Optimal train control at a junction in the main line rail network using a new object-oriented signalling system model

R. Takagi[1], P. F. Weston[2], C. J. Goodman[2], C. Bouch[3], J. Armstrong[4], J. Preston[4] & S. Sone[1]
[1]*Department of Electrical Systems Engineering, Kogakuin University, Japan*
[2]*Rail Research UK, School of Engineering University of Birmingham, UK*
[3]*Rail Research UK, School of Engineering (Civil Engineering), University of Birmingham, UK*
[4]*Rail Research UK, Transportation Research Group, School of Civil Engineering and the Environment, University of Southampton, UK*

Abstract

On a main line railway network with many junctions, the delay of a train is likely to cause delays to many other trains, especially because of conflicts at junctions. Optimising one junction, however, may have an adverse effect on other parts of the rail network because of the mixed-traffic situation of most main line railways. To approach the complicated problem of optimal re-scheduling in response to the delay of a train, an efficient algorithm must be sought.

The authors have taken a junction as an example, and have performed numerical optimisation on a case when the services through this junction are disrupted. The objective criterion is the weighted sum of train times. The optimisation program uses the Object-Oriented Multi-Train Simulator (OOMTS) developed by Birmingham University, as an embedded simulator. In the optimisation routine, a Genetic Algorithm (GA) was used to optimise the order of route setting.

In this paper, the authors give details of a model junction, and a brief explanation of the OOMTS. The authors then explain how a GA can be applied to solve this problem, especially the chromosomal expression of the problem. The results of numerical optimisations for different weighting parameters are shown based on which the authors discuss the feasibility of the proposed method.
Keywords: object-oriented, multi-train simulator, railway signalling, junction, train pathing, optimisation, Genetic Algorithm.

WIT Transactions on The Built Environment, Vol 88, © 2006 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/CR060481

1 Introduction

On a main line railway network with many junctions, the delay of a train is likely to cause delays to many other trains, especially because of conflicts at junctions. Junctions without flyovers, as are commonly seen in the British rail network, tend to make matters worse. Also, most main line railway lines run mixed traffic with a combination of fast and slow trains, making the junction management even harder. Therefore, controlling a junction in response to a disruption by minimising the accumulated delay over a sequence of trains at that point may still have serious adverse effects elsewhere in the rail network. Thus, the problem of optimal re-scheduling in response to a delay of a train is computationally a difficult task, and an efficient algorithm must be sought to approach this problem.

In an attempt to address this problem, the authors have taken a junction as an example, and have performed numerical optimisation on a case when the services through this junction are disrupted. The junction that has been used for the study is Abbotswood Junction on the Birmingham to Bristol Line in the National Rail network of the UK. The optimisation program uses the Object-Oriented Multi-Train Simulator (OOMTS) developed by Birmingham University [1], as an embedded simulator. In the optimisation routine, a Genetic Algorithm (GA) is used to optimise the order of route setting.

2 OOMTS: an outline

The first version of the Object-Oriented Multi-Train Simulator (OOMTS) at the University of Birmingham was written by Siu [2] in the early 1990s. The logic of the program is based on a previous Fortran version [3], under constant development since 1973 and used by many rail operators and manufacturers around the world.

The model implemented in OOMTS comprises of five “subsystem managers” [4], each containing data (or the instances of objects necessary for running the simulation) for a railway “subsystem”. The Simulation Manager is at the top of the structure, managing all these subsystem managers and the actual running of the simulation itself.

The following are brief explanations of the five subsystem managers:

1. Network Manager: This subsystem manager holds instances of objects containing data for the topography and track profile (gradient, curve radius and line speed) of the rail network to be simulated.
2. Signal Manager: This subsystem manager holds instances of objects containing data for the signalling system.
3. Rolling Stock Manager: This subsystem manager holds data for the types of rolling stock to be used in the simulation.
4. Power Network Manager: This subsystem manager holds instances of objects for the electric power supply system.
5. Train Service Regulator: This subsystem manager holds instances of objects containing data for various aspects of train services, including timetable information.

OOMTS was originally designed and tested for metro lines and suburban lines, for example the Island Line of Hong Kong's MTR Corporation, and has successfully been used by researchers at the University of Birmingham, latterly within Rail Research UK, the universities' centre for railway systems research [5].

