Paper from: *Computers in Railways VIII*, J Allan, RJ Hill, CA Brebbia, G Sciutto and S Sone (Editors).
ISBN 1-85312-913-5

Train schedule diagram drawing algorithm considering interrelationship between labels

H. Izumi[1], N. Tomii[1,2]
[1]*The University of Electro-Communications, Japan.*
[2]*Railway Technical Research Institute, Japan.*

Abstract

Train schedule diagrams, in which various kinds of information about the train operation is illustrated, are by far the most important charts in railways. Although it has become popular to draw them by computers, train schedule diagrams drawn by computers often have defects such as train numbers overlap or train numbers are not so well arranged. We regard the train schedule diagram drawing problem as a sort of edge label placement problem, where a train number is put to the line that denotes the corresponding train as its label. We introduce two types of penalties, namely a static penalty that is calculated from the absolute position of a label and a dynamic penalty that is calculated from the whole arrangement of the labels considering the interrelationship between labels. We have developed a GA based algorithm which searches for an arrangement of labels such that the weighted sum of the static penalty and the dynamic penalty becomes the minimum. We confirmed that the algorithm works effectively through experiments using actual train schedule data.

1 Introduction

Train schedule diagram is a kind of a chart in which information of operational schedules of railway is illustrated. Train schedule diagrams (**diagrams**, in short) are by far the most important charts in railway operation because various kinds of and huge amount of information is stored in a very compact manner. Thus, high visibility is required for the diagrams.

Traditionally, diagrams have been drawn by experienced experts. Recently, it has become popular to draw them by computers, but diagrams

Paper from: *Computers in Railways VIII*, J Allan, RJ Hill, CA Brebbia, G Sciutto and S Sone (Editors).
ISBN 1-85312-913-5

output by computers have defects such as they fail to eliminate overlaps of train numbers or overlaps of train numbers and other symbols. Also, they have another defect such as train numbers are not so well arranged to get a higher visibility. Thus, it cannot be said that we can get diagrams with high visibility by computers.

Recently, as the progress of geometric information systems (GIS), a problem to arrange labels to the symbols on a map is attracting a considerable attention. Especially, a problem to put labels as the names to line-like symbols such as roads and rivers on a map is called an Edge Label Placement Problem (**ELP**), and several research results have been reported [1][2][3][4]. ELP is proved to be an NP-hard problem [4], and attempts to pursue near-optimal solution by heuristic methods have been used.

Diagram Drawing Problem (**DDP**) can be considered to be similar to ELP if we regard train numbers as the labels of lines which correspond to the trains, and seek for the arrangement of the labels which gives the best visibility. On the other hand, in diagrams, it is not enough to avoid the overlap of the labels but their positions have to be adjusted considering the interrelationship among them. For example, there are a couple of requirements with regard to the arrangement of the labels such as; train numbers for the trains of the same type should be put on the same vertical positions and train numbers should not be too congested. These conditions concerning the interrelationship among labels are not considered in the past researches of ELP. Hence, we cannot apply these research results to DDP.

In this paper, we consider DDP as a problem to put labels of train numbers to the lines which depicts trains. We formalize DDP as follows:

1. We generate positions where labels are possibly arranged. These positions are called **Label Positions (LPs)**.
2. We assign penalties to the label positions in which labels are arranged.
3. We introduce two types of penalties. One is a static penalty which is given to each label position which has a label in it, and is calculated from the property of the label position such as its absolute position etc. The other one is a dynamic penalty, which we newly introduce in this paper and is calculated from the interrelationship among labels. Then we seek for the arrangement of the train numbers which minimizes the total of these two penalties by using Genetic algorithm (GA).

In chapter 2, we show the outline of DDP, and we give its formalization in chapter 3. Then in chapter 4, we introduce an algorithm for DDP based on the genetic algorithm together with experimental results conducted using actual train schedule data.

Paper from: *Computers in Railways VIII*, J Allan, RJ Hill, CA Brebbia, G Sciutto and S Sone (Editors).
ISBN 1-85312-913-5

2 Diagram Drawing Problem

2.1 Train schedule diagram

A part of a train schedule diagram is shown in Figure 1. A train schedule diagram (sometimes just called a **diagram**, in this paper) is a chart which illustrates the movements of trains in course of time which is depicted along the horizontal axis. A diagram consists of lines which correspond to the movements of trains (**train lines**), train numbers for each train line, time symbols from which we can know the departure/arrival times of trains at stations by the unit of fifteen seconds, station lines which are drawn horizontally and show positions of the stations, time lines which are drawn vertically and show the times and so on.

