COMPUTERIZED TRAIN SCHEDULING

N.B. Nedeljkovic
Westrail

N.C. Norton
Westrail

ABSTRACT:

Most of the existing train scheduling computer models are designed either to solve immediate train control problems or to broadly simulate potential future situations. There is a clear need for a model to assist in the production of master train schedules.

Westrail has developed heuristic techniques to generate master train schedules recognising the relative priorities of trains. The method relies heavily on man-machine interaction, particularly when deciding which of several feasible timetables is operationally preferable. The ideas were translated into a FORTRAN program, the Single Line Train Scheduler (SLTS), which caters for manual and automatic signalling and can be used on single and double track lines.

In addition to printing the timetable, the SLTS program draws the associated train diagram on a VDU or on paper. The model is being implemented in Westrail for generating master train schedules.

1. INTRODUCTION

Effective train scheduling is an essential component of all railways' operations. Poor scheduling results in unnecessary delays to passengers and/or customers' goods, with a consequent loss of business; higher crew costs; and greater investment in track, locomotives and rollingstock.

The need for efficient techniques to develop train schedules has been particularly evident in a number of recent Westrail (1) planning exercises. For example, a study of motive power reduced the perceived need for new locomotives by six units, a considerable capital saving which would not have been achieved without the preparation of a number of train schedules. More recently Westrail has developed a series of new marketing and operational initiatives to enable it to operate as a viable commercial organisation in the recently deregulated Western Australian transport market. To develop these initiatives a series of train schedules had to be prepared appropriate to a variety of scenarios.

In both these examples the preparation of train schedules was time consuming, but essential for the planning process to be effective.

The approach taken to planning train movements depends on the time horizon being considered. In long term planning research is usually directed to the adequacy of physical resources, particularly track, locomotives and rollingstock. Some simulation models have been developed which can be applied in long term planning (see for example Rudd and Storry, 1974(b)), but we know of no such model which is precise and which can be applied to generate timetables for operational use.

In contrast to long range planning is train control, where decisions on train schedule alterations have to be made quickly, with little or no possibility for changing the locomotive and crew schedules. Consequently, train control computer techniques developed to date aim to minimise train delays. Compared to train scheduling problems in the planning phase, problems in train control involve a smaller number of trains and line sections, considered over a shorter time span. They are

---

(1) Westrail is the trading name of the Western Australian Government Railways.

therefore more amenable to a mathematically rigorous treatment, an example of which is the model developed by Sauder and Westerman, 1983, which uses the "branch and bound" integer programming technique to evaluate every combination of possible train crossings and select those with minimum overall delay.

Between these two extremes (long range planning and operational control) lies short term planning, including the preparation of train timetables for operational use. In short term planning the total availability of physical resources is constrained, but crew, locomotive and rollingstock schedules are not fixed and in fact must be considered as a part of the train scheduling problem. In our experience the total train, locomotive, crew and rollingstock scheduling problem is generally considered as a series of subproblems each of which is solved largely independently. These subproblems may be sequential in nature (eg assigning crew schedules to a previously prepared train schedule) or non-sequential with a need for integration (eg independent scheduling of movements by geographic divisions). An overall solution to the scheduling problem is obtained by repeatedly adjusting the subproblems and their solutions until an integrated set of solutions satisfying all system constraints is derived.

The Operations Research Section of Westrail started to research the train scheduling problem about three years ago with the aim of developing techniques to assist with the production of train schedules, and to substantially reduce the time taken to finalise a schedule.

The first step was to review available literature on the subject to avoid duplicating the efforts of others. One of the most promising lines of research appeared to be the development of the Single Track Simulator (STS) by Rudd and Storry, 1974(a) and 1974(b), and in 1981 Westrail made changes to this model in an attempt to convert it to the short range planning scheduling tool it sought. STS was designed as a simulation model for use in long range planning and does not provide for many of the features required for the preparation of operational timetables. For example, there is no provision in the input data to STS to specify scheduled stops to a train.

In 1982 Westrail abandoned STS and began development of a new model. In this paper, we describe the philosophy and techniques adopted and describe the capabilities and constraints of the new model in its current state of development.

In summary, the new procedure uses heuristics with the emphasis on the interaction between the computer and the user. The user retains the responsibility for ensuring operational objectives are met but delegates much of the scheduling process to the computer model, which produces train schedules satisfying physical constraints and safeworking regulations. The user is provided with great flexibility in controlling train movements and in complex situations with heavy railway traffic one can expect three of four user-computer interactions before a satisfactory solution is obtained. The computer model, called the Single Line Train Scheduler (SLTS), has been written as a computer program which produces a train diagram (a time-distance graph representing the schedules) and train timetable in less than 5 CPU minutes (2). The operator then examines the diagram and specifies any changes required in broad terms, or in detail, before re-applying the model.

