White Rose Research Online
eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Leeds.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/91214/

Version: Accepted Version

Article:

Wren, A, Fores, S, Kwan, A et al. (2003) A flexible system for scheduling drivers. Journal of Scheduling, 6 (5). pp. 437-455. ISSN: 1094-6136

https://doi.org/10.1023/A:1024854522373

© 2003, Kluwer Academic Publishers. This is an author produced version of a paper published in Journal of Scheduling. The final publication is available at Springer via http://dx.doi.org/10.1023/A:1024854522373. Uploaded in accordance with the publisher's self-archiving policy.

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

A Flexible System for Scheduling Drivers

Anthony Wren, Sarah Fores, Ann Kwan, Raymond Kwan, Margaret Parker and Les Proll [*] [†]

School of Computing, University of Leeds, UK

SUMMARY

A substantial part of the operating costs of public transport is attributable to drivers, whose efficient use therefore is important. The compilation of optimal work packages is difficult, being NP-hard. In practice, algorithmic advances and enhanced computing power have led to significant progress in achieving better schedules. However, differences in labour practices among modes of transport and operating companies make production of a truly general system with acceptable performance a difficult proposition. TRACS II has overcome these difficulties, being used with success by a substantial number of bus and train operators. Many theoretical aspects of the system have been published previously. This paper shows for the first time how theory and practice have been brought together, explaining the many features which have been added to the algorithmic kernel to provide a user-friendly and adaptable system designed to provide maximum flexibility in practice. We discuss the extent to which users have been involved in system development, leading to many practical successes, and we summarise some recent achievements.

KEY WORDS: driver scheduling; public transport; mathematical programming; heuristics

1 Introduction

The problem of scheduling public transport and its drivers has received much attention, particularly among the OR community, over many years. Wren [1] has been involved since 1967 in exploring many different solution techniques which can be applied to NP-hard public transport scheduling problems, and, in particular, to the driver scheduling problem. The most successful commercial driver scheduling packages, e.g. TRACS II [2] and HASTUS [3], model the problem using mathematical programming.

Mathematical programming approaches are basically of two types. Generate-and-Select techniques, including TRACS II, rely on a pre-generated set of shifts. As, for computational reasons, this set can only be a small subset of those potentially available, such techniques cannot guarantee to find an optimal solution to the model. In principle, column generation/branch and price techniques [4, 5] can overcome this difficulty by generating 'useful' shifts from the implicitly defined full set of potential shifts, as and when required. In practice, even using these latter techniques, the heavy computational load means

---

[*] Correspondence to: Les Proll, School of Computing, University of Leeds, Leeds LS2 9JT, UK

[†] Email: lgp@comp.leeds.ac.uk

that proven optimal solutions to the model cannot, in general, be found. Fortunately intelligent use of heuristics and increasing computer power allow, in many cases, acceptable schedules to be achieved in reasonable times, using either generate-and-select or column generation.

More recent solution attempts have included methods which employ local search strategies to improve current schedules. Kwan et al [6] describe the use of genetic algorithms and suggest how they can overcome the limitations of a totally mathematical programming approach in solving ever larger problems. Smith et al [7] discuss how constraint programming can be used to reduce the problem size and aid subsequent attempts to find a schedule. Shen and Kwan [8] describe a tabu search heuristic for constructing shifts. Despite some successes, none of these approaches has been shown to improve consistently on mathematical programming over a wide range of problem scenarios.

Most of the theoretical aspects of the TRACS II system have been presented elsewhere [2, 9, 10, 11, 12, 13, 14, 15]. The purposes of this paper are to show how theory and practice have been brought together in a user-friendly and adaptable system providing maximum flexibility. We present a picture of the whole, and discuss some of the features which have been added to the algorithmic kernel to make it more usable in practice and to highlight some examples of its use.

In Section 2 we explain our motivation, while Section 3 outlines the driver scheduling problem. Section 4 provides an overview of the TRACS II system, while Section 5 introduces the user interfaces. Section 6 surveys some of the highlights of the system development over a period of twenty years. Section 7 summarises some uses of TRACS II, showing several of the difficulties which have been overcome. The present state, and some possible future developments, are summarised in Section 8.

2 Motivation

The specific scheduling needs of transport undertakings vary greatly, depending on the nature of the area being served, the type of operation (road or rail), the views of individual managements and drivers' representatives, and historic developments. Many driver scheduling systems are installed for users after lengthy tests and modifications lasting several months, and are thus very costly. Our design philosophy therefore has been to provide a modular system which requires a minimum of change from one application to another, and which can be easily be used by schedulers with a PC, whether or not they are computer literate. To this end we have designed TRACS II with a wide range of parameters which reflect the needs of most organisations, and can be easily understood by newcomers. The algorithms are independent of the scheduling rules. A large set of potential driver shifts (up to a few million) is generated early in the process, each candidate being tested against predefined parameters. This set is refined, first by heuristics, and then by a specialised integer linear programming (ILP) process, in order to obtain a near-optimal working schedule. The system is driven by a graphical user interface.

TRACS II serves both as a regular scheduling tool and as a device for determining the costs of proposed revisions of drivers' conditions or new service levels. Its benefits lie both in its ability to achieve substantial operating efficiencies and the power it gives to management to examine speedily alternative operating scenarios.

3 Driver Scheduling

Once a vehicle schedule has been devised, the problem of driver scheduling involves partitioning the vehicle work into driver *shifts* which must be valid according to labour rules, while the total set of shifts (the *schedule*) should reflect the operator's definition of efficiency. The measure of efficiency may be the total number of shifts used, the total cost in paid hours, or some combination of the two; it may occasionally include a measure of subjective quality.

A typical shift will constitute a day's work for a driver, and will generally include assignments to more than one vehicle, with provision for at least one meal break. Although we refer here for simplicity to a day's work, we include night shifts which may start late on one day and finish during the morning of the next day. We also refer in this text to the scheduling of a single day at a time, although this may include the option of associating some morning work with night shifts from the previous day.

Drivers can only change vehicles at specified points. These are usually places conveniently located to an interchange or rest area or where there is suitable transportation to or from another such point. The location/time pairs of such places can be identified from the vehicle schedule and are known as *relief opportunities (ROs).* The indivisible time periods between successive ROs on a particular vehicle have to be allocated to a single driver and are known as *pieces of work.* A shift will often include consecutive pieces of work on one vehicle followed by work on other vehicles. The gap between such spells of work on different vehicles may be long enough to allow a meal break, or it may simply conform to rules regarding the minimum time necessary to change vehicles. Figure 1 shows a section of vehicle work and an example of a valid shift that might be formed.

