University of Leeds

SCHOOL OF COMPUTER STUDIES

RESEARCH REPORT SERIES

Report 96.28

Heuristics Ancient and Modern : Transport Scheduling Through the Ages

by

Anthony Wren

Division of Operational Research & Information Systems

September 1996

Abstract

Heuristics developed over thirty-five years for transport scheduling are presented. These have been generated in response to requirements to solve practical problems, and most are now in regular use by bus and train companies. Mathematical programming models have been formulated for some of the problems, but have been inappropriate on their own; in some cases, heuristics have led to a reduced problem which has then been solved by integer linear programming. The paper explores the relationships between heuristics and other approaches and emphasises the need to convince users of the suitability of the overall system. Where appropriate, indications are given of difficulties in achieving practical implementation

Keywords: Scheduling, transport, integer linear programming, heuristics, implementation

The author has been designing and using heuristics to solve practical transport scheduling problems since 1961. Throughout this period he has been driven to produce software which meets one over-riding objective:

- It must be better *in the view of the end user or decision maker* than other available ways of solving the problem for which it is designed.

With this in mind, powerful heuristics have frequently been the obvious tools for a particular problem, and some of the more recent arguments about the relative merits of artificial intelligence and operational research approaches seem misplaced. For example, Grant (1986) defines OR as a discipline which is only interested in optimality, and claims that AI methods are superior because they seek practical, possibly sub-optimal solutions. Yet the paradigm displayed above is essentially an OR paradigm which has driven the development of heuristics within the OR community for many years.

In this paper the design of practical methods of scheduling transport is traced in broad chronological order through the author's own OR-based experience, with reference to related work elsewhere. Heuristics are initially seen as stop-gaps until mathematical programming, generally integer linear programming (ILP), should be powerful enough to replace them. However, the ILP processes have as yet been adequate in their own right only for relatively small problems in these fields. Heuristics have therefore been developed to assist these ILP processes, and the combination has led to successful solution of large and complex problems.

Barr et al. (1995) and Hooker (1995) have recently presented cogent arguments on the design of tests for heuristics. Hooker writes that "typically an investigator has a bright idea for a new algorithm and wants to show that it works better, in some sense, than known algorithms". Much OR however starts with the need to solve a particular problem. New heuristics or other approaches may be devised for the problem, and may subsequently be compared to other algorithms, but often there are no other algorithms for the problem being solved. Where such algorithms have been published, the problem instances are often unavailable for testing. This is particularly true in scheduling problems of all types, where very specific conditions may exist. Methods outlined here therefore have not usually been tested against other algorithms.

The paradigm set out above implies that any algorithm, be it a heuristic, a mathematical process, or a combination of these, may be superseded if something perceived to be better is presented to the user or to the user's managers. This fact drives searches for ever better methods, and some of these are indicated in Sections 7 and 9. Decision makers, however, are often unimpressed by what are seen as academic arguments in favour of one algorithm or another, and must be persuaded by practical and user-friendly demonstrations.

Throughout the period covered here there has been a conflict between the user who understands the need for a powerful tool to solve the problem in hand, and the senior manager to whom the gloss that surrounds the methodology, or the skill of the salesperson, has been more important than the quality of the solution obtained. This is indicated in the following text, because it is desired to emphasise that quality of algorithm alone is not sufficient to achieve implementation. The methods presented in Sections 4 through 6 are incorporated in the BUSMAN system (Wren and Chamberlain, 1988), in widespread commercial use, while Section 8 outlines a train driver scheduling system which is now solving practical problems.

1. Rail locomotive scheduling (1961-1964)

The simplest view of a locomotive scheduling problem is that a set of locomotives of a single class is to be allocated to a set of trains in such a way as to minimise the number used, and within that to minimise the amount of light running (locomotives running without trains). The problem can be modelled in mathematical terms as a standard Assignment Problem with a row and column for each train, and solved by the Hungarian algorithm. However, the number of trains in our first practical examples ranged up to 150, requiring cost matrices of 22.5K, well beyond our 1960 Pegasus computer's capacity of 7K words for data and code together.

A heuristic was therefore devised which had the merit of producing the optimal solution in all problems small enough to be checked against the Hungarian algorithm. In the thirty-five years since the initial research, it has been applied to many thousand real and often large rail and bus scheduling problems of the above form without being bettered. (However, one may construct artificial cases where the heuristic terminates in a local optimum if which the constraint matrix of minimum journey times between points violates the triangular inequality.