It was successfully tested a number of times on Westrail's busiest lines (between Robb Jetty and Collie and between Kwinana and Kalgoorlie) as well as on rural lines with different signalling modes. A typical example involved 57 sections and 40 trains per day, with line occupancy of two thirds of the available time within the busiest 3 hours. As a result of the tests, Westrail has purchased a dedicated computer system on which the SLTS model is being implemented both for long range planning purposes and to compile master timetables.

The following sections provide a general description of the problem and the model's logic and capabilities, together with a sample timetable and sample train diagram.

2. THE TRAIN SCHEDULING PROBLEM

Most of the published work on vehicle scheduling relates to road or air transport. With train scheduling different problems are encountered. Safety regulations require a section of line to be clear before a train can enter it, resulting in a variety of train control procedures depending primarily on traffic density (sophisticated centralised traffic control systems are installed only on the busiest lines). Further, throughout Australia and in many overseas countries the majority of railway lines are single track. Trains

---

(2) On an IBM 4341 computer.

travelling in opposite directions can only cross where passing loops have been constructed, as shown in Fig. 1a.

A crossing results in at least one of the two trains being delayed, and the selection of which train to delay at each crossing is the crux of the train scheduling problem.

There are several effects of delaying a train. Each delay

- changes the viable locations at which the train may cross other trains on its route, and may increase the number of trains it has to cross;
- increases the journey time and delays the arrival time of the train to the disadvantage of clients;
- delays the time at which the locomotive and rollingstock will become available for other traffic;
- increases crew working time; and
- results in a loss of momentum, which is particularly undesirable for heavily laden trains.

Consequently when two trains meet the decision as to which train to delay depends on

- the type of train (generally passenger trains have a higher priority than freight trains);
- the effect of the decision on other train movements;
- the effect on clients and other railway operations of delaying the arrival time;
- the potential for utilizing the locomotive and/or rollingstock on other traffic after the trains arrival; and
- the length of time the crew has been working.

The scheduling model that has been developed by Westrail employs an interactive approach which combines an operator's detailed knowledge of the constraints on train movements with the processing power of the computer. The operator makes a judgement of each train's relative priority taking account of the variety of operating constraints s/he is aware of (but which are not examined in detail by the computer program). The computer performs a number of detailed calculations required to determine train travelling times and provides the capacity to rapidly evaluate a wide range of possible crossing decisions, taking account of the delays that ensue and the priority of the trains being delayed. The operator reviews the resulting train schedule and can either:

- change the input data, including train priorities (representing his judgement of a variety of operating considerations), and rerun the program in search of a better solution, or
- identify particular crossing decisions s/he wishes to override and instruct the program accordingly.

After a few iterations the operator will have one or more potential train schedules which satisfy both programmed operating constraints and those constraints and targets known to the operator.

The computer requires an explicit description of the network and trains. A knowledge of the input data required by the model makes it easier to understand the problem and the various calculations required to determine train times and to ensure that restrictions imposed by safety considerations are satisfied. The input data for the SLTS model are as follows.

(i) Track Configuration:

- Station name
- Station position (distance from a reference point)
- Number of passing loops at each station
- Crossing delays (time required for signalling and switching at each station)

- Safeworking at each station (staff or automatic signalling)
- Section type (single or double track) (3)
- Minimum intervening time between two trains occupying the section
- Section travel times for each speed group (4)

(ii) Train Details:

- Origin station
- Destination station
- Train priority
- Train number
- Speed group
- Earliest departure time from the origin
- Scheduled stops, stations and durations
- Preceeding train number (required only for "offset" trains) (5)

(iii) Run Details:

- Heading (for reports)
- Weighting factors for different train priorities (6)

Output from the computer model is in the form of a printed train timetable and a graphical train diagram. The train timetable and the movement of trains along a railway line can be conveniently represented by train diagrams with the X and Y coordinates representing time and distance, respectively. An example of a train diagram is depicted in Fig. 2. The lines representing train movements are called train paths, each of which is completely specified by the arrival and departure time at each station.

---

(3) In accordance with Westrail rules and signalling arrangements (except in emergency situations) trains occupy the left hand line of a double track section.

(4) Section travel times are classified into speed groups so that all trains belonging to one speed group have identical travel times to other trains in the group.

(5) See section 4 of this paper.

(6) To take account of train priorities, the delay to a train is multiplied by the weighting factor associated with the train's priority ranking. The program then seeks to minimise the total of the weighted delay times.