3 The model used in the optimisation

Abbotswood Junction, on the Birmingham to Bristol Line in the UK, is used as the model on which the optimisation was performed.

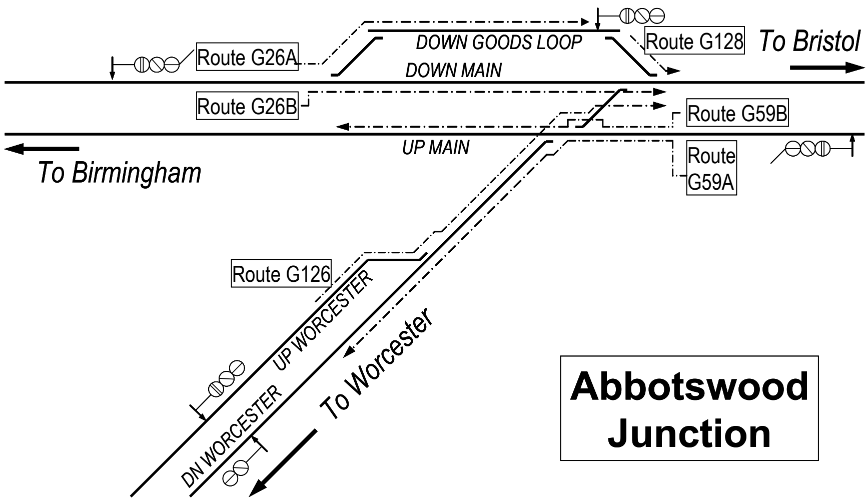

Figure 1: Abbotswood Junction with six possible routes.

Figure 1: shows the schematic of the junction. The Birmingham to Bristol Line is double track, and a single track branch line from Worcester joins here. Another feature of this junction is the Down Goods Loop, which allows passenger trains from Birmingham to Bristol Line to overtake a freight train. There are six possible routes for trains, and no reversing of trains is allowed.

The junction is modelled using the actual detailed data of the signalling system that is currently in use. This detailed data also covers the nearby part of the network, namely:

1) about five kilometres from Abbotswood Junction towards Birmingham,
2) about five kilometres from Abbotswood Junction towards Bristol, and
3) about 500 metres from Abbotswood Junction towards Worcester.

For the optimisation study, ten kilometres of double-track have been added to both the Birmingham and the Bristol end of the Birmingham to Bristol Line track data, and one kilometre of double-track has been added to the Worcester end of the branch line data. Virtual stations are defined at the far ends of each of these tracks, and trains are assumed to travel between these stations.

Figure 2: shows the assumed schedule of trains run through Abbotswood Junction. The trains involved are: one freight train on the Down Main track from Birmingham to Bristol; three passenger trains each for both Down and Up Main lines between Birmingham and Bristol; and three passenger trains each for both Down and Up Worcester tracks between Worcester and Bristol.

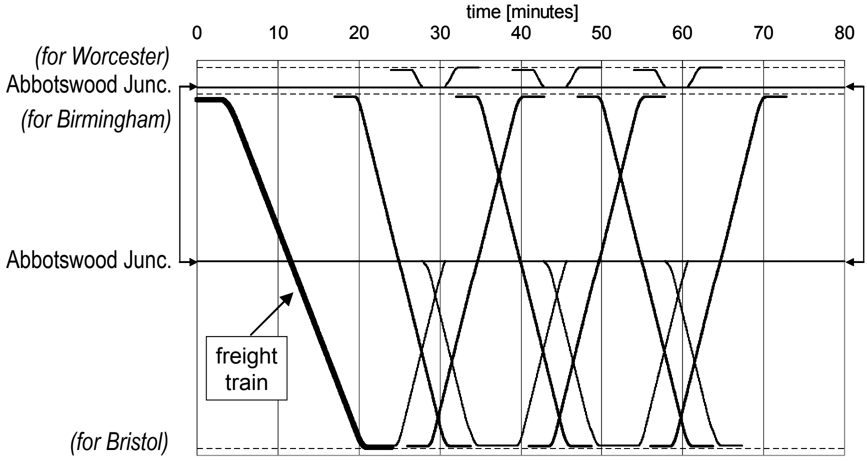

Figure 2: Assumed schedule of trains including one freight and twelve passenger trains.