A strong lower bound to the required number of locomotives can easily be calculated as the maximum number of simultaneously operating trains, where two trains are effectively simultaneous if the locomotive used for the earlier finishing one cannot reach the starting point of any other train until after the departure of the second. This bound mechanism was presented by the author to Stern and Ceder (1983) who proved its validity. A (possibly infeasible) solution is developed using this bound as the number of locomotives; trains are taken in order of departure time and are assigned free locomotives rather arbitrarily; if no locomotive is free, then some locomotive is allocated infeasibly, and a penalty is assigned equal to the time needed to move the locomotive to the relevant point, minus the time available (which may be negative). The result is a linked list of activities (trains) for each locomotive, where each link has either a cost equal to the time necessary to move between points or a penalty as above.

The heuristic examines all pairs of links, exchanging their end points if this produces a reduction in total penalty, or a reduction in total cost for no increase in penalty. A neutral exchange is made if no change in total cost or penalty results; this may create a new pattern which can allow the heuristic to escape from a local optimum at the next pass. The process stops when two complete passes produce no improvement.

Many experiments were carried out with different starting solutions, including some in which very large infeasibilities were present. All experiments, however bad the initial solution, converged for a given data set to the same total penalty and total cost, apparently the global optimum. In particular, where a feasible solution was known to exist with the given number of locomotives, such a solution was always found.

Where a penalty remained after convergence, the original approach was to increase the number of locomotives by one and to try again. It was soon realised, however, that it was worth presenting the penalised links to the client, who could often retime a train in order to remove the penalty. In the first practical exercise implemented for British Railways (BR) in 1963, the number of locomotives was reduced from 15 to 12 (after two accepted retimings) and the light running was reduced by 28% (Wolfenden and Wren, 1966).

The above had involved a set of freight trains running day and night, ferrying traffic around London, principally between the docks and marshalling yards. The solution had to be adjusted to allow the locomotives to be returned from time to time to the depot for maintenance. This adjustment, which was carried out manually by the author, followed another heuristic designed by him, which could not easily be coded within the computer's capacity. This preserved the savings, although three additional locomotives were required to allow for maintenance, both in the "before" and "after" situations.

It may be noted that the switching procedure is equivalent to pairwise swapping of the elements of a solution matrix for the assignment problem if the penalties are given precedence in the cost matrix. A major difficulty in having the process accepted by our peers in 1963 was the lack of a proof of optimality. The term "heuristic" was unknown to them and the author was forced to use the term "trial and error" in presenting the process. By the time heuristics had been made respectable, for example in Mueller-Merbach's classification (1974), the method would have been recognised as "eager but tedious".

The system was used by BR's OR department for several years on many parts of the network with considerable success. However, it appeared obvious to management that if a scheduling system could be developed within a university, then a much better system could be developed by BR engineers. The process was therefore abandoned when the Pegasus computer became obsolete, and an in-house system was developed over many years, but never implemented. This was outlined by Wren (1971) with the approval of its developers. BR still did not have an automatic locomotive scheduling system in use when the organisation was split up in 1994, and the author has been asked by its successors to provide such a system!.

2. Delivery vehicle routeing (1966-1970)

The delivery vehicle routeing problem (VRP) has attracted much attention in the OR community since the late 1960's. It was originally formulated as the problem of designing routes for vehicles from a single depot subject to constraints of weight and distance. The first practical solution method was probably the heuristic of Clarke and Wright (1964) which step-wise built up a feasible solution, and this formed the basis of several commercial routeing packages. Later, complicating factors such as many depots, different vehicle types, time windows for certain deliveries, etc. were introduced, and it became evident that fully automatic systems were seldom suitable for solution of practical day to day problems. Several authors, for example Laporte (1991), have surveyed solution methods. Although mathematical programming formulations are possible, these are much more complex than those of locomotive scheduling (but see Foster and Ryan (1976) for an early approach using heuristics to reduce the size of a problem which was then solved using integer linear programming).

Wren and Holliday (1972) designed a multi-depot system which out-performed existing single depot systems on common trial data sets, and which was used to assist in strategic decision making (e.g., depot siting, creating regular route patterns) by several major companies. This commences by building routes using a sweep algorithm as later proposed by Gillett and Miller (1974). A restriction may be placed on the number of vehicles available, in which case some points may initially be excluded. A number of refining heuristics are then applied, principally to reduce costs, although some try to reduce the number of routes used, and others seek to fit points that were omitted initially, possibly replacing others of lower priority. The cost reduction routines each perform a different action: moving points from one route to another; exchanging points between two routes; removing crossovers from routes. A special sub-program applies the sweep process and some of the cost reduction routines to the points in adjacent pairs of routes; the new route structures are kept if they are better.