Many factors are involved in specifying working conditions of drivers, including a maximum total driving time, a maximum spell spent driving without a break, time periods in which breaks may take place, minimum duration of meal breaks, etc. These may be different for each operating organisation. Not only does the specific range of acceptable hours vary, but particular working practices may be encouraged or prohibited, e.g. by limiting the number of shifts with particular characteristics. Labour agreements are also often different in different countries or for different transportation types. Geographical considerations also affect the type of shift which may be appropriate; for example, long-distance train drivers may work on only two vehicles, out and back, while intensive urban bus operation may allow several changes of vehicle in order to package most efficiently the work within a minimum set of shifts.

Large transport undertakings often involve the use of many driver depots. While urban bus companies frequently restrict their drivers to vehicles from their home depots, allowing the total driver scheduling problem to be sub-divided, there is often more flexibility within regional companies, although there may be restrictions on the routes or vehicle types which may be operated by drivers from particular depots. Scheduling of large train operations tends to be more complex. We have successfully solved problems in which drivers have been spread among more than twenty depots, while drivers from any one depot are only qualified to drive particular routes and particular traction types; as any particular route can generally be driven by drivers from more than one depot, the total problem cannot efficiently be sub-divided. There may be restrictions on the numbers of drivers available at particular depots. It may also be necessary to constrain the numbers of shifts of particular types at certain depots, for example to ensure that there is a fair balance of early and late shifts.

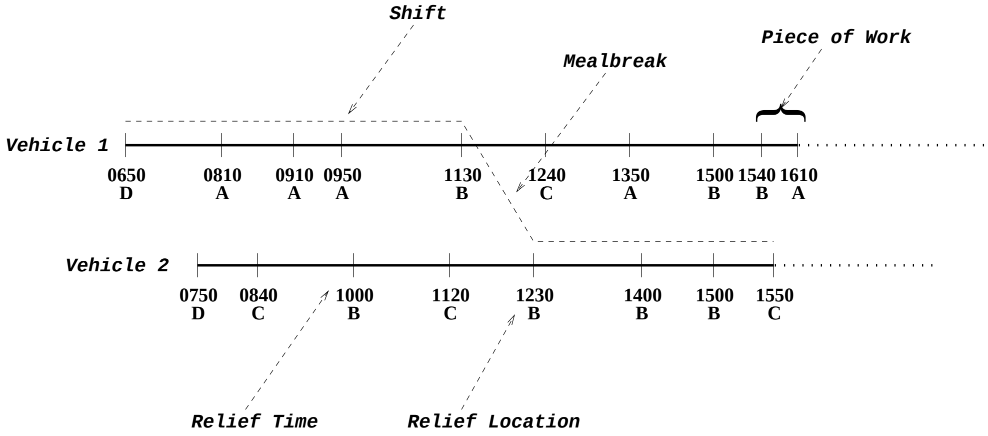


Figure 1: A Section of Vehicle Work

Drivers may have to travel from a signing-on point to the point where they are to commence driving, or between points where successive spells of work finish and start, or from the point where they finish driving to a signing-off point. Such travel may be on foot or by taxi (in which case a standard time allowance is given), or by a frequent bus or train service (in which case a standard time may also be appropriate), or as a passenger on one or more scheduled services (in which case it is necessary to refer to the timetables of any services which may be used).

3.1 *Capturing user requirements*

Many, but not all, of the organisations with which we have worked have a set of written rules setting out the agreements between management and unions, which govern the composition of driver shifts. Some of the organisations also abide by unwritten rules. When we started investigating driver scheduling 35 years ago, we spent considerable time with several different bus operating authorities in order to compile a body of rules which, by suitable adjustment of parameters, would together meet all the conditions we had met, both written and unwritten. This was initially a difficult and complex process.

Our normal pattern in the early days of our research was to try to extract all relevant rules at an initial meeting, and to obtain sample copies of some existing schedules. We would then examine the schedules, looking for examples where the rules as we understood them had apparently been violated. These examples were then raised with the company management, and a better understanding was achieved. We would then try ourselves to create a schedule by hand, and would present this to management. Sometimes this would lead to our being told that we had violated some further unwritten rules, or had misunderstood some situation. Sometimes too, we would be told that we had failed to take advantage of some hidden ways of circumventing the given rules.

Two typical difficulties of our early days may illustrate some of our difficulties in capturing the rules. Much of our early work was done with Leeds City Transport. About two years after starting this we were able to present to management the first computer-produced schedules that appeared satisfactory to us. We were told that we had broken some rules which had not previously been given to us. On asking why these rules were only being revealed now, we were told that they had not wanted to discourage us by telling us all the difficulties at once! Two years work had to be restarted.

Another example came from Dublin in the 1970s. An important constraint in driver scheduling governs the maximum continuous time a driver may spend in charge of a bus. This was not written in the list of rules given to us, so we asked for clarification. We were told that the maximum was 4hrs 30mins, but the chief scheduler added the rider that they only exceeded this "if necessary". We asked when it would be necessary, and were told that it would be necessary whenever it enabled them to make an improvement. Under pressure they conceded that the absolute maximum was 4hrs 45mins, but after some thought one of the schedulers added that they hardly ever exceeded this! When we examined their existing schedules we found an example of 5hrs 7mins continuous working time, so we set this as our maximum, adding penalty costs to any shift with a stretch exceeding 4hrs 30mins.

Despite such difficulties, by the 1980s we had been able to put together a comprehensive list of parameters which would often be sufficient to satisfy all the needs of a new user, although new situations would arise from time to time. The first live user of one of our systems was London Transport buses, and we worked closely with LT staff over a period of 18 months to ensure that all their conditions could be met. As a result, the system was installed successfully at the end of 1984, and is still in use by some of LT's successor companies today.

As our portfolio of successful installations has grown, so the attitudes of schedulers have changed from a position of disbelief to one in which most potential users are keen to ensure that the system will meet all their requirements. Nowadays good schedules can be obtained for most new users simply by appropriate settings of a very comprehensive list of parameters. However, it is important that we do not pressurise schedulers into accepting anything that it not completely satisfactory. From time to time we extend the system's parameter list. TRACS II separates the building of potential shifts from the optimisation phase, and the generation of potential shifts itself follows context-free processes. These processes invite the system to validate potential shifts, and the validation routines can be expanded as necessary to include new features.

When we turned to train driver scheduling in 1994 we had to increase the parameter list quite extensively, and indeed we re-wrote the shift generation process completely. However, the rail scheduling rules were normally well defined, and it was not difficult to add the relevant new features while ensuring that all previously captured features of the bus industry were preserved.

4 TRACS II

TRACS II is a driver scheduling system consisting of a suite of modules. It is designed to be as portable as possible, so its main optimising component is independent of the rules governing the construction of any shift. Our earlier IMPACS system was installed for London Transport Buses in 1984 and has been used as the main bus driver scheduling tool in London ever since. IMPACS and TRACS II together have been used to construct real operational schedules in around 100 bus and train companies, either as the regular driver scheduling tool, or to test alternative strategies or to make once-off schedule improvements. Most recently, TRACS II has been installed in each of the twenty-five operating companies of the UK's largest bus group, with around 10,000 vehicles. Each of these companies has a different historical background; some are former city-owned operations, others are regional private companies, while two are divisions of the former London Transport. The scheduling conditions therefore vary greatly between companies.