The main restrictions are that train paths cannot intersect on single track sections and that the number of trains in a station at any time cannot exceed the number of passing loops. Traffic safety considerations (such as headway times) also have to be met. Two (or more) trains are said to conflict if they cannot reach the next station without one being delayed by altering its train path (Fig. 1b). The problem of train scheduling is to construct a timetable such that all above restrictions are satisfied and the total of weighted delays is minimized.

The problem is similar to the job scheduling situation, with the sections of track representing machines, the stations representing bins and the loops at stations thought of as bin capacities.

The restrictions in the train scheduling problem are however much more severe and that handicaps the generation of feasible solutions (Otway, 1980). In fact, for railway lines with heavy traffic, the construction of a single feasible solution is not a trivial task.

3. METHOD OF SOLUTION

In this section the decision process employed in the computer processing phase is considered in more detail.

3.1 Train Scheduling as a Multi-stage Decision Process

The problem described in the preceeding section can be perceived as a multi-stage decision process where at each stage a choice among a number of alternatives is to be made. Thus, the set of all feasible solutions to a train scheduling problem can be viewed as a possibilities tree T, with branches taking place at times of decisions regarding conflict resolutions.

In the adopted approach the tree T, which represents the totality of possibilities resulting from resolving all conflicts in all possible ways, has only a conceptual meaning as it is never generated by the program. Instead of it, the program generates train paths (from the origin to the destination) in the order of train priorities and departure times from the origin. If a train conflict is detected, a number of (feasible) resolutions are investigated and only one of them, considered to be the best option, is retained.

In this process already scheduled paths can be altered (if found to be in conflict with a new path). At any point in time the program retains only a partial feasible schedule, which is progressively extended to include new paths until a complete solution to the problem is reached.

The main difference between the SLTS decision process and the methods employed in the Single Track Simulator (Rudd and Storry 1974(a) and 1974(b)) is that the Single Track Simulator moves all trains simultaneously, one section at a time, in order of train departure times. Conceptually it has an internal clock and all trains are scheduled (or more correctly their movements are simulated) as the internal clock advances. This procedure is computationally fast but suffers from the inability to step backwards in time to improve on previously scheduled (simulated) movements.

The new SLTS model's decision processes are not time based but train based. Further, as additional train paths are generated previous conflict resolutions can (and are) changed.

The quality of the final SLTS program solution depends on how correctly the ways to solve individual conflicts are selected. As it was found that priority rules alone could not give satisfactory answers, a look-ahead method has been developed to resolve train conflicts.

3.2 The Look-Ahead Method

Suppose that a partial feasible schedule, denoted by Go, is available and that train r is to proceed from its present station to the next station (7). Two possible situations can arise:

(i) train r can proceed without conflict
(ii) a conflict with one or more trains is detected.

---

(7) We follow the terminology suggested by Hart et al., 1968, although the SLTS look-ahead method is quite different to their approach.

Case (i) is simple as train r can proceed to its next station without delay. If (ii) occurs, then the look-ahead method (consisting of elementary subroutines) is applied to the existing partial feasible schedule Go. It is assumed that all the train paths in Go, excluding those belonging to conflict trains, are fixed and cannot be altered. Further conflicts can be encountered while attempting to resolve the original conflict. They are resolved in the same manner, by applying the look-ahead method.

The whole process can be thought of as the generation of the tree T' shown in Fig. 3 where the nodes correspond to conflicts and branches to different options for resolving conflicts. Any complete path in T', from its root node to an end node, represents a solution to the conflict.

The further one looks ahead the better the solution that is obtained. The highest level employed in the generation of the tree T', determines the depth of the look-ahead method. For practical reasons (computing time, available memory and complexity of the program) one has to limit the number of levels in the tree T'. (Without a limit on the number of levels, the whole process could become endless, resulting in a cyclic graph rather than in a tree). To resolve the conflict, the complete path in T' with the minimum weighted delay is selected.

As in the case of the tree T, the complete tree T' is never actually generated. A set of selection rules has been developed to limit the number of branches at nodes (which makes it practical to look-ahead to a greater depth). Consequently paths with large weighted delays are discarded.

The main conceptual differences between the SLTS method and the method proposed by Cherniavsky, 1972, are:

(i) In contrast to the method of Cherniavsky, 1972, which generates pseudo solutions, the SLTS considers only feasible solutions.

(ii) Conflicts can and do involve several trains, not just two trains.

(iii) The SLTS look-ahead method does not require a part of the train diagram to be extracted in order to obtain tree T', but examines all the train paths in the current partial feasible solution. Existing train paths which are associated with conflict trains may be altered.