The refining heuristics each search all points or routes, and are called cyclically in turn. Each individual heuristic is terminated when one complete pass produces no change, or two successive passes produce only neutral changes; the cycle of routines is terminated according to the same rules, or when a predefined processing time limit is reached. Thus each call of each heuristic finds a local optimum. Moving on to another heuristic corresponds to changing the shape of the search space. Overall, the results compared favourably with other heuristics then known. However, resources were not available to develop a full commercial system.

3. Heuristics for bus driver scheduling (1967-1978)

That the bus driver scheduling problem could be formulated as a set covering or set partitioning problem was known from the early 1960's. However, no practical solution could be obtained from such a formulation until much later. Throughout the period 1967-1978 the author and his associates devoted much attention to seeking a good heuristic method (Wren, 1971; B. Manington and Wren, 1975), culminating in the TRACS system (Parker and Smith, 1981). Others (e.g., Elias, (1964)) also devised heuristics throughout the 1960's and 1970's; several of the more successful may be found in Wren (1981a).

Bus driver scheduling is the problem of designing legal shifts so that all scheduled buses have a driver assigned at all times of day. Shifts have constraints on their duration, on the position of meal breaks and on other features which may be negotiated locally. Drivers can only be changed at *relief opportunities* when buses pass suitable points. Most shifts consist of two stretches of work separated by a meal break. While each stretch would ideally consist of work on a single bus, it is often necessary in order to fit all the bus work together that a stretch should contain more than one *spell*, on different buses separated by small paid breaks. Wren (1981b) and Wren and Rousseau (1995) have surveyed computer methods for bus driver scheduling, many of which are now in regular use by bus companies.

When first presented with this problem the author hoped that a similar approach to that used in locomotive scheduling could be devised, i.e., that one might construct an infeasible schedule and then refine this until eventually a good feasible schedule was obtained. Unfortunately in removing the infeasibilities the schedule was forced to become progressively more fragmented, introducing wasteful gaps between spells of work, and thus requiring too many shifts.

Attempts to devise heuristics to escape from the resultant local optimum were unsuccessful. These consisted of routines which exchanged work between shifts in several ways. While the refining routines were often (indeed usually) successful in improving existing manual schedules, it became evident that they were not capable of moving from a poor schedule, such was initially constructed, to a good one. The conclusion that it was unlikely that heuristics could move from a poor solution to a good one was borne out by observation of manual schedulers; when a trainee constructed a schedule it would be checked by a senior colleague; unless the first attempt was reasonably good, the senior scheduler would find it easier to start again than to adopt the poor schedule as a basis for adjustment.

Much time was therefore spent in improving the starting routine, and several approaches were designed. Examination of the problem structure enabled the team to identify certain critical features which would lead to inefficiency unless properly addressed, and by about 1974 we had designed a process which would produce reasonably good initial schedules for three sample bus operators. The system (known as TRACS) incorporated techniques gleaned from human schedulers, and would nowadays be described as an expert system, both knowledge-based and rule-based. The initial schedules were refined by the heuristics designed several years earlier, resulting in solutions which were at least as good as manually produced ones. Although it had proved impossible to get schedulers to explain the processes they used to construct schedules, they generally accepted that the approach adopted was similar to their own.

Over the next few years feasibility studies were carried out for some dozen bus operators. The constraints on schedule construction (shift length, etc.) vary greatly between companies, as do the "shapes" of underlying bus schedules. Initial experience indicated that while the heuristics behind the starting solution had to be adapted to deal with new situations, these adaptations could be designed and accommodated within a developing general method after about three person-months of research. We were therefore hopeful that we should ultimately develop a general starting heuristic which would be applicable with minor, if any, modifications to new situations.

During this time, several organisations had implemented schedules produced by the system, and one had used the programs to experiment in conjunction with the drivers' union with many different sets of possible constraints to help design a new mutually acceptable set of rules. This last company entered negotiations to buy the system; these broke down when its in-house computer group objected to operating a system which could not guarantee an optimal solution, despite its consistently achieving better schedules than those produced manually.

