

An Intelligent Search Technique for Solving Train Scheduling Problems: Simulated Annealing and Constraint Satisfaction

M.T. Isaai¹

This paper presents a hybrid scheduling technique for generating the predictive schedules of passenger trains. The algorithm, which represents a combination of simulated annealing and a constraint-based heuristic, has been designed using an object-oriented methodology and is suitable for a primarily single-track railway with some double-track sections. The search process gets started from a good initial solution created by the scheduling heuristic and continues, according to the simulated annealing search control strategy. The heuristic is also used in the neighborhood exploration process. This hybrid approach solves the problem in a short span of time. Simulation experiments, with the real data of manual timetables and two corridors of Iran's railway, show the superiority of the hybrid method to the heuristic designed and the manual system, in terms of the three performance measures used.

INTRODUCTION

The planning of train schedules, which is a crucial task in rail operations planning, is an optimization problem and its complexity has been addressed in various publications. For a single-line network, one solution methodology is to resolve the set of existing conflicts on allocating track sections to the competing trains. It can be shown that the size of solution space grows exponentially as the number of conflicts increase [1]. To resolve a conflict which involves two trains, either of the two may get precedence. Accordingly, for a problem including n conflicts, the number of solution possibilities is 2^n .

Traditionally, train scheduling experts manually construct the timetables using a distance-time graph, which shows the progress of any train toward its destination with respect to time. It also represents the potential train conflicts and possible decisions. To facilitate the scheduling process and to increase financial returns from huge investments made in the rail industry, various classes of the problem have been investigated and a number of practical decision support systems have been developed and used in real environments. The goal of such systems would be to create appropriate timetables, investigate the robustness of

schedules in the presence of unforeseen events and to analyze the capacity of the network with reference to a given transport demand. The simulation approach appears to have caught the attention of researchers and service-planning professionals extensively and has been embraced relatively well by the industry. A few simulation tools have been developed and used in practice. For example, VISION, SIMON, SIMU-PLAN and CAPTURE/PROVING [2-5] have been used for timetabling, TTS and LCAS [6,7] for capacity analysis and Dispatch Analysis Model [8] for both purposes. A general modeling framework has been proposed in [9] for use in simulating rail operations. The complexity of the problem and the possibility of simulating detailed aspects of the problem have contributed to the invention and use of train dispatching rules.

The train scheduling problems have also been formulated using analytical or OR-based techniques, including (non-) linear (mixed) integer programming [10-12] and network optimization models [13,14]. Amongst the solution techniques developed to solve the problem are branch & bound [1], heuristics [15], Lagrangian relaxation [12] and the shortest path method [11, 13]. OR-based solution techniques have provided valuable insights into the problems. However, few models have solved 'real-life' problems [9] and succeeded in practice. The reasons could be in the simplified assumptions, the long computing time required and the difficulties in including the domain's knowledge. Promising

1. Graduate School of Management and Economics, Sharif University of Technology, Tehran, I.R. Iran.

works in the field of artificial intelligence encourage further research in the field to solve the problem more efficiently. Examples are the works of [16,17] in the field of knowledge-based systems and [18] in constraint satisfaction methods. Additionally, AI-based search techniques have been developed by [19-21] and the combinations of knowledge-based techniques with simulation and OR-based methods are presented in [22,23].

This paper presents a hybrid train scheduling algorithm, HeuSA, which is a combination of a simulated annealing control strategy and the constraint-based heuristic introduced in [20]. The hybrid model is tested using the data of manually created timetables and the rail network of Iran. The goal is to find appropriate solutions in a short span of time through starting the search process from a good solution and reducing the likelihood of being trapped in local optima. In addition, new performance measures can easily be used with the model.

In the remainder of the paper, first, the problem at hand is represented and brief discussions on simulated annealing and constraint propagation is provided. Then, insights into the hybrid algorithm developed are provided and simulation results are presented. Finally, concluding remarks are provided.