Essentially, TRACS II has two main phases: generation of a large set of potential shifts, possibly up to a few million or more, and selection of a subset of these shifts which together form a schedule. In practice, there are several stages in the solution process of TRACS II, controlled through a workbench or user interface (Section 5):

- Data preparation, input and validation;
- Prespecification;
- Identification of travel possibilities between ROs;
- Generation of one or more sets of valid potential shifts;
- Reduction of the shift set (if required);

- Merging sets of shifts;
- Computation of penalties;
- Selection of a subset of shifts which covers the vehicle work and minimises an objective function;
- Postprocessing;
- Output;
- Shift validation.

4.1 *Data preparation, input and validation*

The system depends on two main data files. The first, the TRA (Travel) file, contains details of the vehicle work to be covered (represented by sequences of ROs), together with information about any other services which may be used by drivers travelling as passengers between relief points. This file also contains data regarding relief points to be used, a table of standard walking or riding times between them, allowances for drivers to sign on or sign off at each point, details of depots and any restrictions affecting them, and other data relating to the physical system to be scheduled. The first process of TRACS II, VALIDATE, checks the format and logic of the data in this file.

The second main file, LAB (Labour), specifies the rules governing the formation of individual shifts. Some of these rules are hard, in the sense that they must be followed, while others are soft, in the sense that they are used by the scheduler to guide the system to generate a reasonable set of potential shifts out of the many billions that might be allowed by only the hard rules. Parameters within this file define the different types of shift which are allowable, and for each type of shift such features as the maximum amount of continuous driving, the positions within a shift of meal breaks, and their minimum durations, the minimum amount of work within a shift, and the maximum elapsed time from start to finish of a shift. Some of the features may be considered as hard parameters by some users and soft parameters by others.

4.2 *Prespecification*

A facility is provided whereby the user may specify in advance that certain shifts, or features of shifts, must be included in the schedule. This may be used when special features are required, even if they might detract from optimality. When a user specifies a complete shift, that shift need not be legal according to the parameters specified; sometimes a scheduler knows that there is vehicle work which cannot be covered without a dispensation to break some rules.

Partial shifts may be specified. For example, it is possible to specify the start of a shift, and/or its finish, allowing the system to determine how best it should be filled out; this is particularly useful when it is intended that morning and afternoon school runs should be undertaken by the same driver. The user may also specify certain ROs when drivers must be changed, or may link two ROs together to show that a driver finishing a spell at the first of these should then take up another vehicle at the second specified RO. Partial specification must be such that it is possible to construct a legal shift including the specified feature.

Although prespecification may be useful in certain circumstances, it is generally used sparingly. Many users are happy to allow the system to produce the full schedule without their intervention.

4.3 *Identification of travel opportunities*

It has already been stated that drivers may have to travel between portions of a shift, or to or from its start or finish. In urban bus operation it is often sufficient to specify standard walking or riding times between each pair of points, but in train operation, and in some bus operation, drivers have to travel as passengers on scheduled services. The TRAVEL component determines, for each RO (time and place) specified in the TRA file where the driver may leave a vehicle, the earliest time that a driver can reach each of the other points on the system. The journey may involve travelling on up to two scheduled services, with intermediate walking if appropriate. Similarly, for each RO, it determines the latest time a driver may leave each of the other points in order to take over the driving of a vehicle at the RO in question. It should be noted that the system calculates travel opportunities for each of the ROs separately; the best route to take between two relief points depends on the availability of scheduled services at the time in question.

TRAVEL uses the vehicle data, together with the table of standard walking or riding times, to compute possible passenger journeys for drivers. It is possible for a driver also to use services which are not to be scheduled (possibly services provided by a different company), or to join or leave a vehicle at a place which is not a relief point on the service in question. Such services and additional points are identified in the data with codes indicating that they are not to be used other than for passenger travel by drivers. It is also possible to specify times at which taxis might be used in certain circumstances.

TRAVEL produces a file detailing all the many passenger travel opportunities. It is quite normal to have over 1000 ROs, and up to 100 actual relief points, so a further file is produced here which is used in the output (Section 4.10) to provide a detailed description of the journeys used in the course of any shift.

4.4 *Generation of one or more sets of valid potential shifts*

The next stage, BUILD, makes use of the above travel information to generate potential shifts which will later be subjected to the scheduling process itself. It considers all the possible combinations of pieces of work which satisfy the given parameters and can be accommodated within a shift with no more than four spells of continuous work. Although manually produced schedules frequently have shifts with more than four spells of work, this has usually arisen because human schedulers frequently have difficulty in fitting all the work efficiently into good shifts; they are left with some pieces of work which they exchange with small pieces of shifts they have previously constructed until they are able to fit everything together, often in a less than efficient manner. It has been our experience that TRACS II has always been able to create schedules more efficient (see Section 3) than the manual ones without exceeding four spells of work, and frequently without using more than three. Between any two spells of vehicle work there is necessarily unproductive time, so schedules with fewer separate spells of work per shift are inherently more efficient. Despite this, we do know of cases where it might be desirable to have a few shifts with five or more spells of work; the case for this can often be identified from properties of the data, and we envisage allowing the system in future to create a few potential shifts with more than four distinct spells.

Since the number of possible legal combinations of pieces of work is generally enormous, BUILD applies several filtering heuristics during the generation process to eliminate some partial shifts that are considered to be redundant. Such partial shifts include those which are entirely contained within other partial shifts with the same start and end ROs but covering more driving work. There is no upper limit to the number of shifts generated by BUILD. The size of the final set depends on the number of ROs, the frequency of ROs on any particular vehicle, and the slackness of the parameters. A typical scheduling run might generate 100,000 potential shifts, but cases with well over a million have been processed.

If all pieces of vehicle work cannot be covered by a BUILD run, the user has two options. The parameters can be relaxed and the full run repeated, in which case an unnecessarily large number of shifts may be generated. Alternatively, a new run may be undertaken in which relaxed parameters are used to generate only shifts which contain work which was not covered by the first run. It is sometimes appropriate to run BUILD several times with different parameters in the LAB file, rather than to do a single run using parameters with very wide ranges in order to cater for all possibilities, and TRACS II allows this. The sets of shifts generated by different runs may be merged later (Section 4.6).