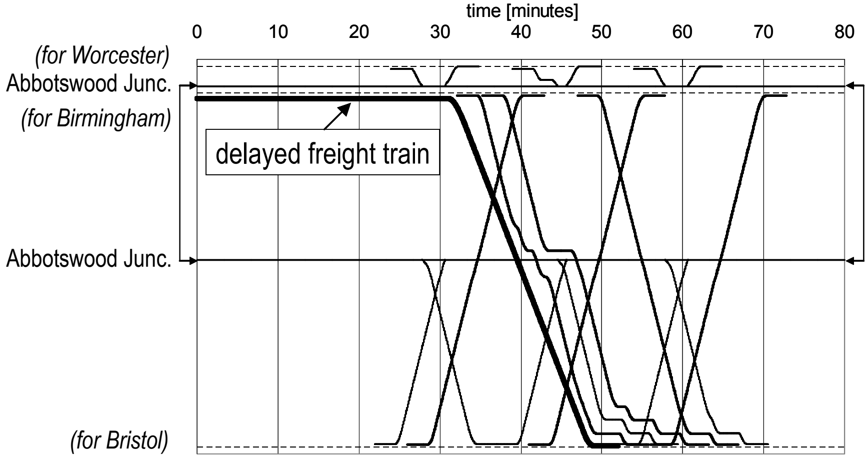

Figure 3: Result of applying “first-come, first-served” junction control with a 30-minute delayed freight train.

4 Assumed incident and the response by the non-optimal controller

The authors have assumed that the departure of the freight train is delayed for 30 minutes, causing the following Birmingham to Bristol passenger services to suffer reactionary delays.

Figure 3: shows the result of applying “first-come, first-served” type control, which is by no means optimal in this situation. Because the freight train is not scheduled to use the Down Goods Loop at Abbotswood Junction, the slow freight train causes more delay to the subsequent passenger trains at their destinations. There is also room for improvement by re-ordering passenger trains.

It is obvious that, provided the passenger trains have at least as high a priority as the freight train, the optimal solution must include the use of the Down Goods Loop by the freight train, which will enable one or more passenger trains to overtake the freight train. Then the decision must be made as to which of the following passenger trains should overtake the freight train.

5 Application of Genetic Algorithm

The authors have applied a Genetic Algorithm (GA) [6, 7] to identify the optimal control strategy for this situation.

To apply a GA to any optimal control problem, one has to express the control inputs as a “chromosome”, for which an evaluation can be given. For the most basic variation of GA, known as Simple GA (SGA), a chromosome must be a fixed-length binary sequence. Also, one has to define genetic operators; in the case of SGA they are mutations, crossovers and selections.

In the problem of train control at junctions, the control inputs are the setting of routes. The fitness of a particular routing sequence will be based on the time intervals between the scheduled departure times and the simulated arrival times, with weighting according to train priority. Here, it is assumed that the departure times and the performance of trains are given, which means that a set of control inputs can be defined as the order in which routes are set. One possible chromosomal expression, therefore, is the queue of routes; one example in the case of Figure 1: is (G26A, G26B, G128, G126, G59B …).

However, using the queue of routes as the chromosomal expression has disadvantages. In the case of Figure 1:, if the head of the queue is G26A and the first train to arrive on the Down Main track from Birmingham is a passenger train (not allowed to go into the goods loop), then the system comes to a deadlock situation. Also, if there is already a train in the goods loop and the head of the queue is G26A, then the system comes to a deadlock situation because two or more trains are not allowed to be in the goods loop at the same time.

To avoid these disadvantages, the authors propose the use of a state-space trajectory as the chromosomal expression. In terms of control of railway junctions, the state of the rail network can be expressed as the queue of trains on any of the tracks that connect junctions. In the model used in this paper, the order in which trains arrive at Abbotswood Junction is fixed, and the state expression can be simplified to a combination of four integers, as shown in Figure 4:. Using this state expression, a chromosome can be defined as a sequence of system states, starting from the initial state, $(A, B, C, D) = (4, 3, 6, 0)$, and ending at the final state, $(A, B, C, D) = (0, 0, 0, 0)$. A change from one state to another indicates that a route is set. For example, in Figure 1:, the change of state from (4, 3, 6, 0) to (3, 3, 6, 1) means that the route G26A is set. Taking constraints into consideration is easier; for example, the constraint that two or more trains cannot be in the Down Goods Loop at any one time can be easily expressed as $D \leq 1$.