PROBLEM REPRESENTATION

The problem consists of generating a predictive schedule for a set of passenger trains traveling on a primarily single-line railway that includes some double-line sections, using Iran's rail network and train data. For any train being considered, the following data is specified by experts: Ready time for boarding at the origin station, minimum stop time considered for reasons such as boarding and alighting at the origin and any intermediate station and point-to-point running time on the track between two adjacent stations, called a block. Data of train operations in the past can be used to work out the above train attributes. Decision

variables associated with a train are departure times at the origin and en route stations and arrival times at the destination and en route stations. Thus, stop time at each station is taken as a dependent variable with a value equal to, or greater than, a minimum time planned. The problem constraints are, as follows:

1. For each train, t , traveling between two adjacent stations, namely from s to $s + 1$:

$$\text{Arrival}_{t,s+1} - \text{Departure}_{t,s} = \text{Running time}_{s,s+1}. \quad (1)$$

2. For each train, t , at each station, s , en route:

$$\text{Departure}_{t,s} - \text{Arrival}_{t,s} \geq \text{Planned stop time}_{t,s}. \quad (2)$$

Planned stop time is a constant greater than, or equal to, zero.

3. For any pair of trains, $t1$ and $t2$, traveling in the same direction, either on a single or double-line track (Figure 1: a, b, c) from station s to its immediate adjacent station, $s + 1$:

$$\text{Departure}_{t1,s} - \text{Arrival}_{t2,s+1} \geq 0,$$

or:

$$\text{Departure}_{t2,s} - \text{Arrival}_{t1,s+1} \geq 0. \quad (3)$$

For any pair of opposite trains, $t1$ and $t2$, traveling on a single-line track between immediate adjacent stations, s and $s + 1$, if $t1$ is traveling from s to $s + 1$ (and $t2$ from $s + 1$ to s):

$$\text{Departure}_{t1,s} - \text{Arrival}_{t2,s} \geq \text{Safety interval},$$

or:

$$\text{Departure}_{t2,s+1} - \text{Arrival}_{t1,s+1} \geq \text{Safety interval}, \quad (4)$$

where the safety interval is the time interval between consecutive occupation of a single-line track by opposite trains (in this model, safety interval = 1 minute) (Figure 1: d, e and f).

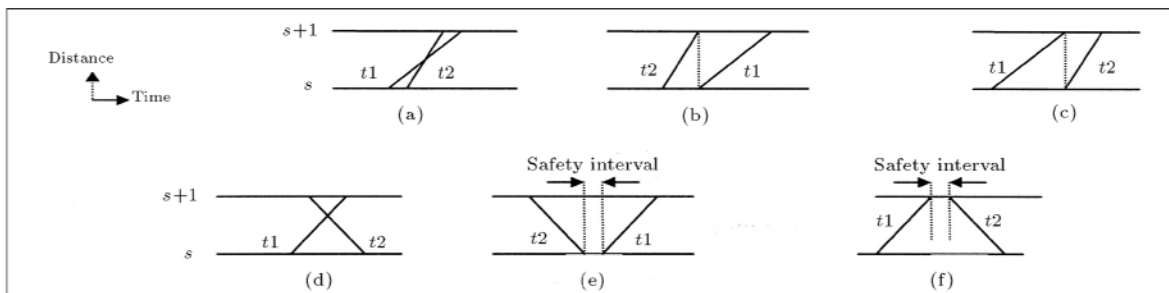


Figure 1. A distance-time graph for two trains traveling in the same direction in three situations: (a) A resource conflict exists; (b and c) The conflict is resolved using one of the above constraints; (d, e and f) Similar graphs have been drawn for opposite trains. In this type of graph, consecutive departure and arrival times of each train are connected by a line to represent train movements and stops.

4. For each train, t , at its origin station, o :

$$\text{Departure}_{t,o} \geq \text{Ready time}_{t,o} + \text{Stop time}_{t,o}. \quad (5)$$

The following safety constraints are added to eliminate the possibility of collisions at stations. These constraints ensure that two opposite trains, $t1$ and $t2$, approaching do not enter station s simultaneously.

$$\text{Arrival}_{t1,s} \geq \text{Arrival}_{t2,s} + \text{Safety interval},$$

or:

$$\text{Arrival}_{t2,s} \geq \text{Arrival}_{t1,s} + \text{Safety interval}. \quad (6)$$

5. Each station can accommodate trains using two kinds of track, i.e., platforms and non-platforms; passenger trains stopping for boarding and alighting are planned to use platforms for the comfort of passengers and trains stopping for other reasons are routed to other lines. However, generating a tight schedule for a set of long-distance trains on a single-line railway may make the violation of such a rule inevitable, let alone the consequences of any unforeseen events in real life. For any station with p platform and n non-platform lines at any point in time, τ , within the scheduling horizon:

The no. of trains stopped on platforms $\leq p$,

The no. of trains stopped on non-platforms $\leq n$.
(7)

In cases where the above line assignment rule has to be overridden, the total number of trains stopped must be less than, or equal to, $p + n$.