(iv) SLTS introduces man-machine interaction into the decision process.

Obviously, the final solution depends very much on the quality of the heuristics employed in generating trees T and T'.

4. IMPLEMENTATION & EXAMPLES

The SLTS model provides for intermediate (centralised traffic control) signals, scheduled stops, single or double track sections of line and sections of line out of service for a specified period (due to maintenance). Particularly useful is the facility to specify different train priorities at specific stations which enables the planner to "force" a desired crossing decision at a particular station. This facility would normally be used after an initial run, as part of the interactive process.

A facility to handle unattended manual stations differently to other stations is incorporated in the program. A train must stop at an unattended station with manual signalling to obtain authority to proceed except when another train is already waiting to cross (enabling the authority to be collected "on the run").

In addition, under current operating procedures in Westrail, crossing delays at unattended manual stations depend on the length of the train (both the guard and fireman have to walk to the staff cabin), so the program also includes an option to double the 'normal' delays of selected trains at unattended manual stations. As headway times and crossing delays are input to the program, SLTS can be used either for manual or automatic signalling, or a combination of both.

A provision is made for a "chain" of trains by allowing for so called "offset" trains with an earliest departure time from the origin equal to the arrival time of a specified train at that station, plus a specified "offset" time. Each chain must have a first and last train (i.e. a chain cannot be closed). The offset train facility is extremely useful in situations where the same physical train is loaded and unloaded in a cyclical manner, or when locomotive utilisation or other considerations dictate a relationship between the arrival time of one train and the departure time of another.

The program generates a timetable for a full week with different train schedules each day. The 24 hour clock is used for arrival/departure times, whilst - and + signs in front of times denote the fact that the event occurs the previous and following day, respectively. Other than this the layout of the timetable (a sample of which is given in Fig. 4) is self-explanatory.

In addition to producing the timetable, the program plots train diagrams (Fig. 2) on a VDU screen, or on paper. To view the VDU diagram in more detail the operator can specify which part of the diagram is to be displayed, as illustrated in Fig. 5 which shows the enlarged bottom-left quadrant of the previous diagram.

The program consists of some 4000 executable statements in 36 subroutines. Since it is written in FORTRAN it is virtually machine independent. So far it has been successfully tested on IBM 4341, HP 9000 and ICL PERQ-2 computers.

5. CONCLUSION

The SLTS model has been successfully tested under real-world constraints. It is an efficient train scheduling algorithm which provides quick answers (5 CPU minutes per computer run, approximately) for complex train scheduling problems on a single railway line.

Consequently, it relieves the planner of tedious and time consuming work allowing him to try many more alternatives than would otherwise be possible. All too frequently the time consuming nature of the train scheduling process results in the first feasible solution being adopted.

With the SLTS a number of potential schedules can be quickly generated, enabling fine tuning of master schedules. In a planning environment the SLTS speeds up the planning process and enables more operating options to be examined within realistic time frames.

In addition to the generation of train diagrams and timetables for general operating or planning purposes, the model is also suitable for determining (present and future) line capacities, 'best' ways of upgrading the track and for evaluation of alternative marketing strategies and operating regimes by varying the speed (and or length) and number of trains.

Further developments to the model are planned. In particular it is intended to develop locomotive and crew scheduling routines which link into the existing model to create a comprehensive scheduling package for trains, locomotives and crews.

REFERENCES

CHERNIAVSKY A.L., (1972) A Program for Timetable Compilation by a Look-Ahead Method, Artificial Intelligence 3, pp 61 - 76.

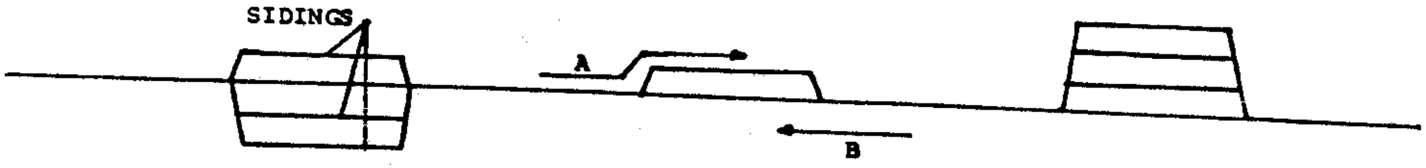
HARI P., NILSSON N.J. and RAPHAEL B.A., (1968) A Formal Basis for Heuristic Determination of Minimum Cost Path, IEEE Trans. SCC 4(2), pp 100 - 107, July.

OIWAY N.J., (1980) Scheduling Problems with Restricted Intermediate Storage, Ph.D. Thesis, Department of Mathematics, University of Adelaide, July.