Diagrams contain various kinds of and huge amount of information necessary to operate trains in a very compact manner, thus diagrams may well be said to be by far the most important charts in railway operation.

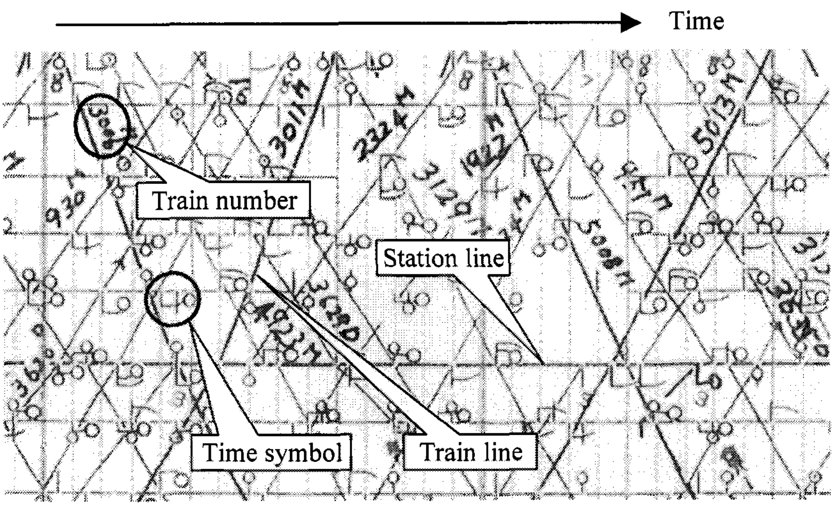

Fig. 1. An example of a train schedule diagram (a part).

Train diagrams are drawn and printed when train schedules are renewed, and distributed to the people who are engaged in train operation business. Although diagrams drawn by computer systems are becoming commonly used, diagrams output by computers sometimes have the defects from the viewpoints of visibility such as it is difficult to recognize train numbers or departure/arrival times of trains because sometimes train numbers overlap each other or train numbers overlap with time symbols. At present, in order to overcome these problems, following means are devised.

Paper from: *Computers in Railways VIII*, J Allan, RJ Hill, CA Brebbia, G Sciutto and S Sone (Editors).
ISBN 1-85312-913-5

- to decrease the probability of the overlap by drawing train numbers in small letters.
- by elaborating the positions of the train numbers (for example to draw train numbers of upward trains and downward trains in completely different places), to diminish the possibility of the overlap.
- to draw all or a part of train numbers by hand.

These means have, however, have problems in visibility and the labor needed to make the diagrams have higher visibility.

Until now, no attempts to treat DDP as a kind of optimization problems have been made.

2.2 Requirements for DDP

Major requirements for DDP algorithms are as follows:

1. Diagrams have to be drawn based on the fundamental rules of diagram drawing as shown in Table 1.
2. Algorithms have to be generally applicable to the diagrams in various kinds of railway lines that might differ in density of trains and the distance between stations and so on.
3. Algorithms are required to produce diagrams with high visibility.

Table 1. Fundamental rules of diagrams drawing.

1	Train numbers have to be put in the right hand side for upward trains and in the left hand side for downward trains.
2	Train numbers have to be put in positions so that the correspondence between the train numbers and the lines is clearly recognized.
3	At least one train number has to be put for a train line for a section that is prescribed as an interval between two station lines (we call this interval a **prescribed section**).
4	Time symbols have to be drawn on the lower side of the station lines for upward trains and on the upper side for downward trains.

Some of the examples of the third requirement are:

- Train numbers must not overlap each other.
- Train numbers and time symbols must not overlap each other.
- Train numbers should be put in a position that is easy to identify such as just in the middle of two station lines etc.
- Train lines should not cross with train numbers if possible.
- Train numbers and station lines should not intersect if possible.
- Vertical positions of train numbers for the trains of the same type should be the same if possible.

Paper from: *Computers in Railways VIII*, J Allan, RJ Hill, CA Brebbia, G Sciutto and S Sone (Editors).
ISBN 1-85312-913-5

- Train numbers for different types of trains should not be arranged too densely.

