

Crew operation scheduling based on simulated evolution technique

K. Kataoka & K. Komaya

Industrial Electronics & Systems Laboratory, Mitsubishi Electric Corporation, 8-1-1 Tsukaguchi-Honmachi, Amagasaki, Hyogo 661-8661, Japan
E-Mail: kataoka@soc.sdl.melco.co.jp

Abstract

Crew operation scheduling problem is a large-scale combinatorial problem. Only human experts could make schedule with trial and error in a long time by hand. We propose a new framework for crew operation scheduling using simulated evolution technique, which is used to simulate trial and error approach effectively. Several kinds of knowledge base are defined to raise the effectiveness of the search algorithm to find constraint satisfaction solution. We can find a feasible solution in several minutes with our prototype system, which is developed for the high-frequency subway line.

1 Introduction

Since a crew operation scheduling problem is a large-scale combination problem, only human experts can make schedule with trial and error in a long time by hand. For example, as the density of train diagram in urban railways goes higher, crew operation scheduling also goes more complex. Instead, for excess of user availability, the interval of diagram modification goes shorter and shorter. Therefore, the support for scheduling is a great problem.

The quality of crew operation scheduling was not main aspect in some reasons: non-systematicness of the process of scheduling, the fuzziness of evaluation for scheduling by human experts, no evaluation by passengers themselves instead of the train diagram. It

was hard to search only constraint satisfaction schedule even for human experts.

We propose a new framework for crew operation scheduling using simulated evolution technique, which is one of optimization method based on simulating biological evolution. Constraint satisfaction solution can be found with this method applying the similar search process to human experts' thought, which can be refined by the user in this framework.

In this paper, we analyze the problem structure first, then propose the framework adequate for the structure. We also show the effectiveness of this framework through experimental results.

2 Train crew operation scheduling problem

2.1 Train crew operation schedule

Generally, the amount of crew shifts is given beforehand and human experts prepare expected amount of shifts to assign all train travel to each shifts. No crew can continuously drive train too long, therefore regular shift must have breaks including meal break. Crews may change driving train to be relieved by another crew generally only at some designed handover stations where they can take a break. Each of train travel logically divided at these stations is assigned to crew shifts, as scheduling unit. We call them '*resource*' which is defined as a portions of train travel between the departure time of a certain station and the arrival time of another station. Each train must be driven only by single crew, another saying, each resource must belong to a certain shift in principle.

In addition to a set of resource as elements, a scheduled shift must be fixed their work pattern and crew depots it belongs. Usually, there are morning work (morning part of night shift), daytime work, night work (midnight part of night shift), and some extra works are defined as work pattern when needed.

2.2 Difficulty

Each shifts has some properties, such as compulsory working hours, driving hours, rest hours. Additional properties are also considered in some cases which is difficult to formulate such as fatigability based on driving train type and geographical features of driving area.

Each shift must satisfies many explicit and implicit constraints,

some of them are fuzzy ones, such as meal break. When changing a set of resources belonging to a shift, value of some properties cannot be changed continuously. It is difficult to find solution satisfying both of all constrains for schedule and shifts, satisfying all resource assignment.

Off course, human experts have same difficulty to make schedules. They make schedules to repeat generate and test many times based on expertise, sometimes with some extra consideration to allow constraint violation. Even human experts hardly can find constraint satisfaction schedule.

3 Scheduling using simulated evolution

3.1 Simulated evolution

Simulated evolution technique is a kind of solving combinatorial problems based on an analogy between the natural selection process in biological environments by iterative improvements[1]. The biological solution to the selection process is the evolution from one generation to the next one by eliminating ill-suited designs and keeping near-optimal ones. Nature also has mutation process to prevent the development of species from getting stuck at local optima.

In simulated evolution, a solution is considered as a set of species (partial solution), and evaluation function of solution consists of each value of species. If only better species survive at next generation, value of solution also might be better. No assurance to find optimum solution is for simulated evolution, but it is known to find near-optimum solution effectively.