4.5 *Reduction of the shift set*

The selection phase (Section 4.8) can sometimes be speeded up without loss of quality if the number of shifts generated above can be reduced intelligently. The SIEVE process may be applied separately to each or any of the sets of shifts generated above. It ranks each potential shift using a combination of three attributes: an index which reflects its apparent efficiency (time worked divided by time paid), the number of other potential shifts covering the piece of its work which is covered by the smallest number of other shifts, and the average number of other potential shifts covering the individual pieces of work making up the shift. The user specifies a target number of shifts to be retained from the total set. The lowest ranked shifts are discarded, and the last two attributes of the remaining shifts are updated; this process continues until the target is reached. The user may then inspect a table showing for each of the apparent efficiency indices the number of shifts discarded or retained, and may choose to reinstate a certain number of these, in which case the most efficient are reinstated first.

SIEVE was very important some years ago because the ILP stage (Section 4.8) was rather limited in the size of problem which it could handle. It has been used less often in recent applications, given improvements in the sophistication of the generation process and in the power of algorithms used to solve the ILP.

4.6 *Merging sets of shifts*

When more than one set of potential shifts has been generated by BUILD, these may be merged at this stage to give a single set, with any duplication removed. This feature is frequently used in UK bus applications involving mixed urban/rural operations in which rural-based drivers commonly work to different agreements than do urban-based drivers.

4.7 *Computation of penalties*

The user may associate penalties with undesirable, albeit legal, features of shifts. These are defined through the Workbench and applied to each remaining generated shift at this stage. Users may repeat the process from this stage with different penalties in order to achieve desired features of a schedule, but regular users normally work with a single set of penalties which they have developed from experience.

Penalties are also used to generate ‘minimum change’ schedules. These may be used either when short-term changes are made to the vehicle schedule or when permanent small adjustments are made, but it is desired for operational reasons to retain as much of the existing shift pattern as possible. In these circumstances, TRACS II reads the existing driver schedule within the BUILD process (Section 4.4), which ensures that shifts in that schedule, all of whose ROs still exist, are present in the set generated and are assigned negative penalties.

4.8 *Selection of a subset of shifts*

The set covering model (1) is used to select a subset of the potential shifts which together cover all the vehicle work. Since not all valid shifts have been generated, it may not be possible to cover all pieces exactly once as would be required by a set partitioning approach. However any objective to maximise schedule efficiency should implicitly aim to minimise the number of pieces of work allocated to more than one shift. Any overlap may be edited afterwards, or may indicate that a relaxation of parameters might be adopted in order to allow shorter shifts.

Given a problem with M pieces of work and N previously generated potential shifts, we can define the set covering model to be:

$$\begin{array}{ll} \text{Minimise} & \sum_{j=1}^{N} D_j x_j \\ \\ \text{Subject to} & \sum_{j=1}^{N} A_{ij} x_j \geq 1 \text{ for } i = 1,..,M \\ \\ & \text{Plus any side constraints} \\ \\ & x_j = 1 \text{ if shift } j \text{ is used in the solution}, 0 \text{ otherwise} \\ & A_{ij} = 1 \text{ if shift } j \text{ covers piece of work } i, 0 \text{ otherwise} \\ & D_j \text{ is determined by the objective function used} \end{array} \qquad \mathbf{(1)}$$

Objective function. Three different objective functions are available in TRACS II. The user may choose to minimise the number of shifts, minimise shift costs, or minimise a lexicographically ordered combination of the two using a Sherali weighted cost function described by Willers et al [12]. Costs of individual shifts may be modified by subjective penalties calculated during the previous stage. It is also possible to attach a cost to any piece of work covered more than once.

Constraints. There are necessary constraints on the model (1) which arise from the set covering nature of the model. The user may also add side constraints governing the number of shifts in the schedule in various combinations of shift type and depot, the total number of shifts or the schedule cost.

Solution method. The integrality conditions are first relaxed to find the optimal linear programming (LP) relaxation according to the objective function chosen, then a specialised branch and bound process is adopted to obtain an integer solution representing a schedule. As computing power and memory has become less of an issue, more shifts have usually been generated than in the past, but storage space is not unlimited, and it is inefficient to store within the LP process the large number of generated shifts which will not be used. For problems which have a small generated set of shifts (currently fewer than 30,000) a dual approach, described by Willers et al [12] is used. If the shift set is larger, then a primal SPRINT-like [16] technique described by Fores et al [14] is employed, which ensures that all relevant generated shifts are considered when forming the optimal LP, but not all are stored internally. Both primal and dual methods are accelerated by providing an initial solution. Several heuristic methods of doing this are provided within TRACS II, and the default constructs a schedule by sequentially looking at the currently uncovered piece of work having the fewest shifts available to cover it. From these shifts, that covering the largest duration of currently uncovered work is selected, thus attempting to cover all work with a small number of shifts.

Both processes guarantee optimality of the relaxed model over all available shifts, but we do not normally make all shifts available in the search for an integer solution. Smith and Wren [9] showed over a range of experiments that a good integer solution could normally be obtained using only those ROs (basic ROs) used by shifts forming the basis of the relaxed LP, and this conclusion has been confirmed by later practical cases. The user may therefore choose to eliminate at this stage all shifts which use any non-basic ROs. If the primal approach has been used, the process will have already selected a subset of shifts from all those available in such a way as to retain a range of shifts which should be useful to the branch and bound search, as well as to major iterations of the primal simplex method. Although adequate solutions are normally found by proceeding to the branch and bound stage with this reduced subset, there will be other potential shifts which are not present in this subset, but use only basic ROs. The user may specify in advance that such shifts should be added to the subset, up to a current limit of 30,000 shifts in total.

If an objective function is used which requires shift minimisation, a target cut is introduced after the LP relaxation has been solved. The target cut is a side constraint which specifies that the number of shifts must be exactly (or, optionally, at least) the rounded up number of shifts in the LP solution. The branch and bound search requires a branching strategy, i.e. rules to select the node to explore next, and to construct the branches. There are a number of node selection rules available in TRACS II which consider the extent of the integer infeasibility at the node, and how much the objective function has degraded at this node. There are four entities which can be branched on: shifts and relief opportunities, which proved satisfactory for single depot problems, and types and depots, which have been added recently to aid the solution of multi-depot problems (see Fores et al [2]). These are used in a hierarchical manner; several such hierarchies are available. Within the branch and bound phase, dual steepest edge [17] is used to solve the LP at each node. The branch and bound process continues until either a specified number of nodes has been expanded or a sequence of nodes has been encountered without any improvement to a previously found integer solution.

4.9 *Postprocessing*

It is possible in the solution to a set covering problem that some pieces of work may be included in more than one shift. The user is presented with a list of duplicate covers, and may remove such overcover by manually deleting appropriate shifts. Alternatively, TRACS II will remove overcover automatically, following heuristic rules, such as removing overcovered work from the beginning or end of a shift, or in such a way as to reduce the work content of the longer shifts. Such rules are acceptable to most users.