3 Formalization of DDP

3.1 Basic ideas

We regard DDP as the following problem:

- We consider a diagram as a chart which consists of lines, train numbers, station lines and time symbols.
- We treat train numbers as slanted rectangles and time symbols as upright rectangles.
- Positions of lines, time symbols and station lines are assumed to be fixed.
- We consider train numbers as labels for the train lines, and treat DDP as a problem to find the arrangement of the labels which minimizes a certain measure.

Then, we introduce an algorithm to draw a diagram based on the following ideas.

- First, we prepare positions for each train line where its labels are possibly set. These positions are called **label positions** (**LP**s).
- We then calculate a penalty for each LP. The penalty is called a static penalty which is calculated from several factors such as the absolute position of the LP, whether the LP intersects with station lines or another train lines etc.
- We calculate a penalty for the whole arrangement of the labels, which is called a dynamic penalty.
- We try to find an arrangement of the labels which minimizes the weighted sum of these two penalties by using the genetic algorithm.

3.2 Generation of label positions

Label positions (LPs) are slanted rectangular regions where labels might be possibly put. We equally generate k LPs in the prescribed section. Here, k is called **LP generation number** and we assume the value of k is given beforehand considering the complicatedness of the diagram. The size of the rectangle is decided from the length of the train numbers and the size of the fonts used. In Figure 2, we give an example of generation of LPs when $k = 5$.

3.3 Static penalty

We introduce the following four penalties as the static penalty of a label position.

[SP1] A penalty imposed by the absolute position of the LP.

Paper from: *Computers in Railways VIII*, J Allan, RJ Hill, CA Brebbia, G Sciutto and S Sone (Editors).
ISBN 1-85312-913-5

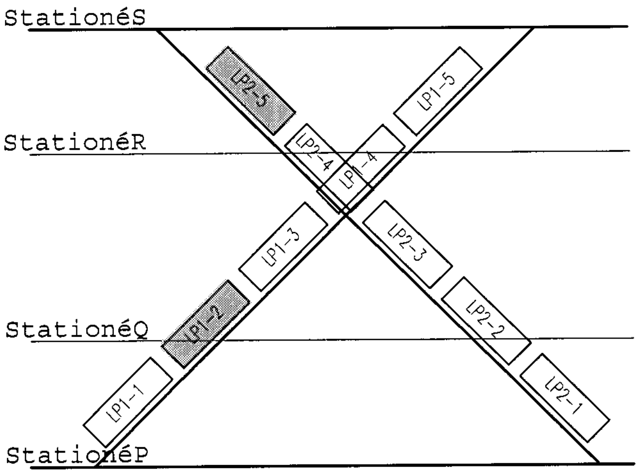

Fig. 2. Generation of Label positions.

[SP2] A penalty imposed by the intersection of the LP and station lines.

[SP3] A penalty imposed by the overlap of the LP and time symbols.

[SP4] A penalty imposed by the intersection of the LP and train lines.

SP1 is introduced so that the difference of visibility affected by the absolute positions of train numbers should be reflected to the evaluation measure. In fact, it is most favorable to put a train number in the middle position of the prescribed section, if we can ignore other conditions. If this is not desirable for some reason, then the second best position is near the starting end of the train line. In case of Figure2, the values of SP1 for LP1s are set so that they gradually increase from LP1-3, LP1-1, LP1-2, LP1-4, LP1-5.

SP2 is a penalty introduced to avoid the intersection of LPs and station lines. In case of Figure 2, LP1-2, LP1-4, LP2-2 and LP2-4 intersect with station lines, thus certain amount of SP2 is given to these LPs.

SP3 is introduced to prevent LPs from overlapping with time symbols. SP4 is a penalty introduced to avoid the intersection of LPs and train lines. In Figure 2, LP1-4 and LP2-4 overlap with train lines, thus certain amount of penalties are added to these LPs.

The overall static penalty for an arrangement of labels is calculated as the total sum of the static penalty of LPs on which labels are put.

3.4 Dynamic penalty

The dynamic penalty is calculated to the whole arrangement of the labels. We introduce the following three penalties as the dynamic penalties.

[DP1] A penalty caused by the overlap of the labels.

Paper from: *Computers in Railways VIII*, J Allan, RJ Hill, CA Brebbia, G Sciutto and S Sone (Editors).
ISBN 1-85312-913-5

[DP2] A penalty imposed by the irregularity of the vertical positions of labels.

[DP3] A penalty imposed by the density of labels.

