tracks for holding are decided. Taking care to monitor the workers' duty hours, the work order is decided reflecting the recommendations of a human scheduler. The holding tracks are also determined. In the process, the

system also takes care of any interference with trains that have already had been assigned schedules, and decides the order of shunts and holds between different trains.

4 PERT structure that represents the shunting schedule

4.1 Condition violation detection by the PERT structure

A PERT structure that represents a shunting schedule is created based on the initial solution or a previously created solution. Figure 3 shows an example of such a structure. A PERT structure is a directed acyclic graph (DAG) of the graph theory. The vertices in the structure represent the begin/end of a shunt/hold/work. The arcs in the PERT structure represent each train's shunt/hold/work, and the relationships between trains and works. For example, if there is a train that uses a route interfering with another train's route, an arc that represents a constraint of the order of shunting is required. If there are multiple works to be done by the same worker, arcs between the vertices of the works are required to express the worker's constraint.

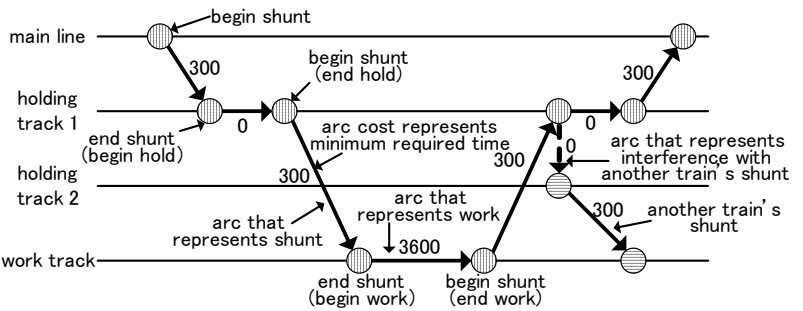

Figure 3: An example of the PERT structure that represents the shunt schedule.

If there is a constraint violation or an inconsistent condition in the solution, the graph structure will contain loops, therefore the system can easily detect if a solution is infeasible or not by using PERT calculations.

The solution is physically feasible if no loop is found in the PERT structure. However, the solution may still contain train departure delays from the depot. In other words, the solution should be modified until all departing trains meet their designated departure times.

4.2 Solution modification by manipulating the PERT structure

A solution schedule and the corresponding PERT structure are convertible with each other if the solution and the PERT are feasible and consistent. Therefore, manipulating the PERT structure modifies the solution.

Figures 4 and 5 show examples of PERT structure manipulations. The order of shunting is swapped in Fig. 4 by changing the position of a shunting order arc. The train's holding track is changed in Fig. 5 by rewriting the information on the vertices and changing the shunting routes.

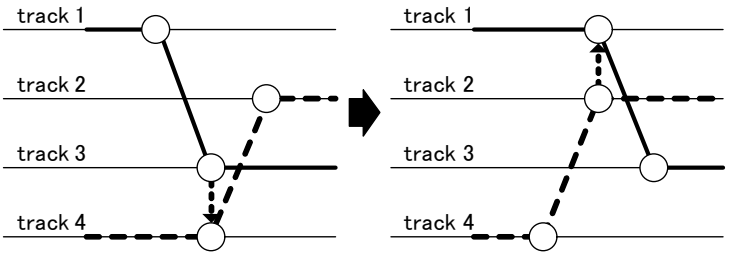

Figure 4: An example of swapping the order of shunting.

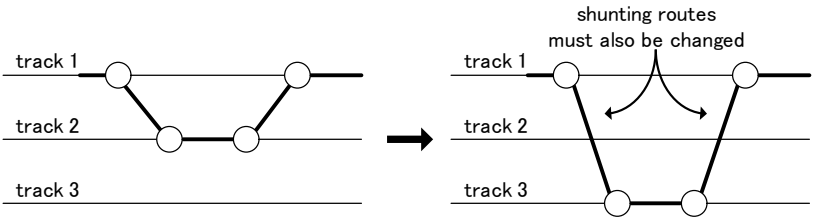

Figure 5: An example of changing stay tracks.

Table 1 shows the methods for manipulating the PERT structures that are implemented in our system. The system searches for a feasible, consistent schedule whose departures are all in time by repeatedly using these methods.

Table 1: Methods for manipulating the PERT structure.

Method	Description
Swap shunt order	Swapping order of shunts that use conflicting routes
Swap track use order	Swapping order of holding a track between two trains
Change holding tracks	Changing holding tracks for a train
Swap work order	Swapping order of work on a train
Swap worker order	Swapping order of work between two trains
Change work teams	Changing workers for a work

4.3 Investigating the causes of late departures

When late departures are found in a solution, there must be causes for the latencies. The system can investigate these causes by tracking back the critical paths from the late departure vertices on the PERT structure.

Table 2 shows the relationship between a critical path terminus, the causes of lateness, and the candidate methods of modifying the solution. The system can enumerate the causes and candidate methods to modify the solution by using the relationships given in Table 2.

Table 2: Relationship between critical path terminus, causes of latency and candidate methods of modifying solution.