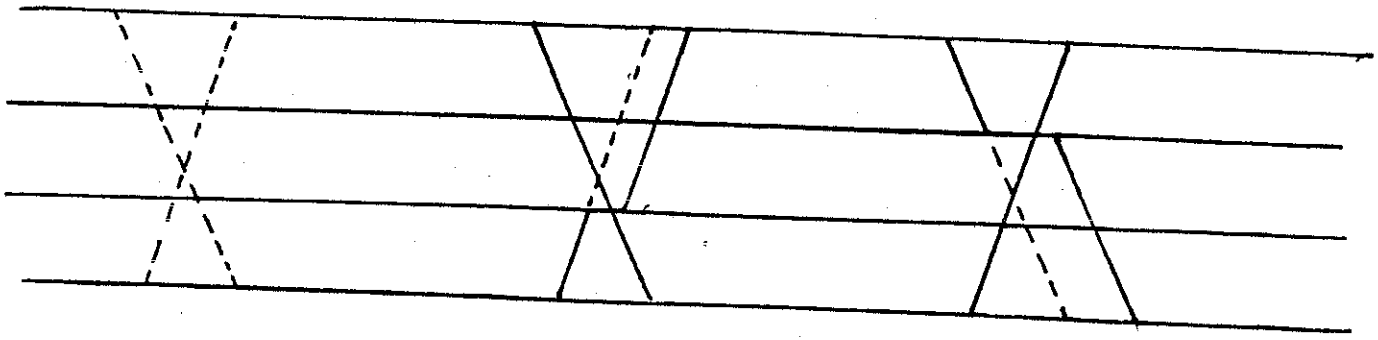
RUDD D.A., and SIORRY A.J. (1974a) Single Track Railway Simulation - New Models and Old, Form SDI0058, IBM Systems Development Institute, Canberra.

RUDD D.A., and STORRY A.J, (1974b) Single Track Simulation (STS), Form SDI0057, IBM Systems Development Institute, Canberra.

SAUDER R.L. and WESIERMAN W.M. (1983) Computer Aided Train Dispatching: Decision Support Thru Optimization Norfolk Southern, Atlanta, Georgia.



(a) Single track railway. A & B are trains going in opposite directions.



(b) "Intersection of train paths in a section" and two ways of solving the conflict.