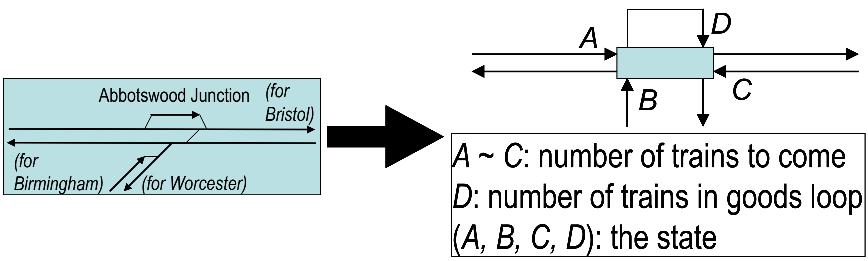

Figure 4: The state of the network.

A mutation operation can be defined as a partial change in the trajectory. Upon mutation the constraints can be taken into account so that the chromosome after the mutation is still a “feasible” solution. The probability of the occurrence of a mutation is to be higher when the mutation transforms an original chromosome to another one that has a smaller Levenshtein distance [8] to the original.

A crossover operation between two chromosomes is possible only when there are one or more common states (other than the initial and the final states) in the chromosomes. If there are two chromosomes P (sequence $S_{P,\,1}, S_{P,\,2}, \ldots, S_{P,\,N}$) and Q (sequence $S_{Q,\,1}, S_{Q,\,2}, \ldots, S_{Q,\,M}$), and if $S_{P,\,i} = S_{Q,\,j}$, then the crossover of these two chromosomes at $S_{P,\,i} = S_{Q,\,j}$ creates a new chromosome which has the sequence of either $(S_{P,\,1}, S_{P,\,2}, \ldots, S_{P,\,i}, S_{Q,\,j+1}, S_{Q,\,j+2}, \ldots, S_{Q,\,M})$ or $(S_{Q,\,1}, S_{Q,\,2}, \ldots, S_{Q,\,j}, S_{P,\,i+1}, S_{P,\,i+2}, \ldots, S_{P,\,N})$. If no common state is found in the sequences of P and Q except for the initial and final states, then the pair of chromosomes P and Q are called “infertile”.

Using this chromosomal expression, the problem can be solved in broadly the same way as the SGA.

6 Optimisation results and discussion

The authors have performed numerical optimisation on five different cases, using GA with the proposed chromosomal expression as explained in the previous section.

For all cases, a weighted sum of train times is used to evaluate a chromosome. The only difference between cases is the values of the weighting coefficients for passenger and freight trains. All other conditions are unchanged, remaining as described in the earlier sections.

Also, the parameters for the GA are common for all cases. There are 60 chromosomes per generation, and the algorithm terminates when 100 generations have been calculated. Given the population and evaluation of the current generation, the next generation of populations is determined as follows.

1) If the best evaluation among the current generation is worse than the very best which existed in the past, add the very best chromosome to the current generation population.
2) Create a group of 18 chromosomes by crossovers. One chromosome is created by the crossover between two parent chromosomes selected from the current generation, and is added to the new group. A chromosome in the current generation with better evaluation is assigned a higher possibility of being selected as one of the parents. If the selected pair is “infertile” and cannot create a crossover “child”, then mutation is applied to one of the pair and added to the new group instead.
3) Create the group of 42 chromosomes by selecting from the current generation of population, allowing the selection of the chromosome twice or more. The possibility that a chromosome in the current generation is selected is the same as the function used in 2).
4) Add the groups of chromosomes generated by 2) and 3) to create a group of 60 chromosomes.
5) Apply mutation to 80 per cent of the chromosomes in the group created in 4).