4.10 *Output*

The standard output from TRACS II is a list of shifts in summary form, showing for each shift the signing-on and -off times, times and places where each vehicle is taken up and left, durations of portions of work, and of meal breaks, the duration of the shift, and its cost. This is appropriate for most bus companies, and can be read together with the vehicle schedule to give full instructions to the drivers. An alternative form of output is similar to that used by most British train operators. This incorporates information from the vehicle schedule to give full details of each shift, including individual trips worked, and specification of journeys undertaken as passengers. These output formats may be customised for individual users as required.

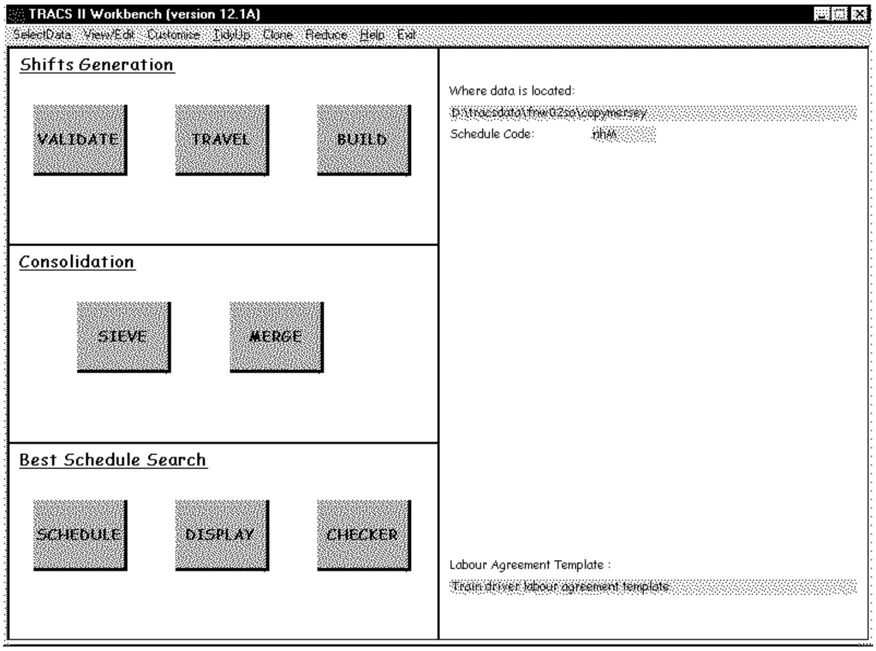
4.11 *Shift validation*

It is possible when a new user is being trained that the first schedule produced is less efficient than expected. In such circumstances it may be necessary to examine carefully the parameters being used in the LAB file, or to determine whether all travel opportunities have been correctly set up in the TRA file. Often a valid shift is being prohibited for some reason such as wrongly set parameters. The user may compare the existing manually produced schedule with that produced by TRACS II. Where a good shift exists in the former, but not in the latter, the user should try to work out why TRACS II has not used the shift. To this end, a CHECKER routine has been provided. The user enters details of the shift in question, and CHECKER reports whether it violates any rules. The user may then alter any appropriate parameters.

While an experienced user will normally obtain a good schedule at the first attempt, if there has been a significant change in scheduling rules or in the nature of the operation it may be that the resultant schedule is not as good as might have been hoped. Although in this case there is no manual schedule for comparison, a good scheduler can often spot that shifts with some characteristics are not present as expected. Again, CHECKER may be used to identify the problem, so that parameters may be adjusted accordingly.

5 User Interface

The modules of TRACS II are run via the Workbench shown in Figure 2. This also provides an interface for creating data files, setting parameters and penalties, cloning data sets to test experimental 'what-if' scenarios etc.

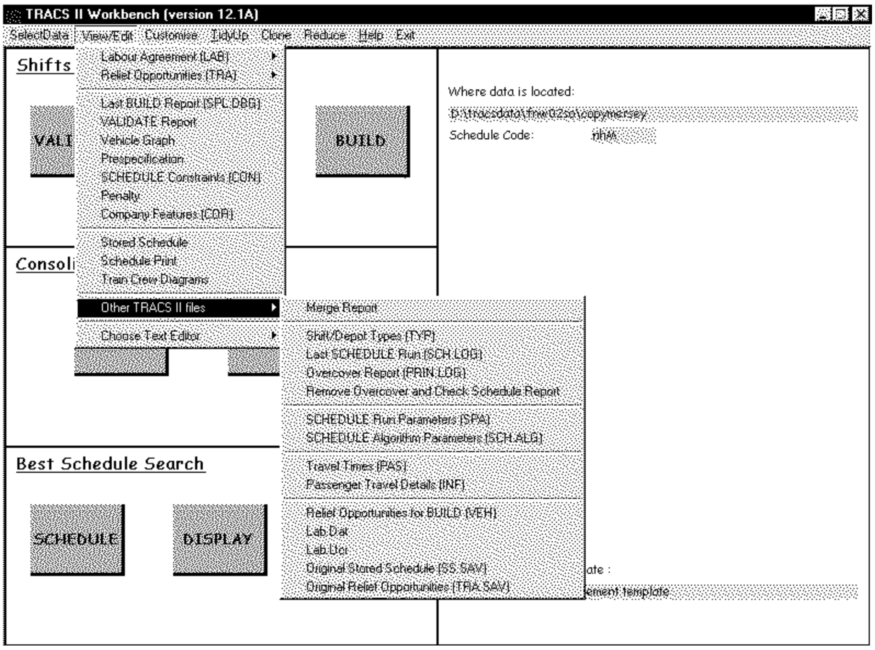


Where data is located:
D:\tracsdata\fnw02so\copymersey
Schedule Code: nhM
Labour Agreement Template :
Train driver labour agreement template

Figure 2: The TRACS II Workbench

The left-hand panel contains push buttons for each module in the suite, while the right-hand panel contains information about the problem being tackled. Each module produces its own report, either on screen, or as a log file, or both, in order to assist the user to re-run stages with altered parameters as appropriate. The toolbar provides access to tools for the creation and editing of data files. For example, the View/Edit button will produce a drop-down menu as in Figure 3.