Fig. 1

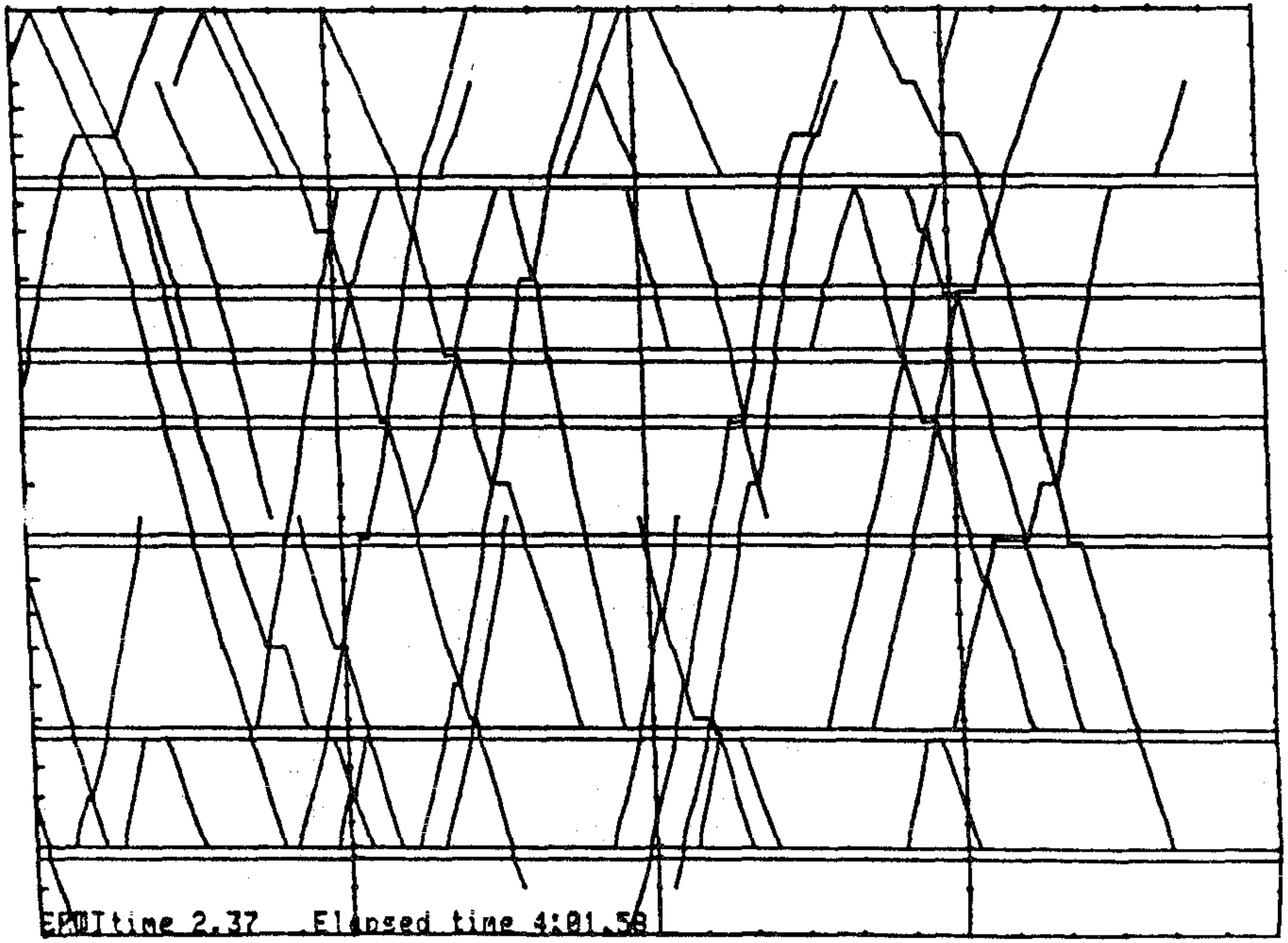


Fig. 2 Sample Train Diagram

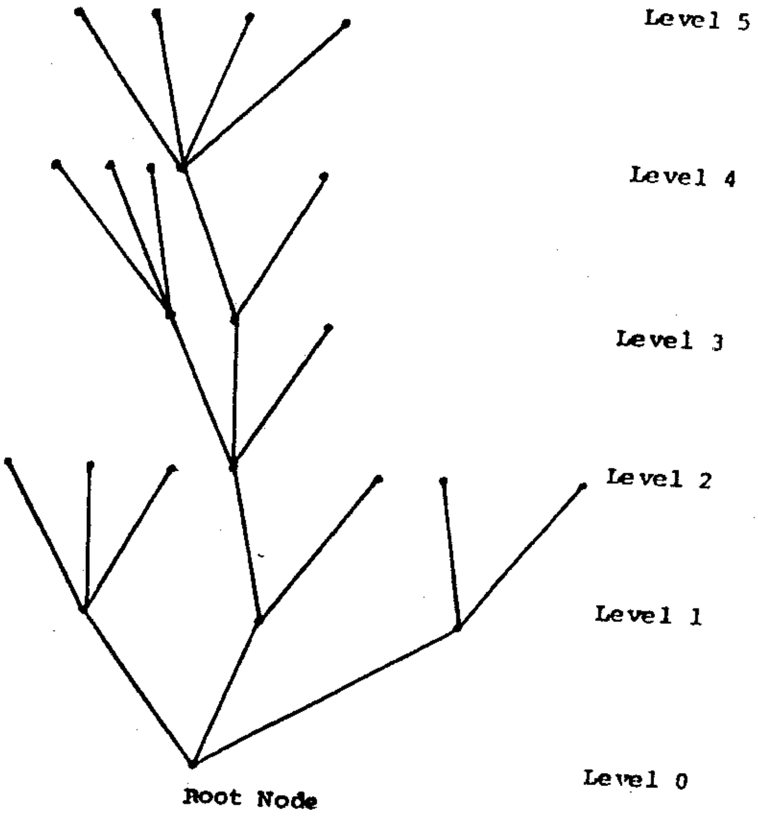
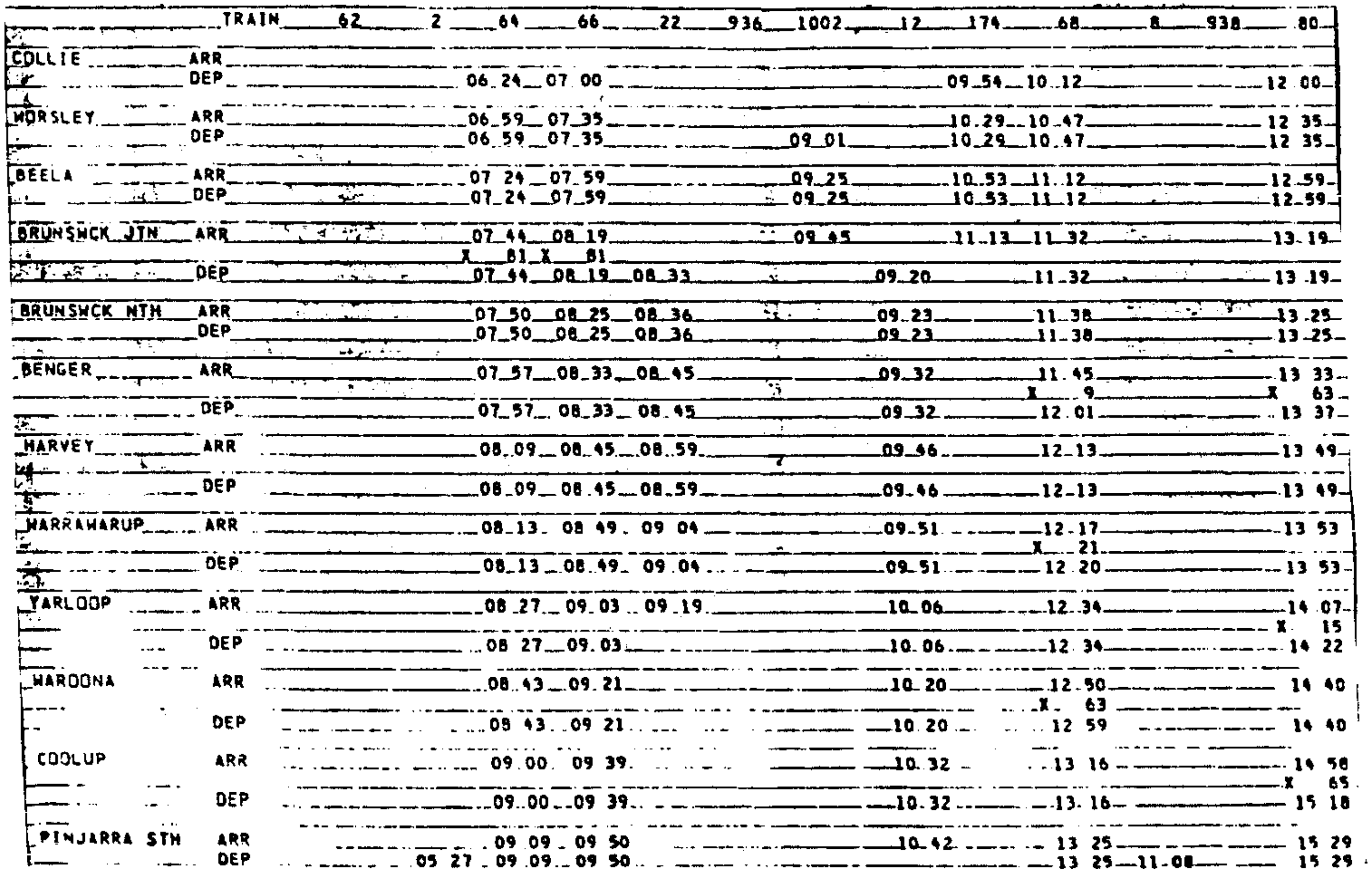


Fig. 3 Example of tree T' of depth 5 generated when resolving a conflict.



	TRAIN	62	2	64	66	22	936	1002	12	174	68	8	938	80
COLLIE	ARR													
	DEP			06.24	07 00					09.54	10.12			12 00
WORSLEY	ARR			06.59	07.35					10.29	10.47			12 35
	DEP			06.59	07.35			09.01		10.29	10.47			12 35
BEELA	ARR			07 24	07.59			09.25		10.53	11.12			12.59
	DEP			07.24	07.59			09.25		10.53	11.12			12.59
BRUNSWCK JTN	ARR			07.44	08.19			09.45		11.13	11.32			13.19
				X 81	X 81									
	DEP			07.44	08.19	08.33			09.20		11.32			13.19
BRUNSWCK NTH	ARR			07.50	08.25	08.36			09.23		11.38			13.25
	DEP			07.50	08.25	08.36			09.23		11.38			13.25
BENGER	ARR			07.57	08.33	08.45			09.32		11.45			13 33
											X 9			X 63
	DEP			07.57	08.33	08.45			09.32		12.01			13 37
HARVEY	ARR			08.09	08.45	08.59			09.46		12.13			13 49
	DEP			08.09	08.45	08.59			09.46		12.13			13 49
WARRAWARUP	ARR			08.13	08 49	09 04			09.51		12.17			13 53
											X 21			
	DEP			08.13	08.49	09.04			09.51		12 20			13 53
YARLOOP	ARR			08 27	09.03	09.19			10.06		12.34			14 07
														X 15
	DEP			08 27	09.03				10.06		12 34			14 22
WAROONA	ARR			08.43	09.21				10.20		12.50			14 40
											X 63			
	DEP			08 43	09 21				10.20		12 59			14 40
COOLUP	ARR			09.00	09 39				10.32		13 16			14 58
														X 65
	DEP			09.00	09 39				10.32		13.16			15 18
PINJARRA STH	ARR			09 09	09 50				10.42		13 25			15 29
	DEP		05 27	09.09	09 50						13 25	11-08		15 29