The definition of a performance measure (mathematical objective function) can help measure the quality of any solution and compare solutions generated by different scheduling methods. Such measures are also used for directing moves across solution space looking for higher quality solutions. In this work, the term ‘‘imposed waiting time’’ is used to refer to a delay imposed on a train at a station, due to the occupation of the requested section of main line by another train. When departure time at the origin and running time on the track are predetermined, the scheduling aim could be to distribute the imposed waiting time amongst involved trains, with respect to an appropriate criterion. In this paper, SWWT, AUWT and MRWJ measures are used.

SWWT (The Sum of Weighted Waiting Time)

In the real world, trains are of different value in the eyes of passengers and managers of a railway system.

Therefore, each train can be assigned a weight and the objective of a scheduling system may be to reduce the SWWT, as represented below:

$$\text{Minimize } \sum_{i=1}^{\text{Trains}} \left(\text{Weight}_i \times \sum_{\text{Stations}} \text{Im posed waiting time} \right).$$

AUWT (Average Unit Waiting Time)

The spread of delays, with respect to journey duration, appears to be important where fast, slow, long- and short-distance trains are involved. For a given journey, a fast train is expected to face less delay than a train with lower speed. This is justifiable with regard to the investment made to provide a faster service and the higher fares paid by the customers. Plus, a short-distance train may not be expected to face the same amount of delays as a long-distance train. Therefore, the aim could be to minimize AUWT as represented below, where unit waiting time is the ratio of waiting time to minimum travel time for each train being considered.

$$\min \left[\text{Avr. Trains} \left(\frac{\sum_{\text{Stations}} \text{Waiting time}}{\sum_{\text{Tracks}} \text{Running time} + \sum_{\text{Stations}} \text{Planned stop}} \right) \right].$$

The drawback of AUWT is the possibility of generating very good schedules for most trains at the expense of generating very bad schedules for a few others.

MRWJ (Maximum Ratio of Waiting Time to Journey Duration)

MRWJ is better when allocating the same amount of waiting time to a long-distance train, where a short-distance train is undesirable and the problem includes both types of train. MRWJ is represented as:

$$\min \left[\max_{\text{Trains}} \left(\frac{\sum_{\text{Stations}} \text{Waiting time}}{\sum_{\text{Tracks}} \text{Running time} + \sum_{\text{Stations}} \text{Planned stop}} \right) \right].$$

For a given schedule, the lower the maximum ratio of

delay to journey duration, the higher the quality of the schedule would be.

SOLUTION TECHNIQUES USED

Simulated Annealing Algorithms

This technique refers to a process which simulates the process of cooling a collection of hot vibrating atoms. The basic annealing algorithm was proposed by Metropolis, Rosenbluth, Rosenbluth, Teller & Teller [24], while simulated annealing was initially developed by Kirkpatrick, Gelatt & Vecchi [25] for solving combinatorial optimization problems, which yielded promising results. This technique originates from the conjunction between the annealing process and certain characteristics of the resulting substance. It is believed that, in practical contexts, the state of a substance at low temperature is remarkably affected by the annealing process and, thus, a slow cooling process is recommended.

The algorithm is a search control strategy for finding the best, or nearly the best, solution for a problem without exhaustive navigation of the solutions space. It is intelligent in the sense that it is capable of escaping local optimums and plateau and it is not a brute force search technique. SA (Simulated Annealing) algorithms operate under the assumption that a neighborhood can be constructed for identifying adjacent solutions, which can be reached from a solution already generated. The neighborhood mechanism depends on the nature of the problem being treated and the way it is modeled. For example, pair-wise exchanges (or swaps) are frequently used in connection with permutation problems to create a neighbor solution [26]. In a basic algorithm, the search process gets started from an initial solution generated randomly or by a heuristic and continues in a loop through exploring a randomly selected solution in the neighborhood of the current solution at any iteration. For a minimization problem, the move to a better solution (downhill move) is made automatically and the move to an inferior one (uphill move) probabilistically, with the probability of $e^{-\Delta/T}$, where T is called temperature and $\Delta = (\text{new solution's objective function value} - \text{current solution's value})$. This is a simple, general probabilistic guidance strategy for avoiding the local search descent disadvantage [27]. T is initially set to an appropriately high value to facilitate the selection of low quality solutions in early stages. Then T declines gradually, by cooling rate α , where $0 < \alpha < 1$, to reduce the probability of accepting lower quality solutions and converge into better ones. Also, the number of iterations using the same T value increases gradually by $\beta \geq 1$ to allow more effort with lower T values, hoping to achieve further improvement in later stages of the search process. The process stops