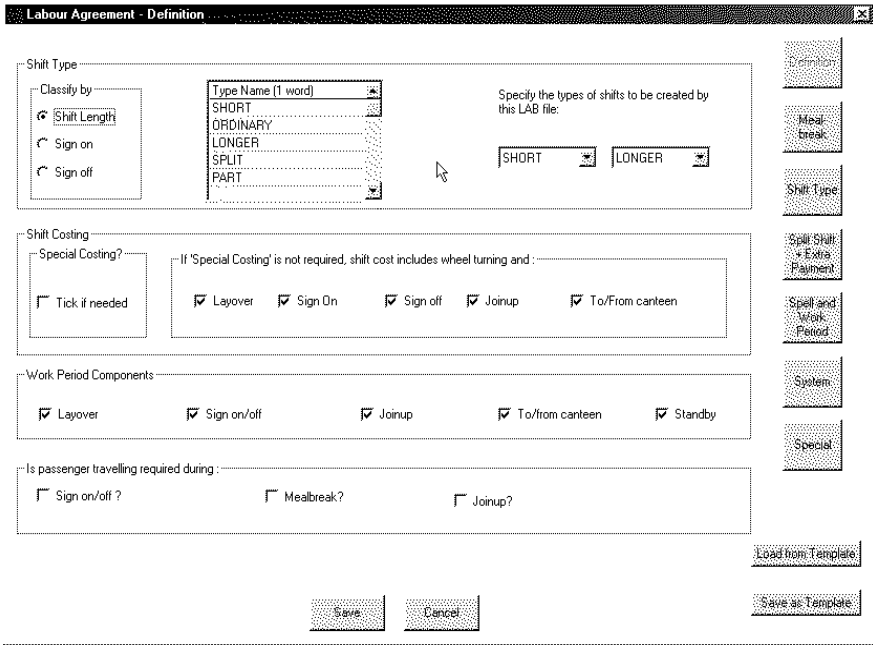
Vehicle work data in the TRA file is usually supplied to TRACS II from an external timetabling system, and can be edited from the Workbench. Occasionally the data may be entered manually. Labour rules may be prepared using a graphical user interface, part of which is shown in Figure 4. This interface contains a total of seven screen pages for editing various parameters that are applicable to most public transport applications. Users can specify different types of shift required in the scheduling problem, e.g.,



Where data is located: D:\tracsdata\fnw02so\copymersey

Schedule Code: nhA

Figure 3: View/Edit Facilities in the TRACS II Workbench



Shift Type

Classify by: ● Shift Length ○ Sign on ○ Sign off

Type Name (1 word)
SHORT
ORDINARY
LONGER
SPLIT
PART

Specify the types of shifts to be created by this LAB file: SHORT LONGER

Shift Costing

Special Costing? ☐ Tick if needed

If 'Special Costing' is not required, shift cost includes wheel turning and : ☑ Layover ☑ Sign On ☑ Sign off ☑ Joinup ☑ To/From canteen

Work Period Components: ☑ Layover ☑ Sign on/off ☑ Joinup ☑ To/from canteen ☑ Standby

Is passenger travelling required during : ☐ Sign on/off ? ☐ Mealbreak? ☐ Joinup?

Figure 4: The Labour Agreement Interface

for four-day weeks, five-day weeks, part-time, etc. It also provides a facility to specify any extra payment to shifts with special features.

The user need only provide details of the vehicle schedule, the labour agreement rules and any penalties to be adopted. All other files are created as earlier stages of the solution process are completed, but defaults used in these files may be altered by the user between stages.

6 System Development

TRACS II incorporates lessons learned through our use of the IMPACS driver scheduling system which we first installed in London Transport Buses in 1984. Although this only allowed 5000 potential shifts and a maximum of fifty nodes in the branch and bound process, it has continued in use with few modifications until the present date. An enhanced version was installed in Greater Manchester Transport in 1985, and subsequently adapted for PC use within the BUSMAN system [18], and installed in about thirty bus and light rail companies until 1992. Work on TRACS II started in 1994; initially this was for rail driver scheduling [19], but throughout the development we ensured that it was capable of solving a comprehensive range of bus driver problems accumulated during our earlier work. Knowledge gained from new situations is continually incorporated into the system in order to broaden the use of the application.

Large problems had to be decomposed for solution by IMPACS, and special routines were developed both to set up sub-problems and to carry over work from one sub-problem to the next. However, as computers became faster and as we introduced more powerful solution processes, the need for decomposition decreased, and TRACS II can now solve problems with several hundred shifts in the resultant schedule. The largest problem tackled to date in a single run had about 2000 pieces of vehicle work and 1.5 million generated shifts.

Many column generation approaches (e.g. [20, 21, 22]) construct new shifts incrementally as their need is recognised within the process, using shortest path, dynamic programming or constraint programming methods. We have avoided this approach, as in our experience, the costs of individual shifts are not incremental and cannot be determined until the whole shift is evaluated. Our approach also allows us to separate the optimising algorithms, which are the most complex part of the system, from the problem specific domain, responsibility for which is largely delegated to the heavily parameterised BUILD module. This is a major contribution to speedily achieving successes with new clients.

Throughout the development and installation of TRACS II, we have worked closely with users to enhance the system, improving the Workbench, adding to parameters and introducing new recommended solution strategies. Solution strategies within the optimising module are controlled by two parameter files, default versions of which are created by the Workbench. The principal parameters in the first of these define the objective function to be used and the length of the branch and bound search. The second allows users more detailed control over the optimising algorithm, including choice of the shift subset to be submitted to the branch and bound search, target cut, branching strategy and granularity of the search. The flexibility of this approach allows users to choose strategies most appropriate to their needs and experiences, and developers to readily perform the experimentation necessary to make recommendations on solution strategies.

Although users have a wide range of parameters which govern solution strategies, most users accept the default settings. However, the branch and bound process cannot guarantee that solutions are found on every occasion, and adjustments may be made if particular difficulties arise in obtaining solutions when new types of problem are presented. It is normal that once good solutions have been obtained, the user can see the types of shift required. At that time it is possible to adjust the shift generation parameters in the LAB file so that fewer potential shifts are generated, after which the default parameters can normally be used to obtain a good solution to the ILP.

The computer time required to obtain a good solution can vary, both in the BUILD process and in the ILP, from a few minutes to many hours, but once suitable parameters have been set for a new user it is normal to expect a solution to the largest problems in about an hour. However, this is dependent on the user's own choice of shift generation parameters. A user who can afford to wait several hours for a solution may choose very slack parameters in the hope that this enables the ILP process to find a better solution, while if time is short, tight parameters ensure that a reasonable solution is found quickly. In experimental work, we have allowed some large problems to run for several days on a fast PC, but we rarely find any significant improvement after a few hours. Such large problems are better solved by intelligently tightening the generation parameters.

Some of the strategies are controlled by the process itself. For example, there is a range of overall branching strategies in the optimisation module which are controlled by one of the parameter files. However, the default strategy and some others are themselves complex strategies which instruct the process to adopt particular branching tactics depending on the immediate situation.