The last three organisations for whom tests were undertaken upset the earlier confidence in the system, and no acceptable schedule was obtained within the budgeted three months. In Glasgow the working day was longer than met elsewhere, and a desired percentage of "early" shifts was not achieved. In Dublin the times of the peaks were much later than elsewhere, preventing the heuristic which dealt with the morning peak from working satisfactorily. In London the bus schedule had always been carefully designed to accommodate a good driver schedule, and there was generally only one acceptable solution, which the heuristics could not find. We therefore initiated in 1978 a search for a different solution method (see Section 6). It may be remarked that the resulting new system is now used consistently in London and Glasgow, while the Dublin company is using a system believed to be based on a similar model.

4. Bus scheduling (1970-1984)

In 1970 the locomotive scheduling system was adapted under the name VAMPIRES to bus operation, first for a single bus type and single depot. The first trial was carried out for Yorkshire Traction in Barnsley where 23 buses were saved out of 107, partly by improved linking of journeys, and partly by retiming or otherwise adjusting journeys where the system indicated infeasibilities (Wren, 1972). Yorkshire Traction subsequently sponsored developments to allow more than one depot and more than one type of bus.

The multi-depot problem was handled by extending the exchanging concept of Section 1. An initial solution was formed without reference to depot, and vehicles were then allocated to the depot requiring the least empty running to the first journey departure and from the last arrival. When investigating a potential exchange, the best depots for the revised workings were determined on the same basis, and the exchange was accepted if the total costs of journeys to and from the depots and of the links being considered were not worsened. The system was later extended to treat the return of buses to their home depot in the course of the day by assigning appropriate negative costs to links having large idle times. It may be noted that the assignment problem approach which was rejected ten years earlier on the grounds of machine capacity could not have been extended in these ways, as the Hungarian algorithm can only treat the immediate costs of a link, and not the knock-on effect of a change on the rest of the bus working. It will be recalled that the original heuristic did not guarantee, but probably always achieved, optimality. Little can be said about the optimality of the extensions, as the problems were much larger. Bus companies were, however, satisfied with the results, and the extended system therefore fitted our original paradigm quoted at the head of this paper.

The treatment of more than one vehicle type is too complex to detail here. The first tests were carried out on a problem involving nine basic types, any journey having a range of preferred types. A multi-stage process was devised, repeatedly using a classical Transportation Problem to determine shadow costs which guided the heuristics. Although the results of this process were satisfactory where there were genuinely many types of vehicle, the increase in computer time was very great, and situations arose in which lack of optimality could be demonstrated. Shortly after this development, there was a movement in the UK towards bus fleets consisting of a single vehicle type, and the multi-vehicle option has not been used for twenty years.

Practical exercises for several other companies followed (P.D. Manington and Wren, 1975; Smith and Wren, 1981). The first regular user was Greater Manchester Transport who installed the program in 1975. Buses are now scheduled in smaller groups in response to competition, but the program is still used occasionally in Manchester.

Although VAMPIRES was used throughout the 1970’s, with considerable savings, its ability to explore all possible combinations of journey across a region was superfluous when routes were scheduled in relatively small groups, and operations were regular, so that the normal activity for a bus on reaching its destination was to turn and go back to its origin. A simpler system, TASC, was therefore developed in which dead (empty) journeys were only considered if an arriving bus could not match a departure from the same point within a specified time.

TASC (Smith and Wren, 1981) included a backtracking process to try to ensure that where dead journeys were necessary, the most efficient set was used, and was installed for several bus operators in the early 1980's. It was easier to use than VAMPIRES, the data being specified as series of regular journeys on several routes, rather than as individual journeys, and it was initially faster. However, users started to complain that sometimes the solution was not optimal, and the backtracking process was gradually made more complex to satisfy their complaints, adding to the computer time. A simplified version of VAMPIRES was therefore combined with TASC; a preliminary solution was found by TASC, and divided into blocks of continuous work with no long gaps or dead journeys; these blocks were then rescheduled by VAMPIRES as if they were individual journeys.

In 1983 the decision was made to convert the system for microcomputers, and after various trials were conducted it was determined that TASC had become so complex that VAMPIRES was now faster. Therefore the first microcomputer version was based on a simple version of VAMPIRES. This has since been made more complex; for example, a crude but fast method for dealing with more than one vehicle type was added at the request of a client; this incorporates a three-way swapping procedure. This version was installed in some forty bus companies, and is still in commercial use, although it has been suggested that it has become too complex, and that TASC should be resurrected, so that simple problems can be solved speedily! More recently an object-oriented version of the approach has been implemented; this will ultimately allow modular choices to be made by the users.