3.2 Application to crew operation scheduling

The problem structure of crew operation scheduling has some similarities to the solving process of simulated evolution, as follows:

- (a) Crew operation schedule consists of a set of shifts.
- (b) It is impossible for all shifts to get their best cost without violating all constraints. The cost of each shifts and status of free resource highly have influence on cost of schedule. Another saying, each shifts is in competition with each other.
- (c) Selection process in simulated evolution seems like try and error process in making schedules by human experts.

For these similarities, we apply simulated evolution to crew operation scheduling problem (figure 1). There are four steps in simulated evolution:

[initial generation]

Initial schedule consists of a set of shifts, which are generated with simple search algorithm (ex. hill climbing method) from initial set of free resources. If some resources cannot belong to any particular shifts, it is considered as dummy shift with single resource, which doesn't satisfy all constraints.

[evaluation]

Each shift and schedule itself are evaluated their cost with their own function defined as below. If no dummy shift remains, constraint satisfaction solution is found and this search process ends. Otherwise, schedule is treated as temporal solution.

[selection]

Any dummy shifts and some particular shifts with high cost in a schedule are selected and all resources they have are released, in other saying, dead. Selections mechanism is controlled under the cost of each shift. Released resources make new set of free resources, which presents partial search spaces for next generation schedule.

[improvement]

From new set of free resources, another set of shifts are generated as new generation using same algorithm of initial generation.

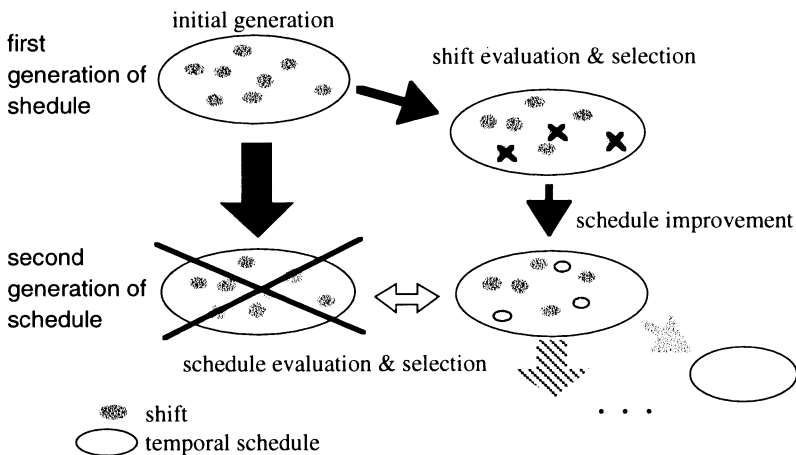


Figure 1: Simulated evolution

These step (evaluation -> selection -> improvement) is iterated until no free resource remains by gradually narrowing search space. Randomness is introduced in selection step to simulate biological competition.

4 Scheduling process

4.1 Working pattern condition

Not all constraints for scheduling are explicitly defined as labor agreement. Some of them are implicit ones based on usage or heuristics for finding feasible solutions. Working pattern condition is prepared to describe these constraints for every working pattern, such as meal break hours limitation. Generated shifts are suited for this working pattern condition.

4.2 Connected graph

Based on multi-layer gathering model[2], train travel is formalized as a set of resources assignable to shifts beforehand. Each resource has possibility to join another resource to be a part of shifts, if all local constraints are satisfied. We mark these two resources with connected graph. When expanding shifts, we can search only marked resources without considering any local constraint again and again.

4.3 Shift generation

Best-first search method is applied for the process to generate each shift. Owing to a huge search space, this process requires cpu time to solve real world problem. Three kinds of knowledge base are prepared to decrease search space with heuristics: seed resource selection knowledge, search width restriction knowledge, and search depth restriction knowledge (figure 2). They are also used to avoid generating same shift-pattern, including some randomness.