when T reaches a value close to zero.

Constraint Satisfaction Problems

A Constraint Satisfaction Problem (CSP) involves a finite set of variables, where each variable has a finite domain and a set of constraints restricts the assignment of values to the variables. The primary aim of solving the problem is to find a solution, which satisfies the constraints involved. In this context, constraints can contribute to the efficiency of a solution method in different ways. A major contribution of the constraints is to problem modeling; constraints make sure that the encoding of a correct model of the problem is indeed a step towards an efficient running programme. Another contribution of the constraints is propagation, which spontaneously produces the consequences of a decision [28]. Constraint propagation may be viewed as a way of using constraints to reduce the computational efforts required to solve a problem, where constraints are not used only for testing the validity of solutions, but, instead, are used to detect inconsistencies and reduce the domains of variables. For example, if variables x and y denote integers, from $(x < y)$ and $(x > 10)$, one can deduce that the value of y is at least 12 and smaller values must be removed from its domain. If constraint $(y \leq 11)$ is added later, a contradiction is detected. Without propagation, the test $(y \leq 11)$ cannot be performed before the instantiation of y .

There are certain methods which use constraint propagation to increase the efficiency of search techniques for solving hard combinatorial problems; for example, look-ahead techniques propagate constraints after labeling any variable by assigning a value, in order to reduce the domains of unlabelled variables and to detect if the problem is insoluble [29]. Once a variable is labeled, domains of unlabelled variables are pruned by removing values, which are inconsistent with the values already allocated to labeled variables.

“HeuSA”, THE HYBRID MODEL

The scheduling model, HeuSA, represents a combination of simulated annealing with a constraint-based heuristic (CBH) and constraint propagation technique. First, a feasible solution is generated using CBH, then, the neighborhood exploration process starts, according to SA strategies. In fact, the CBH creates the first feasible solution from where the search process gets started. To create a solution, it resolves conflicts between trains requesting a block, according to the earliest time they leave the block; thus, the train which releases the track earlier is given priority to occupy it. Ties are broken, according to the earliest time they are ready to occupy the block, i.e., the train with minimum

start time gets precedence. In this process, the station capacity constraints are also taken into account. CBH tends to minimize total local waiting time over all trains involved.

What is the neighborhood concept of SA in this context? For a given feasible schedule, a neighbor solution is a schedule in which the sequence of occupying a block by two requesting trains with overlapping time intervals is changed, and the destructed part of the schedule resulting from the swap is identified and rescheduled using the same heuristic.

A move in the neighborhood of a solution is conducted using the following steps:

1. Identifying a potential move randomly, which is a train delayed at a station for another moving train to finish the occupation of the requested block;
2. Swapping the sequence of activities (occupying the block by the two trains competing); the earliest start time of the two activities are updated and constraints are posted for propagation;
3. Propagating constraints to identify the activities whose start times must be updated; this process could reveal violations of the resource constraints for the part of the schedule affected by the swapping process. (A look-ahead propagation method is used. Further discussions are provided in [20]);
4. Repairing the schedule to modify the earliest start times of the activities affected by the swap;
5. Evaluating the quality of the new schedule in terms of the performance measure selected;
6. Determining the new current schedule using the

simulated annealing move strategy. Checking the threshold to stop the algorithm appropriately. The search process continues as long as T is greater than a limit close to 0. However, the algorithm stops if the number of iterations exceeds a predefined large number (i.e., 5000 used in this work).