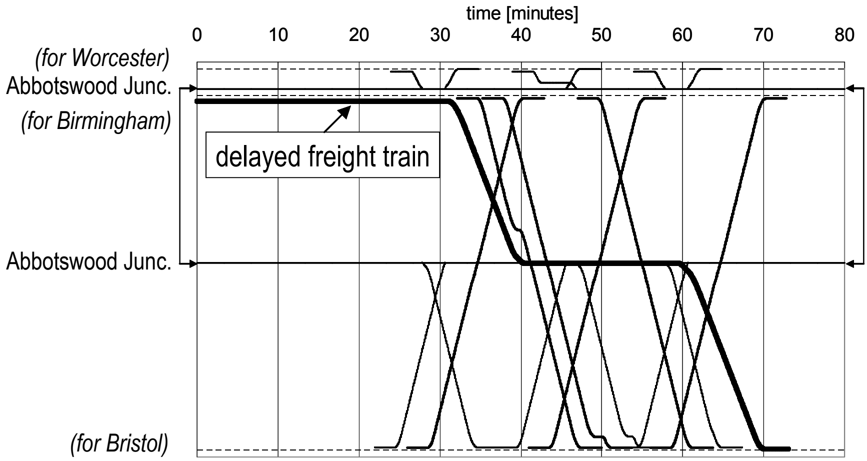

Figure 5: Optimisation result (1) weight of passenger : freight trains = 6 : 1.

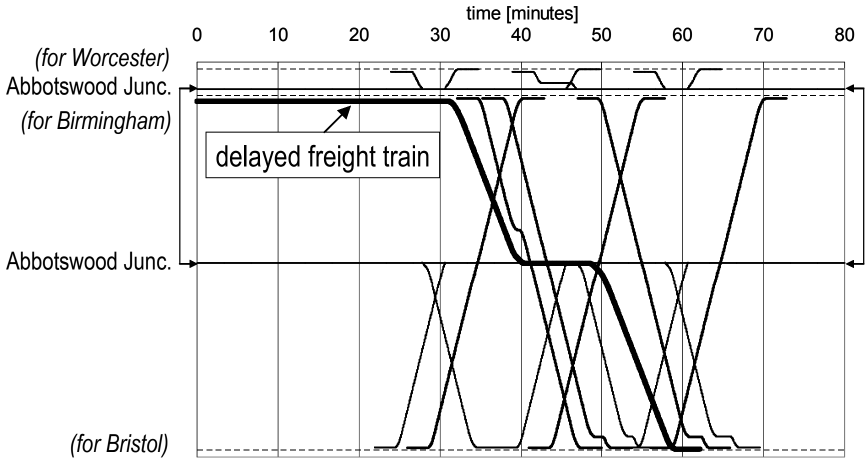

Figure 6: Optimisation result (2) weight of passenger : freight trains = 6 : 3 or 6 : 6.

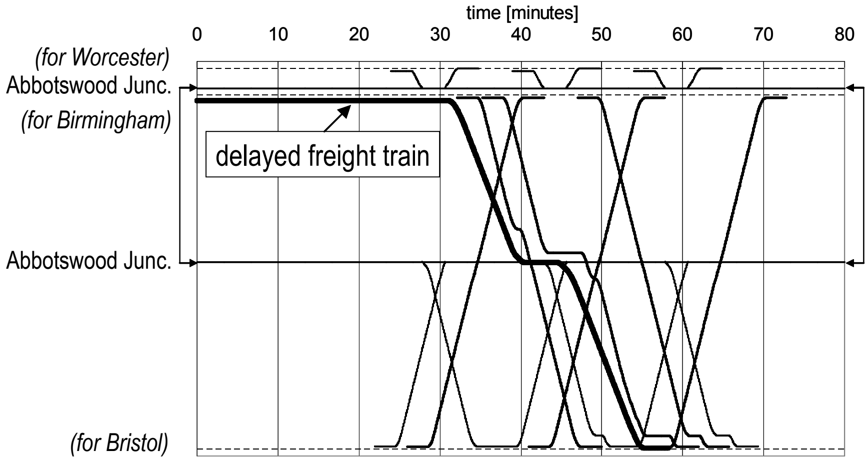

Figure 7: Optimisation result (3) weight of passenger : freight trains = 6 : 12.

Figure 5: shows the case where the weightings of passenger : freight trains is 6 : 1, in which the freight train departs the Down Goods Loop after all passenger trains bound for Bristol have made their way through Abbotswood Junction. Cases for the weightings of 6 : 3 and 6 : 6 yield the same result, shown in Figure 6:, in which the freight train leaves the last two Bristol-bound passenger trains behind. Results for the 6 : 12 and 6 : 24 weightings (shown in Figure 7: and Figure 8:, respectively) show that, as the weight for the freight train increases, fewer passenger trains overtake the freight train at Abbotswood, with Figure 8: showing that no overtaking becomes the optimal solution. These results suggest

that a change in the order of passenger trains and the freight train takes place when, as the weightings given for the freight train becomes larger, the disbenefit of holding the freight train is larger than the accumulated benefit of letting the passenger trains overtake at Abbotswood Junction.