7 Experiences

The two years, 1994-96, during which the original version of TRACS II was being developed, coincided with the fragmentation of the national British Rail organisation into twenty-five separate operating companies, and preparation for their privatisation. About half of these new companies commissioned us to use TRACS II in order to evaluate new operating scenarios involving either new labour rules, or new timetables, or both. For each of these companies, we first ran TRACS II using their existing scenario, demonstrating every time that our system could produce more efficient schedules than the existing manual ones; these first schedules were then used as benchmarks against which the costs of the new scenarios could be evaluated. The largest problem tackled during this period was for a regional rail operator with more than 400 daily shifts and 20 different driver depots. It was necessary to decompose this into five overlapping sub-problems, but some of the sub-problems were large and complex compared to our earlier bus driver experiences, and led to our developing more powerful facilities in later program versions. With the current version of TRACS II, decomposition is no longer necessary [23]. The above exercises, and some subsequent projects, were carried out by our in-house team on behalf of clients. The first application of the system in Reading [24], only three months after TRACS II had been ordered, resulted in an annual saving of approximately £135,000, which was much more than the installation costs. Once this had been achieved, management and unions together used the system to explore the effects of several new operating rules proposed by the unions, resulting in the development of a new set of rules which met some of the union's wishes at marginal increased operating costs.

In 1999 the UK's largest bus group, First, with a total fleet of about 10,000, set out to evaluate a number of commercial scheduling packages. Meilton [25] notes that the evaluation exercise gave 'a comparison of the abilities of each system to solve an identical series of complex duty scheduling problems and, as importantly, a meaningful comparison of the cost for each acceptable answer'. The outcome was that TRACS II was selected as the standard scheduling tool for all twenty-five operating companies, each of which had its own set of scheduling rules, varying from relatively simple broad rules, to very detailed and complex operations involving several depots with different rules of interaction. First indications were that savings on wage cost of more than 2% in situations where scheduling previously had been done manually, and more than 0.5% where previously some form of computer-aided scheduling had been used were attainable. Meilton also notes that 'initially sceptical schedulers are persuaded on the merits of the new system by both its ease of use and quality of results achieved'.

Implementation of TRACS II over the twenty-five operating companies of the First Group (and two others) took place in less than two years from April 2000. During this period many new features were added, both to meet needs of individual companies, and to enhance the TRACS II system. This was accomplished by our working closely with a dedicated member of First staff who helped to identify problems in advance, to help us set priorities, and who smoothed our interaction with staff of the individual companies. The fact that so much was achieved in a relatively short time for companies with wide variations in operating conditions and scheduling rules testifies to our having achieved the objectives set out in Section 2. Training normally takes two days, and schedulers have been able immediately to use the system to develop live schedules. Within the same period our team has produced schedules for a major regional rail undertaking and for a large intensive metropolitan railway. In all measurable cases, the schedules produced have been recognised by the user as being significantly better than manual schedules in terms of the objectives set. Some recent individual experiences follow.

TRACS II has been used to produce train driver schedules for a major train operating company for their summer and winter schedules of 2001 and summer schedule of 2002. These schedules are mainly for evaluation and for comparing to the manual schedules which were produced as parallel exercises. The operation is mainly in the north of England and comprises rural, inter-urban and very intensive urban operations in major cities. One of the main features is the detailed representation of 'route knowledge' by TRACS II. In total, 63 portions of route segment have been identified, some covering portions as small as seven minutes driving time. In many cases, drivers have to travel as passenger to move from one location to another. There are twelve crew depots with the larger ones further divided, giving a total of 19 sub-depots. Some rural depots must have a minimum number of shifts assigned to them. For the crew depots in major cities, there is a preference to minimise the number of shifts assigned to them. The latter case is achieved by penalising potential shifts of undesirable depots. In all exercises, TRACS II produced schedules with up to 8% fewer shifts than the manual solutions.

The work of a large urban bus depot, where drivers could work on any route, gave TRACS II the opportunity to run on a problem in which there were in excess of 1,274 ROs representing nearly 2000 bus hours. Several schedules were produced detailing the work of 290 drivers. Special techniques have been developed in TRACS II to solve successfully problems of this size. They involve being able to identify those ROs which are most likely to be used in shifts in the resulting schedule. In tackling schedules using the strategy of compiling a number of likely shifts and then selecting those which cover all the bus work, the issue of how this should be applied to large problems becomes critical. With the inappropriate selection of parameters it is easy to compile literally millions of shifts. A more careful selection, while producing fewer shifts, might overlook some relief opportunities which could be critical in producing the best schedule. By being able to identify those relief opportunities most likely to be used by shifts in the final schedule it is immediately possible to reduce the number of relief opportunites to be considered. Parameters may be set at a level which, while impossible for the problem as a whole, will yield shifts of the detail required to add significantly to the ease with which a good schedule can be produced. These techniques were employed on this exercise and resulted in a manageable tool for situations where a large network is to be scheduled all together.

TRACS II is in the course of evaluation by a major underground railway, and we have recently undertaken comparisons with manual schedules on four of its operating lines, the largest of which had about 250 daily shifts operating from three drivers' depots. In each case, drivers had to travel as passengers on scheduled trains within their shifts, sometimes on services of other companies. There were complex rules regarding movement of drivers between platforms at some stations, with preference being given to changeovers taking place in stipulated directions of travel. Drivers from some of the depots were allowed to sign on or off remotely (i.e.without reporting to their home depot) at some agreed points and between certain hours. The first two exercises were each spread over about a month, but the last two, using completely new data and updated scheduling rules, were undertaken together within a period of two weeks. In all cases, TRACS II produced schedules which were recognised by the company as being workable and cheaper to operate than existing schedules.

It is worth commenting on the reaction of staff to the new scheduling system. During the 1970s when the use of computers in schedule construction was first discussed with bus operating authorities, there were substantial misgivings as to whether the drivers' unions would accept schedules that had been produced by computer. However, we do not know of any situation in which this posed a real problem in practice. There have been several instances in which the introduction of the computer system has enabled more driver-friendly schedule conditions to be introduced at little or no extra cost, and in which unions and management have been able to use the system together to develop new conditions acceptable to both. It is now generally accepted by drivers' unions that if a schedule meets all the requirements negotiated with management, then the way in which that schedule has been generated is irrelevant. Other staff directly affected are of course the schedulers themselves, and almost all schedulers whom we have met, including those who had no previous computer experience, have welcomed the introduction of the system. It has removed drudgery from their lives, and has given them more time to experiment with alternative operational practices.

8 Conclusions

Over a period of thirty-five years we have been developing driver scheduling methods in close association with bus and train operating companies. Since 1984 our systems based on combinations of heuristics and ILP have been in operational use in many locations. The TRACS II system consistently produces schedules which are at least as good as those produced manually, and normally considerably better. It is designed to be used on a PC, and has been proven to be flexible, and easily adaptable to a wide range of operating scenarios. It is in regular use by a wide range of bus companies, and has been applied by ourselves on behalf of rail companies, both to produce operational schedules and to explore 'what if' scenarios. The system has been readily accepted by schedulers, even where there has been no previous computing experience. The flexible approach which allows schedulers to influence the generation of potential shifts and the solution strategy has been a major factor in achieving this.

While the system produces excellent schedules, we are continually learning, both from our own experiences and from user feed-back, how it may be improved. Many of the current and imminent improvements are being made to details which are important to the user, but at a level which cannot be described within this paper.