Seed resource selection knowledge is based on the process of human experts, to select first resource (seed resource) of shift as beginning or ending of work.

When searching next resource to join with the seed resource, to check every marked resources needs high time. To cut search time, search width restriction knowledge is used. In the beginning of search, it narrows search space only checking low-cost resources. According

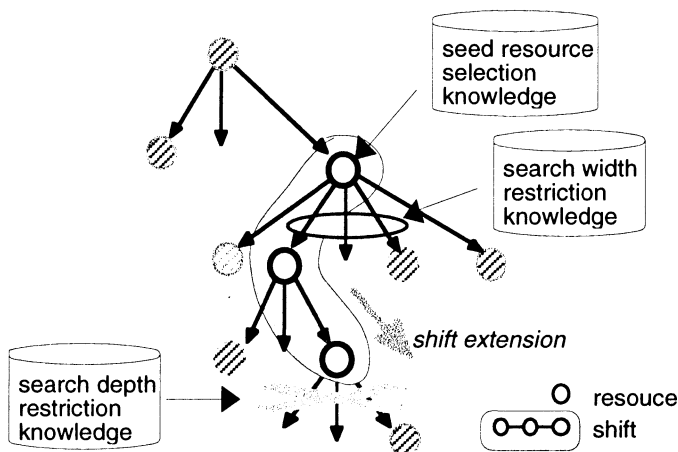


Figure 2: Knowledge for shift generation.

to search process goes on, all resources are checked not to miss the feasible solution.

Search depth restriction knowledge is used to avoid centralization of resources for several shifts.

4.4 Evaluation

We define the adaptivity of i th shift as $f(x_i)$, as below:

$$f(x_i) = \sum_j (Ta_j - Tr_{ij})^2 \rightarrow \min \quad (1)$$

where :

Ta_j : best value of property j

Tr_{ij} : value of property j in shift x_i

We define the adaptivity of schedule x as $F(x)$, as below:

$$F(x) = \sum_i f(x_i) + \alpha \sum_i f'(x_i) \rightarrow \min \quad (2)$$

where :

x : crew operation schedule

$f(x_i)$: adaptivity of generated shift x_i

$f'(x_i)$: adaptivity of remained resource x_i

α : modulus

4.5 Selection

The cost of constraint satisfaction for the schedule is influenced on

dispersion of free resources, therefore selection only based on the adaptiveness of shifts is not effective. Some patterns of shift selection scenarios are prepared which can be selected by human experts.

4.6 Improvement

There seems to be a lot of local minimum with strong peak in this search space. To ease to escape from local minimum, we design to keep multiple temporal solutions in one generation. As to each shifts, a few temporal solution can survive based on each cost.

When feasible solution cannot be found for a long time, or generated solution is not good, users can refine working pattern condition and try scheduling again. It is able to fix some of proposed shift when re-scheduling.

5 Experimental Results

We implement prototype system on EWS (HP 7350:146.8 SPEC-mark89) for actual urban subway line to prove the usefulness of proposed framework. This line has 19 stations including 6 handover stations (Figure 3), and 600 trains (total 6,400 run-km) run there on weekday, and minimum headway is about four minutes in the target diagram (weekdays: Figure 4). We also consider actual constraint, evaluation function, working pattern condition and so on. 306 resources is initially generated to distribute 63 shifts (21x3 types).

When we try to generate solutions ten times, all feasible solutions are found at every trial for less than ten minutes, some of them might

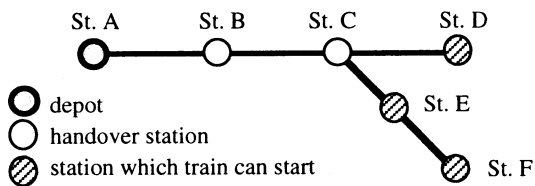


Figure 3: Railway line.