DP1 is a penalty to avoid overlaps of train numbers, and calculated as the number of overlapping labels. DP2 is a penalty introduced so that train numbers for the same type trains should be aligned in a line, and is calculated as the variance of the number of the labels arranged in one horizontal row by the following formula. In this formula, *ave* is calculated as *(the total number of LPs in the prescribed section) / (LP generation number for the prescribed section)*, and $LPnum_i$ is the number of the labels in the i-th horizontal row.

$$\sqrt{(LPnum_0 - ave)^2 + \cdots + (LPnum_n - ave)^2}$$

We calculate DP3 as the number of the labels which exist within a certain distance from one label. Summing up these numbers for each label, we get DP3.

3.5 Total penalty

The total penalty for an arrangement of labels is calculated as the weighted sum of the static and the dynamic penalties, which is expressed by the following formula where P^S and P^D are the static and dynamic penalties and a and b are the weights for them.

$$P = a \cdot P^S + b \cdot P^D$$

4 Diagram Drawing Algorithm considering interrelationship among labels

4.1 Overall construction of the algorithm

We show the overall construction of the algorithm in Figure 3. Since DDP is a complicated optimization problem, we use the genetic algorithm (GA), which is believed to be useful to solve such problems.

4.2 Chromosomes

An individual has information of all the positions of the labels. This information is held in the form of the sequence of the sequential numbers of LPs on which labels are put. Let us assume that in Figure 3, labels are set on LP1-2 for the train line 1 and on LP2-5 for the train line 2 respectively. The chromosome expression for this arrangement of labels thus becomes 25. This way of expression has a merit that the length of chromosomes does not become too long compared with the method to express the existence of the label in each LP by a bit string.

Paper from: *Computers in Railways VIII*, J Allan, RJ Hill, CA Brebbia, G Sciutto and S Sone (Editors).
ISBN 1-85312-913-5

4.3 Generation of the initial population

We generate an arrangement of the labels at random, and repeat this process until required number of individuals for a population is obtained.

4.4 Fitness value of individuals

The fitness value of each individual is calculated by applying the linear scaling [5] to the value calculated by the formula we presented in section 3.5.

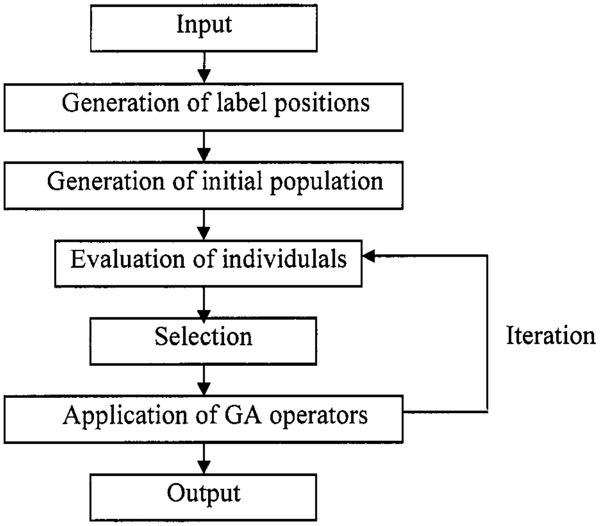

Fig. 3. Overall construction of the algorithm.

4.5 Selection

As the selection method, we use the roulette wheel approach [5], namely individuals are selected depending on the probability which is calculated from the fitness values of individuals.

4.6 GA operators

We employ the crossover and the mutation as the GA operators. The crossover operator produces a new individual from two individuals (parents) by exchanging a part of their chromosomes. The mutation operator modifies a part of the chromosome based on a probability specified as the mutation ratio.

Paper from: *Computers in Railways VIII*, J Allan, RJ Hill, CA Brebbia, G Sciutto and S Sone (Editors).
ISBN 1-85312-913-5

5 Results of Experiments

We have applied our algorithm to the data extracted from actual train schedule. We show a result of the experiments in Figure 4. There are no overlap of train numbers nor no overlap of train numbers and time symbols, and train numbers are well arranged so that intersection with train lines and other symbols is avoided. The data concerning the experiments are shown in Table 2. Also, processes of the convergence are depicted in Figure 5. In Figure 5, the processes for two trials among the ten trials we performed are shown, one we got the result most quickly, and the other is the trial we needed the longest time. In the experiments, we set the LP generation number as 15. Execution time was about two minutes for each experiment (Mobile Pentium . 600 MHz, 192MB).