SIMULATION EXPERIMENTS

The passenger trains scheduling model has been tested using the data of two corridors of Iran's railways. Data are working timetables of passenger trains generated by train planning experts for Tehran-Mashad and Tehran-Tabriz, connecting the capital to two major cities in the northeast and northwest of the country, respectively. The two routes are transit lines connecting Turkey in the northwest to Turkmenistan in the Northeast. Five cases have been considered in this work, where a case includes a predictive schedule produced manually for a fleet of trains. The schedule specifies origin and destination stations, departure time from the origin, running time between immediate adjacent stations en route and stop time at any station en route for reasons such as boarding, alighting and meeting opposite trains. In practice, the schedule covers about 24 hours and it is repeated daily or weekly. Table 1 shows further details of the cases solved.

Tables 2 to 4 compare schedules generated by HeuSA and CBH with manual timetables in terms of SWWT, AUWT and MRWJ. Figures associated with SWWT are in terms of minutes and the figures associated with AUWT and MRWJ have been multiplied by 100 to make them more understandable

Table 1. The specifications of cases solved by the HeuSA.

Case	Route	Trains			No. of Blocks		No. of Stations	Route KM
		Up	Down	Total	Single	Double		
P1	Tehran-Mashad	11	11	22	40	10	51	926
P2	Tehran-Tabriz	8	8	16	47	0	48	736
P3	Tehran-Mashad	8	9	17	40	10	51	926
P4	Tehran-Tabriz	5	5	10	47	0	48	736
P5	Tehran-Mashad	8	8	16	36	5	42	795

Table 2. Comparison on the SWWT.

Case		P1	P2	P3	P4	P5
Manual timetable (MT)		446.0	230.0	365.5	139.0	249.5
Constraint-based heuristic (CBH)		423.5	200.0	301.5	101.0	217.0
HeuSA		388.7	204.5	292.3	101.0	215.0
Improvement (%)	CBH over MT	5.04	13.04	17.51	27.34	13.03
	HeuSA over CBH	8.22	-2.25	3.05	0.00	0.92
	HeuSA over MT	12.85	11.09	20.03	27.34	13.83

Table 3. Comparison on the AUWT.

Case		P1	P2	P3	P4	P5
Manual timetable (MT)		3.27	2.23	2.86	1.97	2.40
Constraint-based heuristic (CBH)		2.42	1.98	2.22	1.43	2.05
HeuSA		2.31	2.02	2.16	1.43	2.06
Improvement (%)	CBH over MT	25.99	11.21	22.38	27.41	14.58
	HeuSA over CBH	4.55	-2.02	2.70	0.00	-0.49
	HeuSA over MT	29.36	9.42	24.48	27.41	14.17

Table 4. Comparison on the MRWJ.

Case		P1	P2	P3	P4	P5
Manual timetable (MT)		7.13	4.74	5.38	4.70	5.80
Constraint-based heuristic (CBH)		6.58	4.54	5.26	3.67	5.69
HeuSA		5.13	4.51	4.65	3.60	4.87
Improvement (%)	CBH over MT	7.71	4.22	2.23	21.91	1.90
	HeuSA over CBH	22.04	0.66	11.60	1.91	14.41
	HeuSA over MT	28.05	4.85	13.57	23.40	16.03

in percent. Since random variables are incorporated into the neighborhood exploration, each case is solved using six random generator seeds and the average value is reported for each objective function. Simulation experiments show that all solutions created by CBH and HeuSA are better than the manual ones; however, the degree of improvement varies with case and objective function. Test results for CBH show that SWWT has decreased by between 5.04% and 27.34%, AUWT by between 11.21% and 27.41% and MRWJ by between 1.90% and 21.91%. On average, improvements achieved by HeuSA are between 11.09% and 27.34% for SWWT, 9.42% and 29.36% for AUWT and 4.85% and 28.05% for MRWJ.

In three out of thirty experiments (five cases, three objective functions, two algorithms, i.e. CBH and HeuSA), HeuSA did not make any improvement over CBH, although solutions were, in terms of objective function value, only slightly different. In fact, the probabilistic search process started from a very good solution generated and navigated solution space by CBH and no improvement achieved by SA. SA can be tuned to progress the search process more carefully and for a longer period of time to outperform the CBH.