We have referred in Section 4.7 to the generation of 'minimum change' schedules. Although facilities for creating these have been requested by users, these have not yet been used in many real situations, and we expect to improve the current facilities as we gain experience. At present we are dealing with planned changes, i.e. those which can be predicted several days or weeks in advance. Emergencies may of course arise in the course of operation, and although we are looking at ways of treating these, the full power of TRACS II would be inappropriate. We can learn from our scheduling experience how to develop quick and dirty approaches, but any automatic approach has to rely on the drivers' acceptance of new workings which may extend the working day and interfere with leisure activities. It is probable that any real-time system will rely on the data within the TRACS II system, and will generate a number of possible quick responses which may be discussed with the staff involved. We do not envisage any fully automatic real-time system.

9 Acknowledgements

We are grateful to many people within the bus and rail industries for their help over many years. In particular, Michael Meilton of First has steered our implementations in that Group, and has provided us with guidance and friendship. We are grateful to the UK Engineering and Physical Sciences Research Council for substantial financial support, most recently under grants GR/M23205, K79024 and K07256.

References

[1] A Wren. Heuristics ancient and modern: transport scheduling through the ages. *Journal of Heuristics*, 4:87–100, 1998.

[2] Sarah Fores, Les Proll, and Anthony Wren. TRACS II: A hybrid IP/heuristic driver scheduling system for public transport. *Journal of the Operational Research Society*, 53:1093–1100, 2002.

[3] N Hamer and L Séguin. The HASTUS system: new algorithms and modules for the 90s. In M Desrochers and J-M Rousseau, editors, *Computer-Aided Transit Scheduling*, pages 17–29. Springer-Verlag, Berlin, 1992.

[4] LA Wolsey. *Integer Programming.* John Wiley & Sons, New York, 1998.

[5] C Barnhart, EL Johnson, GL Nemhauser, MWP Savelsbergh, and P Vance. Branch-and-price : column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998.

[6] RSK Kwan, ASK Kwan, and A Wren. Evolutionary driver scheduling with relief chains. *Evolutionary Computation*, 9:445–460, 2001.

[7] Barbara M Smith, Colin J Layfield, and Anthony Wren. A constraint programming pre-processor for a bus driver scheduling system. In EC Freuder and RJ Wallace, editors, *Constraint Programming and Discrete Optimization*, pages 131–148. American Mathematical Society, 2001.

[8] Yingdong Shen and Raymond S K Kwan. Tabu search for driver scheduling. In S Voss and JR Daduna, editors, *Computer-Aided Scheduling of Public Transport*, pages 121–135. Springer-Verlag, Berlin, 2001.

[9] BM Smith and A Wren. A bus crew scheduling system using a set covering formulation. *Transportation Research*, 22A:97–108, 1988.

[10] A Wren and BM Smith. Experiences with a crew scheduling system based on set covering. In JR Daduna and A Wren, editors, *Computer-Aided Transit Scheduling*, pages 104–118. Springer-Verlag, Berlin, 1988.

[11] M E Parker, A Wren, and R S K Kwan. Modelling the scheduling of train drivers. In JR Daduna, I Branco, and JMP Paixao, editors, *Computer-Aided Transit Scheduling*, pages 213–235. Springer-Verlag, Berlin, 1995.

[12] WP Willers, LG Proll, and A Wren. A dual strategy for solving the linear programming relaxation of a driver scheduling system. *Annals of Operations Research*, 58:519–531, 1995.

[13] Sarah Fores and Les Proll. Driver scheduling by integer linear programming - the TRACS II approach. In A El Kamel P Borne, M Ksouri, editor, *Proceedings CESA'98 Computational Engineering in Systems Applications, Volume 3 Symposium on Industrial and Manufacturing Systems.* LÚnion des Chercheurs et Ingenieurs Scientifiques (UCIS), Villeneuve d'Ascq, 1998.

[14] Sarah Fores, Les Proll, and Anthony Wren. An improved ILP system for driver scheduling. In NHM Wilson, editor, *Computer-Aided Transit Scheduling*, pages 43–62. Springer Verlag, Berlin, 1999.

[15] ASK Kwan, RSK Kwan, ME Parker, and A Wren. Producing train driver schedules under different operating strategies. In NHM Wilson, editor, *Computer-Aided Transit Scheduling*, pages 129–154. Springer-Verlag, Berlin, 1999.

[16] R E Bixby, J W Gregory, I L Lustig, R J Marsten, and D F Shanno. Very large-scale linear programming: a case study in combining interior point and simplec methods. *Operations Research*, 40:885–897, 1992.

[17] JJ Forrest and D Goldfarb. Steepest-edge simplex algorithms for linear programming. *Mathematical Programming*, 57:341–374, 1992.

[18] M Chamberlain and A Wren. Developments and recent experience with the BUSMAN and BUSMAN II systems. In M Desrochers and J-M Rousseau, editors, *Computer-Aided Transit Scheduling*, pages 1–15. Springer-Verlag, Berlin, 1992.

[19] A S K Kwan, R S K Kwan, M E Parker, and A Wren. Producing train driver shifts by computer. In J Allan, CA Brebbia, RJ Hill, G Sciutto, and S Sone, editors, *Computer in Railways V*, volume 1:Railway Systems and Management, pages 421–435. Computational Mechanics Publications, 1996.

[20] M Desrochers, J Gilbert, M Sauvé, and F Soumis. CREW-OPT: subproblem modelling in a column generation approach to urban crew scheduling. In M Desrochers and J-M Rousseau, editors, *Computer-Aided Transit Scheduling*, pages 395–406. Springer-Verlag, Berlin, 1992.

[21] A Caprara, M Fischetti, PL Guida, P Toth, and D Vigo. Solution of large-scale railway crew planning problems: the Italian experience. In NHM Wilson, editor, *Computer-Aided Transit Scheduling*, pages 1–18. Springer-Verlag, Berlin, 1999.

[22] Torsten Fahle, Ulrich Junker, Stefan E Karisch, Niklas Kohl, Meinolf Sellmann, and Bo Vaaben. Constraint programming based column generation for crew assignment. *Journal of Heuristics*, 8:59–82, 2002.

[23] Sarah Fores, Les Proll, and Anthony Wren. Experiences with a flexible driver scheduler. In S Voss and JR Daduna, editors, *Computer-Aided Scheduling of Public Transport*, pages 137–152. Springer-Verlag, Berlin, 2001.

[24] A Wren and RSK Kwan. Installing an urban transport scheduling system. *Journal of Scheduling*, 2:3–17, 1999.

[25] M Meilton. Selecting and implementing a computer aided scheduling system for a large bus company. In S Voss and JR Daduna, editors, *Computer-Aided Scheduling of Public Transport*, pages 203–214. Springer-Verlag, Berlin, 2001.