5. Bus driver rostering (1978-1985)

Mention should be made of driver rostering, where the daily shifts are grouped into rows of weeks or fortnights according to given rules. In many countries drivers then rotate through cycles of rows. The cost depends on the extent to which individual rows exceed a target amount of work. Bennett and Potts (1968) in Adelaide tackled this by considering a matrix in which rows represented fourteen days of work (early week followed by late week) and the elements identified the shifts to be worked on the appropriate days. The first phase of their process distributed rest days through the rows in such a way as to maximise the number of four-day weekends. An initial solution was fitted to this and refined by a heuristic which kept thirteen columns fixed and reassigned the elements of the remaining column optimally using the Hungarian algorithm. Each column was reassigned in turn and the process was repeated until no improvement was produced.

The author and colleagues adapted this process for British problems, replacing the Hungarian algorithm by a heuristic which simply exchanged elements in the column being considered. This was not implemented, as it proved impossible to determine from users when apparent constraints could be violated. Subsequently semi-automatic systems incorporating some of these principles have been implemented for several bus companies.

6. Mathematical programming based methods for bus driver scheduling (1978-1986)

Advances in computer power led us in 1978 to examine again the possibility of basing a driver scheduling system on ILP. Smith and Wren (1988) describe a new system which was first installed for London Transport Buses in 1984 and for many other organisations later. The basis is the generation of many thousand potential shifts followed by a specialised ILP process to select a subset which forms a good schedule; refining heuristics similar to those used earlier modify the solution, and are able to restore shifts rejected by earlier processes.

Heuristics restrict the size of the generated set and drive the branch and bound part of the ILP. These are too complex to describe here, and readers are referred to Smith and Wren (1988) and to Smith (1988) for further details. It is sufficient to say that the generation process is restricted by two sets of heuristics based on knowledge of the problem structure. It has been observed that a good integer solution can be found using only the relief opportunities in the continuous relaxation and usually with the number of shifts indicated by the relaxation; the ILP takes advantage of this in a specialised branch and bound algorithm. The original system (IMPACS) is fully described in a thesis by Smith (1986).

IMPACS is now installed for about thirty transport organisations, including bus, tram and light rail systems. A later version (TRACS II) is used in research, and has been extended for heavy rail (see Section 8). It can solve directly problems which are of a size to satisfy most British companies, who tend to schedule relatively small operating units separately. (Up to about 150 shifts are regularly solved satisfactorily.) However, some organisations do have larger problems, and a decomposition heuristic used by TRACS II for large bus problems is described by Wren and Smith (1988). This analyses the bus schedule and forms sub-problems consisting of buses which fit well together in a driver schedule; the work of inefficient shifts is carried forward from one sub-problem to the next. The heuristic has had very good results in practice, always beating manually produced schedules and those produced by rival systems in test cases. However, users are suspicious of decomposition, seeing its potential drawbacks and believing wrongly that a system which attempts to tackle the whole problem simultaneously must be better. In fact, systems investigated which solve large problems in a single pass depend on heuristics which appear to yield poorer results than this intelligent decomposition.

Other systems which use mixtures of heuristics and mathematical programming are HASTUS developed in the University of Montreal (Rousseau and Blais, 1985) and HOT implemented by Hamburger Hochbahn (Daduna and Mojsilovic, 1988). The first is in widespread use, while the second has been installed in Germany and France.

7. Knowledge-based approaches to bus driver scheduling (1988-1993)

Although TRACS II performs very well on most types of problem, there have been cases where users have encountered difficulties of robustness, especially on larger problem instances, and it would be desirable to find a faster and more stable methodology which might replace some or all of TRACS II. A number of new approaches have been explored (see also Sections 8 and 9). Here we describe one by Zhao, Wren and Kwan(1995).

This knowledge-based approach commences by estimating the number of shifts required in a good schedule, the estimator producing a strong lower bound. It was originally intended to build up a schedule, adding shifts one at a time. After each addition, the residual of unscheduled work would be submitted to the estimator. If the new estimate plus the number of shifts in the developing schedule were equal to the original estimate, then the process would be deemed to be on course. Otherwise the system would backtrack to an earlier decision point and add a different shift. During the development of the system the rules of the estimator (which took into account both the labour conditions and the precise vehicle schedule) were examined and improved as appropriate.