The calculation time required for the optimisation was nearly 5 hours using a personal computer with 1.4GHz Intel Pentium 4 CPU; although this is clearly too long to be practical, considering that a detailed dynamic multi-train simulator has been used embedded in the optimisation process, it does not necessarily mean the GA cannot be used for this kind of optimisation.

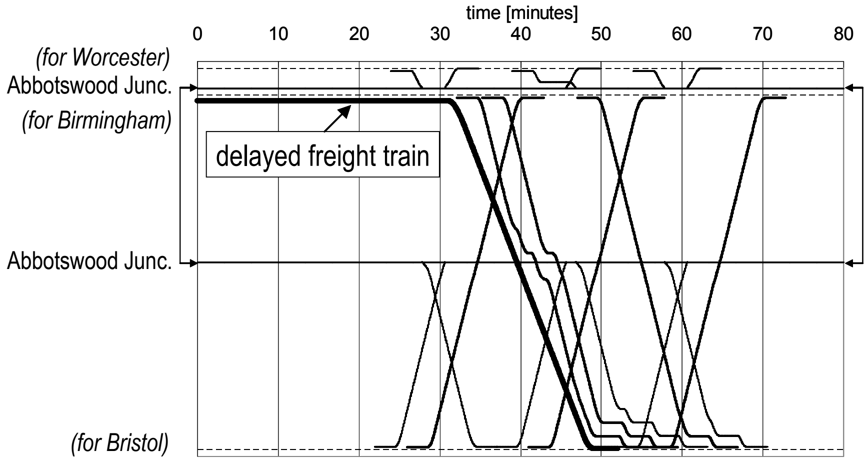

Figure 8: Optimisation result (4) weight of passenger : freight trains = 6 : 24.

The different results presented in this section are generated only by changing the weighting coefficients. However, there are many parameters that have to be determined, such as the size of the population per generation, percentage of chromosomes to be created through crossovers or percentage of chromosomes to mutate. These parameters may depend on track topography, the train schedule and the disruption scenario.

7 Conclusion

A number of numerical optimisations of a railway junction control have been carried out using Genetic Algorithms and using OOMTS as an embedded simulator. A novel chromosomal expression for the problem has been proposed that enables the problem to be solved in a way very similar to that used by the Simple GA. The authors are looking at the application of the same idea to various other problems, especially railway networks with multiple junctions.

Acknowledgement

This research was conducted within RRUK, the Universities' Centre for Railway Systems Research, funded by the EPSRC, U.K. (GR/S12784/01).

References

[1] Takagi, R., *et al.*: "An Object-oriented Signalling System Model for the Study of Delay Minimization by On-line Train Control", *IRSE Aspect 2006*, London, UK (March 2006).

[2] Siu, L. K.: "An Object-oriented Railway System and Power Network Simulator", The University of Birmingham (PhD Thesis) (1995).

[3] Mellitt, B., Goodman, C. J. and Arthurton, R. I. M.: "Simulator for studying operational and power-supply conditions in rapid transit railways", *Proc. IEE*, Vol. 125, No. 4, pp.298-303 (1978).

[4] Siu, L. K. and Goodman, C. J.: "An object-oriented concept for simulation", *Proc. COMPRAIL 1992 International Conference*, Washington DC, USA (1992).

[5] Goodman, C. J.: "Promoting railway research in United Kingdom universities: the genesis of Rail Research UK", *Proc. of a Symposium on Research and Development in Railway Engineering*, Dept. of Elec. Eng., Hong Kong Polytechnic University, 3.1.1 - 3.1.12 (2003).

[6] Nagao, T.: "Optimization Algorithms" (in Japanese), Shoko-do Publishing, Tokyo, Japan (2000), Chapters 10 – 12.

[7] Rothlauf, F.: "Representations for Genetic and Evolutionary Algorithms", Physica-Verlag, Heidelberg, Germany (2002).

[8] National Institute of Standards and Technology (NIST): "Dictionary of Algorithms and Data Structures" (http://www.nist.gov/dads/HTML/Levenshtein.html).