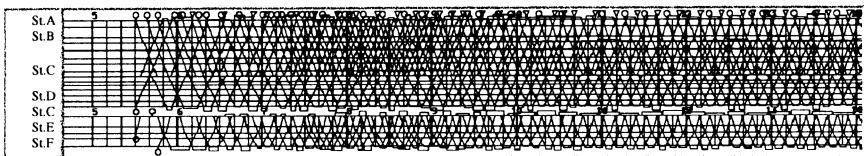


Figure 4: Train timetable.

284 Computers in Railways

have possibility of save one shift. Generation times and Schedule cost are shown as table 1, and a part of best proposed schedule (No.4) is shown as figure 5. The average of value for generated schedule is almost the same as that of actual schedule by human experts.

Table 1: Preparation results.

Trial No.	Generation time(s)	Schedule cost	Saving shift
1	94	1884	night:1
2	378	1684	daytime:1
3	249	1719	daytime:1
4	98	1572	
5	65	2055	daytime:1
6	132	2018	morning:1
7	252	2077	
8	246	2196	daytime:1
9	193	2007	daytime:1
10	362	1608	daytime:1
ave.	207	1882	
actual schedule		1900	

6 Conclusion

In this paper, we have proposed a new framework for crew operation scheduling using simulated evolution technique. This framework has been experimentally observed to give good results for middle scale timetable with actual constraints in a short time. The solving process of this framework resembles that of human experts, therefore it is easy to apply heuristics and expertise. This framework is not guaranteed to get optimum solution, but is available to get adequate schedule to iterate parameter tuning and modification of working pattern condition. We suppose it is suitable to support middle-scale scheduling.

It is unknown whether proposed framework is also effective large-scale scheduling because of explosion of search space. In the future work, we will try large-scale scheduling by adding some extra feature.

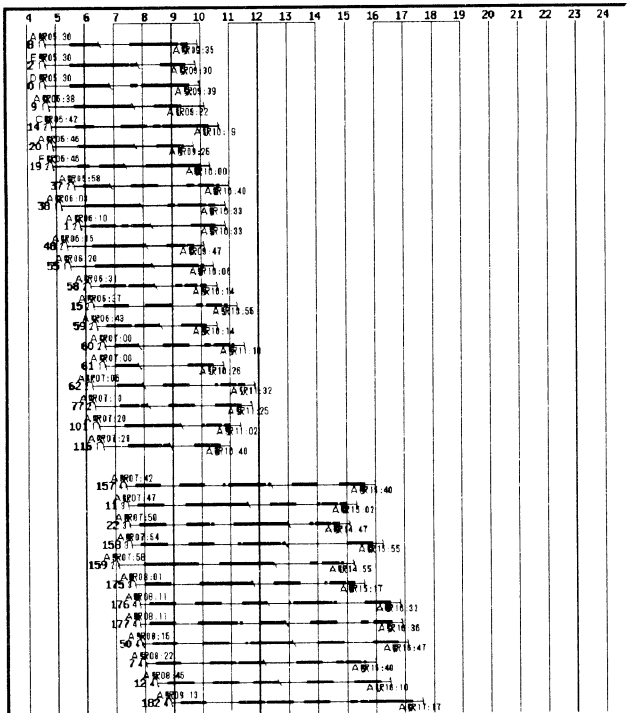


Figure 5: A part of proposed schedule (No.4).

References

- [1] King, R.M.& Banerjee, P., ESP: Placement by Simulated Evolution, *IEEE Trans. Computer-Aided Design*, Vol.8, No.3, pp.245-256 (1989).
- [2] Kataoka, K., Komaya, K., A Model for Railway Crew Allocation Support System, *IEEE Trans. Systems, Man and Cybernetics*, pp.1436-1441 (1992).
- [3] Kataoka, K., Komaya, K., Computer-aided railway scheduling systems for high-density train operations, *COMPRAIL94*, Vol.2, pp.44-50 (1994).