In practice, the coded estimator became very good, and we concentrated on improving this still further, rather than incorporating the additional processes above. The estimator follows in broad terms the rule-based process devised to assist the decomposition of large problems in TRACS II. It examines critical parts of the bus schedule and determines how difficulties may be resolved, building up partial or full shifts which address the perceived problem. Finally, it produces a skeleton schedule which hopefully has the correct number of shifts, although the precise way in which some work is to be included may not have been determined. The current estimator has been tested on seven problems, agreeing with the number of shifts used by IMPACS in each case. The largest problem, with about eighty shifts, was tackled in about a minute on a 386 machine (c.f. about two hours for IMPACS). In fact the system consists largely, but not by design, of a formal encapsulation of the heuristics developed in the 1970's. It is hoped that resources may be found to develop the estimator further to show alternative ways of surmounting some problems, and to highlight those relief opportunities which may be critical. This set of critical relief opportunities should be much smaller than that currently used in TRACS II, and should lead to results of large problems being closer to optimal.

8. Rail driver scheduling (1990-1991, 1994-1996)

We first outline an exercise undertaken over the winter of 1990-91 in which driver schedules were modelled for British Rail, who wished to experiment with alternative definitions of legal or acceptable shifts and to progress to a new agreement with drivers' unions. Train driver scheduling in the UK is considerably more complex than bus driver scheduling, but the accuracy required of the model was such that simplifying assumptions were acceptable. Initially it was hoped to modify the estimator above. However, the changes which BR wished to make would have completely altered the character of some shifts, and it was feared that the rules developed for bus drivers might not present a fair representation of the resultant situation.

Accordingly, modifications were made to the shift generation rules of TRACS II, which was then allowed to proceed to the optimum solution of the relaxed continuous LP. The rules used by TRACS II to reject unlikely relief opportunities were too specific to bus scheduling to be used here, and could not be adjusted within the very short time available for the exercise. This meant that the branch and bound process would have had to be applied to larger problems than usual. However, experience in bus driver scheduling had shown that the number of shifts in the optimal integer solution was almost always equal to that in the continuous solution, rounded up if necessary. In order to satisfy BR, some smaller problems were run through to an integer solution which proved to local managers that the model could lead to an acceptable solution.

A number of heuristics were devised to assist BR in estimating not only the numbers of shifts, but their general forms. As good estimates were obtained, so BR expected ever better fits from the model, which after appropriate adjustments always predicted correctly the number of shifts in cases where the solution was known. The model was deemed to be successful, and was used in-house by BR in the context of a major restructuring exercise (Wren, Kwan and Parker, 1994).

Since 1994 TRACS II has been further extended to solve rail driver scheduling problems. Many detailed adjustments to the heuristics have been made, for example, to allow shifts on more than three different train sets, to provide facilities for drivers to travel as passengers on timetabled trains of the company being scheduled, or of its rivals, and to deal with many depots and different types of traction simultaneously. The system is now being used commercially, and achieving significant savings (Kwan et al., 1996)

It may be remarked, however, that while a demonstration for one major company achieved a significant saving, accepted by the schedules manager, he was overruled in his efforts to adopt TRACS II. Senior management instead commissioned from a major consultant the design of a new system which technical staff believe will not meet their needs. Thus immediate availability of a working system has been sacrificed for a glossy but uncertain future.

TRACS II is applicable to bus, tram and train. It is being further developed to improve the heuristics, while Willers (1995) and Fores (1996) have made significant advances in the mathematical programming components. However, driven by our basic paradigm, further research (see below) is proceeding to investigate various metaheuristic approaches.

9. Metaheuristics for bus and rail driver scheduling (1990-onwards)

Wren and Wren (1995) describe a feasibility study of a genetic algorithm in which an optimal solution was found to a small driver scheduling problem of fifteen shifts. The GA starts with an initial population of schedules formed by selecting shifts at random from the set generated by TRACS II, and if successful would replace the ILP component. However, the GA has not yet found good solutions to larger problems, and improvements are currently being sought.