Table 5 shows search efforts made by HeuSA in terms of CPU time and the number of visited solutions, which is represented by the number of iterations. CBH created the initial solution for each case in less than two seconds. Experimental results reveal that between 4.85% and 29.36% improvements have been achieved, whilst consuming 48 to 228 seconds CPU time of a personal computer of type COMPAQ Pentium. The figures in the above tables show that the HeuSA outper-

forms the manual system in terms of the performance measure value and schedule production time. HeuSA also outperforms CBH, since it yields better solutions in 12 out of 15 tests and the gap is very narrow in the three other tests. From a managerial viewpoint, improvements on SWWT, AUWT and MRWJ appear well enough to encourage the use of the scheduling methods presented as the advisory system for the scheduling of passenger trains.

As explained previously, key algorithm knobs are T , α and β . The initial T value in the model is equal to $(f(Q)/\log(0.6))$, where $f(Q)$ denotes the performance measure associated with the first solution from where the search started. In fact, the probability of making an uphill move to a solution with $f(Q') = 2 \times f(Q)$ is set to be 0.60. The above formula is used to limit the search effort, taking advantage of the high quality of the first solution created by the heuristic. Another advantage is that the initial T value is automatically calculated for any problem being solved and the performance measure used.

Table 5. Performance in terms of CPU time and iterations.

Case	SWWT		AUWT		MRWJ	
	Iter.	CPU	Iter.	CPU	Iter.	CPU
P1	1005	228	543	136	387	120
P2	864	110	507	69	367	65
P3	924	181	531	110	377	102
P4	741	72	462	48	357	48
P5	879	128	519	80	372	75

Table 6. Parameters used to solve P1 to P5.

Parameter	SWWT	AUWT	MRWJ
α	0.92	0.92	0.91
β	1.015	1.02	1.01
Iterations	1	1	1

Table 6 shows the values allocated to α and β for SWWT, AUWT and MRWJ. Typically, α is chosen in a range between 0.90 and 0.99 [30]. The initial values of α and β used to start the experiments for tuning the HeuSA, were 0.95 and 1.05 respectively. The final values are shown in Table 6. The table also shows the number of iterations for each T during the search process.

CONCLUSIONS

This paper represented a combination of the simulated annealing and a constraint-based heuristic. Starting the search process from a good solution created by the heuristic, the algorithm improved solution quality while avoiding exhaustive search and reducing the possibility of being trapped in local optima. Train scheduling experts were able to run CBH or HeuSA, using the objective function desired. HeuSA was fast enough to encourage users to modify input data to enhance quality. This is the practical advantage of the automated advisory system introduced; this encourages further research to devise interactive intelligent techniques capable of tackling real world problems. Data mining can also be used to extract the knowledge of scheduling experts to create an initial solution of higher quality. In fact, technical experts who create timetables manually may find it very difficult to explain the heuristics they use or might be reluctant to transfer their expertise; in such cases, existing train schedule databases, created in the past, can be explored using data mining techniques to learn about rules and facts used by experts.