Vertex that is late for time	Critical path terminus	Causes of latency	Candidate methods of modifying olution
Departure vertex from depot	Shunting order arc	Waiting for another train's shunting	Swap shunt order Swap track use order Change holding tracks
	Arrival vertex to depot of train	Wrong work/shunt order	Swap work order Swap shunt order
		No possibility of in time schedule	Relax constraints
	Worker's order arc	Waiting for worker team	Change work teams Swap work team order Swap work order
	Time constraint arc	Waiting for available worker/track/route	Swap shunt order Change holding tracks Change work teams
Arrival vertex to depot	Shunting order arc	Waiting for another train's shunting	Swap shunt order Change holding tracks

4.4 Evaluation of lateness

We use the total of all lateness in a solution as the evaluation value. For departing trains, the practicable time of the final departure from the depot in a solution is compared with the designated departure time, and its lateness in seconds is added to the evaluation value for the solution. For trains being held overnight, all the works assigned to them should not be delayed to the next day. Therefore the system detects and assigns the works scheduled for the next day as delayed, and this degree is added to the evaluation value.

Since the evaluation value is the total of all lateness, an evaluation value zero means that the solution satisfies all the conditions and is acceptable as a final schedule.

5 Searching a practicable solution

5.1 Greedy algorithm

We use a greedy algorithm as a trial, which is the simplest search algorithm for this kind of combinational optimization problem. In the greedy algorithm, all the

modification methods are tested and the method with the best evaluation value is used. The next search is iterated based on the modified solution, and thus, the solution is gradually improved.

To reduce the search time, we added some modifications to the greedy algorithm. The flowchart of the new algorithm is shown in Fig. 6. In the depot shunting scheduling problem, a modification to the early part of a schedule may materially affect the latter part of the schedule. Therefore, in our system, the modification targets in the PERT structure are enumerated, and the targets are sorted in order of time. From the first target, all the modification methods are tested and if the evaluation value is improved, the best method in the evaluation value is used. If the evaluation value is not improved, the modification methods for the next target are tested.

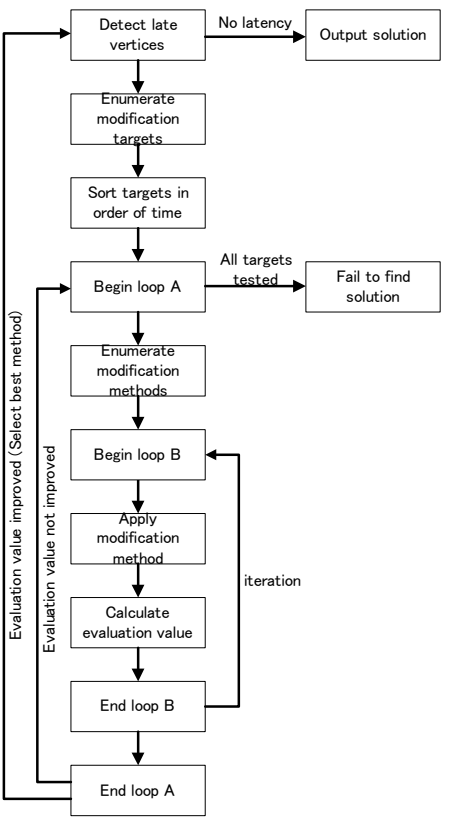

Figure 6: Modified greedy algorithm flowchart.

5.2 Escape from local optimum solutions

The results of the greedy algorithm are easily trapped into local optimum solutions and are not generally the global optimum. With the depot shunting

scheduling problem, this means that we may not be able to get a solution that satisfies all the conditions.

To escape these local optimum solutions, we introduce another rule-based algorithm. Some rules that are combinations of the modification methods shown in Table 2 are defined as follows.

- Track occupancy
- Worker team occupancy
- Creating a new shunting
- Removing an existing shunting
- Departure waiting policy

When a search is trapped into a local optimum solution, the rules are applied depending on the situations.

For example, if a worker is too busy at a certain time and this delays a train, globally changing the order of works may resolve the situation. However, simply swapping the order of consecutive works may worsen the evaluation value, and in such a case multiple iterations of swapping the order of works may improve the evaluation value. The system automatically searches the schedule of each worker and applies the appropriate rules, which are combinations of multiple modification methods.

5.3 Search result

We performed some experiments on our system for some problem data from a real depot. The results are shown in Table 3. The experiments were executed on a PC (Celeron D 2.53 GHz/main memory 1 GB, gcc 3.2.2 on RedHat Linux).

Table 3: Experiments result.

Data	# of trains	# of works	initial solution evaluation (in sec.)	final solution evaluation (in sec.)	# of searches	search time (in sec.)
data 1	25	24	4,365,600	6,420	1,270	7.9
data 2	25	24	3,235,260	0	919	12.5
data 3	24	22	2,919,120	5,400	7,021	28.8

Currently the system can completely solve a problem for only one set of data out of three. For the remaining two sets of data, the final solutions were still trapped by the local optimum. We are now attempting to improve the rule-based escaping algorithm.

6 Conclusion

Our system's final solutions may still contain some late departure trains. However, from the final solutions that have been generated, a human scheduler

can easily resolve the unsatisfactory conditions that remain by making some modifications. This contributes to reducing the scheduler's workload.

We are testing the system for some other complicated data, and are continuously improving the system.

References

[1] Tomii, N. & Zhou, L. J., Depot shunting scheduling using combined genetic algorithm and PERT, *Proc. of COMPRAIL 2000*, pp. 437–446, Bologna, 2000

[2] Sato, T., Kakumoto, Y. & Murata, T., Shunting Schedule Method in a Railway Depot for Dealing with Changes in Operational Conditions, *IEEJ Trans. EIS Vol.127 No.2*, pp. 274–283, 2007 (in Japanese)

[3] Freling, R., Lentink, R. M., Kroon, L. G., & Huisman, D., Shunting of Passenger Train Units in a Railway Station, *Technical Report ERS-2002-074-LIS*, accepted for publication in Transportation Science, Erasmus University, Rotterdam, Netherlands, 2002