Fig. 4

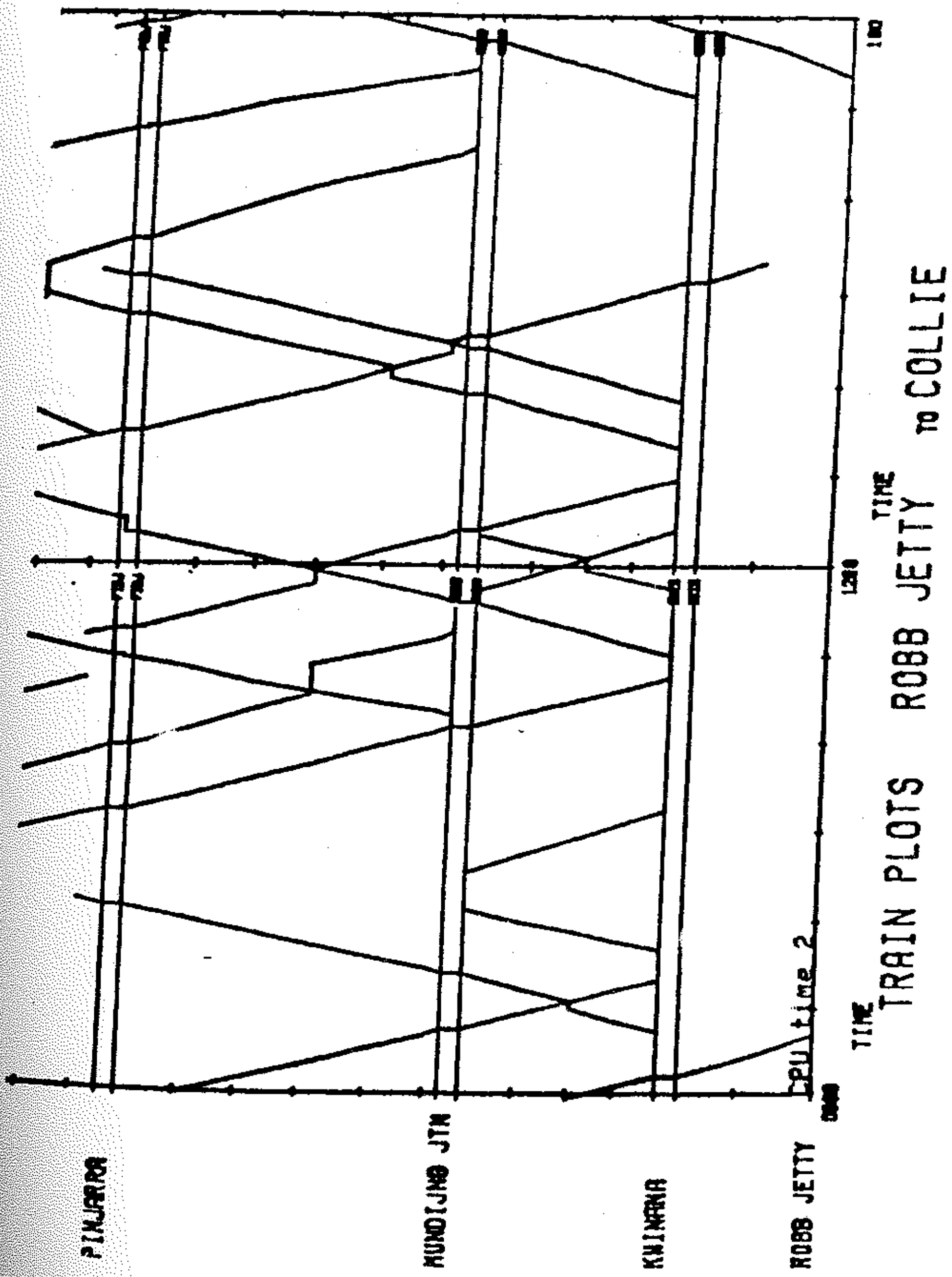


Fig. 5. Enlarged bottom-left quadrant of Fig. 2