REFERENCES

- Higgins, A. Kozan, E. and Ferreria, L. "Optimal scheduling of trains on single line track", *Transportation Research-B*, **30**(2), pp 147-161 (1996).
- Mcguire, M. and Linder, D. "Train simulation on British rail", in *Proc. COMPRAIL 94*, **2**, pp 437-444, Madrid, Spain (Sept. 1994).
- Backman, J. "Railroad capacity and traffic analysis using SIMON", in *Proc. COMPRAIL 96*, **1**, pp 183-192, Berlin, Germany (Aug. 1996).
- Klahn, V. and Dannenberg, H. "Computer aided construction of timetables on long route sections", in *Proc. COMPRAIL 94*, **1**, pp 27-34, Madrid, Spain (Sept. 1994).
- Wardrop, A.W. et al. "Timetable capture and train flow modeling", in *Proc. COMPRAIL 92*, **1**, pp 177-190, Washington DC, USA (Aug. 1992).
- Lidén, T. "The new train traffic simulation program developed for Banverket and its design", in *Proc. COMPRAIL 92*, **1**, pp 533-544, Washington DC, USA (Aug. 1992).
- Van Dyke, C.D. and Davis, L.C. "Computer dispatching simulation of high density, mixed freight and passenger train operations", in *Proc. COMPRAIL 92*, **2**, pp 119-130, Washington DC, USA (Aug. 1992).
- Wolf, G.P. and Baugher, R.W. "Development and application of an advanced microprocessor-based line capacity and train scheduling model", in *Proc. COMPRAIL 92*, **1**, pp 569-588, Washington DC, USA (Aug. 1992).
- Peterson, E.R. and Taylor, J. "A structured model for rail line simulation and optimization", *Transportation Science*, **16**(2), pp 192-206 (1982).
- Carey, M. and Lockwood, D. "A model, algorithms and strategy for train pathing", *Journal of Operational Research Society*, **46**(8), pp 988-1005 (1995).
- Mills, R.G.J. et al. "Dynamic rescheduling of long-haul trains for improved timekeeping and energy conservation", *Asia-Pacific Journal of Operational Research*, **8**, pp 146-165 (1991).
- Brännlund, U. et al. "Railway timetabling using Lagrangian relaxation", *Transportation Science*, **32**, pp 358-369 (1998).
- Mees, A.I. "Railway scheduling by network optimization", *Mathl. Comput. Modeling*, **15**(1), pp 33-42 (1991).
- Goh, C.J. and Mees, A.I. "Optimal control on a graph with application to train scheduling problems", *Mathl. Comput. Modeling*, **15**(2), pp 49-58 (1991).
- Cai, X. and Goh, C.J. "A fast heuristic for the train scheduling problem", *Computers and Operations Research*, **21**(5), pp 499-510 (1994).
- Lin, H-C and Hsu, C-C "An interactive train scheduling work-bench based on artificial intelligence", in *Proc. of the 6th International Conference on Tools with Artificial Intelligence*, New Orleans, LA, IEEE Comput. Soc. Press, pp 42-48 (Nov. 1994).
- Schaefer, H. and Pferdmenges, S. "An expert system for real-time train dispatching", in *COMPRAIL 94*, **2**, pp 27-34, Madrid, Spain (1994).
- Chiu, C.K. et al. "A constraint-based interactive train rescheduling tool", in *Proc. of the 2nd International Conference on Principle and Practice of Constraint Programming-CP96*, Cambridge, USA (Aug. 1996).
- Higgins, A. et al. "Heuristic techniques for single line train scheduling", *Journal of Heuristics*, **3**(1), pp 43-62 (1997).
- Isaaï, M.T. and Singh, M.G. "An object-oriented constraint-based heuristic for a class of train scheduling problems", *IEEE Transactions on Systems, Man*

- and Cybernetics, Part C: Applications and Reviews*, **30**(1), pp 12-21 (2000).
21. Isaai, M.T. and Singh, M.G. "An intelligent constraint-based search method for single-line passenger-train scheduling problems", in *Proc. of the 2nd International Conference on the Practical Applications of Constraint Logic Programming*, pp 79-91, Manchester, UK (April 2000).
 22. Sharma, G. et al. "A knowledge-based simulation approach (K-SIM) for train operation and planning", *Simulation*, **62**(6), pp 381-391 (1994).
 23. Tomii, N. and Ikeda, H. "A train rescheduling simulator combining pert and knowledge-based approach", in *Proc. of the 7th European Simulation Symposium, ESS'95*, pp 534-538, Erlangen-Nuremberg (Oct. 1995).
 24. Metropolis, N. et al. "Equation of the state calculations by fast computing machines", *Journal of Chemical Physics*, **21**, pp 1087-1091 (1953).
 25. Kirkpatrick, S. et al. "Optimization by simulated annealing", *Science*, **220**(4598), pp 671-680 (May 1983).
 26. Glover, F. et al., *Genetic Algorithms and Tabu Search: Hybrids for Optimization*, Graduate School of Business, University of Colorado at Boulder (July 1992).
 27. Osman, I.H. and Kelly, J.P. "Meta-heuristics: An overview", I.H. Osman and J.P. Kelly, Eds., in *Meta-Heuristics: Theory and Applications*, USA, Kluwer Academic Publishers, 2nd Printing, pp 1-21 (1997).
 28. Wallace, M.G. "Practical applications of constraint programming", *Constraints Journal*, **1**(1), pp 139-168 (1996).
 29. Tsang, E., *Foundations of Constraint Satisfaction*, London: Academic Press Limited (1993).
 30. Elmohamed, M.A.S. et al. "A comparison of annealing techniques for academic course scheduling", in *Proc. of the 2nd International Conference on Practice and Theory of Automated Timetabling (in: LNCS 1408)*, pp 92-112, Canada (Aug. 1997).