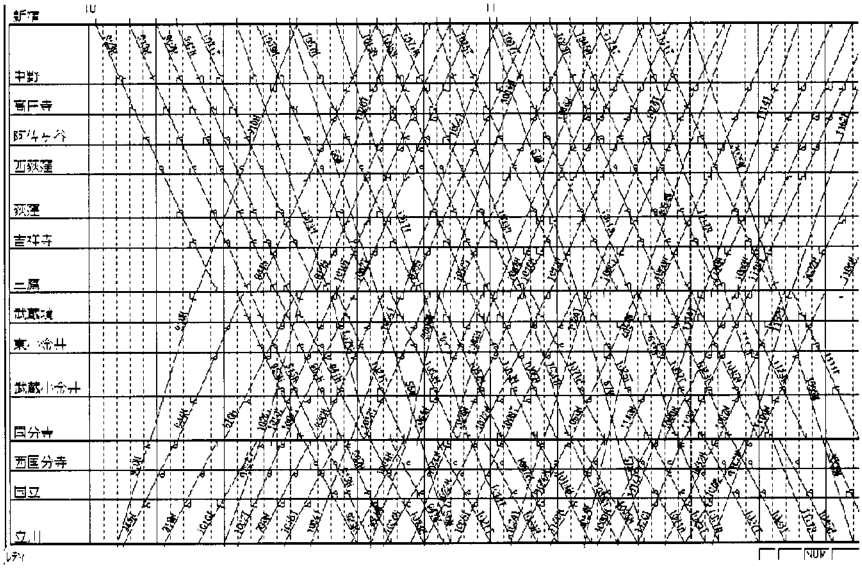

Fig. 4. An example of experimental results.

Table 2. Data concerning the experiments.

Population size	20	Performed generations	3000
Crossover ratio	0.9	Mutation ratio	0.01
Number of trains involved	49	Number of stations	15
Number of labels arranged	147	Number of train types	3

Paper from: *Computers in Railways VIII*, J Allan, RJ Hill, CA Brebbia, G Sciutto and S Sone (Editors).
ISBN 1-85312-913-5

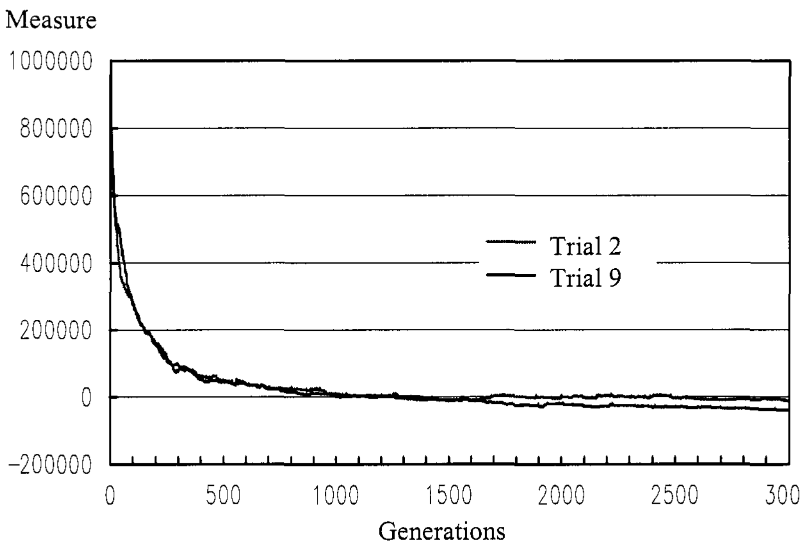

Fig. 5. Process of Convergence.

6 Conclusion

We have introduced an algorithm considering the interrelationship between labels, which can produce a train schedule diagram with high visibility. Then, we have confirmed its effectiveness through several experiments using actual train schedule data.

References

[1] K.G.Kakoulis and I.G.Tollis: On The Edge Label Placement Problem, *Proc. Graph Drawing '96*, LNCS 1190, 1997.

[2] K.G.Kakoulis and I.G.Tollis: An Algorithm for Labeling Edges of Hierarchical Drawings,. *Proc. Graph Drawing '97*, LNCS 1353, 1998.

[3] K. Imai et al: Map Labeling Problems, *3rd Symposium on Integrated Geometric Information Systems*, 1999.

[4] K.G.Kakoulis and I.G.Tollis: A Unified Approach to Labeling Graphical Features. *Proc. 14th ACM Symp. On Computational Geometry*, 1998.

[5] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, *Addison Wesley*, 1989.