Kwan and Wren (1996) are experimenting with another approach in which a GA is used to determine relief opportunities to be rejected in TRACS II or in any other system. A number of random subsets of relief opportunities provide an initial population. The fitness of a subset is a combination of the estimate obtained by the process of Section 7, and the number of opportunities in a subset. Mating of population members proceeds in a suitable way until hopefully a much reduced set of relief opportunities is obtained which yields the same estimate as the full set. It should be possible to generate from these a good set of shifts to provide the basis either for the ILP of TRACS II or for a GA approach as outlined above. As the set of relief opportunities obtained in this way should be much smaller than that currently used in TRACS II it should be possible to solve much larger problems, and initial investigations have proved encouraging.

Under a new major research contract we are investigating constraint programming driven by heuristics to solve various parts of the whole problem. In parallel, tabu search and ant systems (Dorigo, 1996) are being developed as alternative means of solving some or all of the problem.

10. Conclusions

Driven by the need to obtain acceptable methods for practical problems, a number of heuristics has been applied to a range of transport scheduling problems over many years, with considerable success, except in the case of driver scheduling where the topography of the search space has led to poor local optima. In driver scheduling, heuristics have been used in conjunction with mathematical programming processes to produce a widely used system. Recently, formal encapsulation of knowledge has assisted the development of what may prove to be better heuristics, and there is reason to believe that genetic algorithms or other metaheuristics may lead to more robust solution methods for larger problems. Skilful selling to those with purchasing power is often as important as finding the method which produces the best schedule.

References

Barr, Richard S., Bruce L. Golden, James P. Kelly, Mauricio G.C. Resende and William R. Stewart (1995). "Designing and reporting on computational experiments with heuristic methods," *Journal of Heuristics* 1, 9-32.

Bennett, Brian T. and Renfrey B. Potts (1968). "Rotating roster for a transit system." *Transpn.Sci.* 2, 14-34.

Daduna, Joachim R and Miodrag Mojsilovic (1988)., "Computer-aided vehicle and duty scheduling using the HOT programme system." In Joachim R. Daduna and Anthony Wren (eds.) *Computer-Aided Transit Scheduling*. Berlin: Springer-Verlag,133-146.

Dorigo, Marco, V. Maniezo and A. Colorni (1996). "The ant system: optimization by a colony of cooperating agents." *IEEE Trans. on Systems, Man and Cybernetics*, B26, 1-13.

Elias, Samy E.G. (1964). "The use of digital computers in the economic scheduling for both man and machine in public transportation. *Kansas State University Bulletin, Special Report* 49.

Fores, Sarah (1996). "Column generation approaches to bus driver scheduling." *University of Leeds PhD Thesis.*

Foster, Brian A. and David M. Ryan (1976). "An integer programming approach to the vehicle scheduling problem," *Opl.Res.Q.* 27, 367-384.

Gillett, B.C. and L.R. Miller (1974). "A heuristic algorithm for the vehicle dispatch problem," *Opns.Res.* 22, 340-349.

Grant, T.J. (1986). "Lessons for O.R. from A.I.: a scheduling case study," *J. Opl Res Soc*, 37, 41-57.

Hooker, J.N. (1995). "Testing heuristics: we have it all wrong," *Journal of Heuristics* 1, 33-42.

Kwan, Ann S.K., Raymond S.K. Kwan, Margaret E. Parker and Anthony Wren (1996). Producing train driver shifts by computer. In: J. Allen, C.A.Brebbia, R.J. Hill, G. Sciutto and S. Sone (eds.). *Computers in Railways V, Vol.1: Railway Systems Management,* Southampton: Computational Mechanics Publications, 359-370.

Kwan, Raymond S.K.and Anthony Wren (1996). "Hybrid genetic algorithms for bus driver scheduling." *To appear in* L. Bianco and P. Toth (eds.) Advanced methods in transportation analysis, Springer-Verlag.

Laporte, Gilbert (1991). "The vehicle routing problem: an overview of exact and approximate algorithms." *Université de Montréal Centre de Recherche sur les Transports, Publication # 745.*

Manington, Barbara and Anthony Wren, (1975). "A general computer method for bus crew scheduling." *Pre-prints of International Workshop on Urban Passenger Vehicle and Crew Scheduling,* Chicago (49 pp).

Manington, Philip D. and Anthony Wren, (1975). "Experiences with a bus scheduling algorithm which saves vehicles." *Pre-prints of International Workshop on Urban Passenger Vehicle and Crew Scheduling,* Chicago (25 pp).

Mueller-Merbach, Heiner (1974). "Heuristic methods: structures, applications, computational experience." In R. Cottle & J. Krarup (eds.) *Optimisation Methods for Resource Allocation*, English Universities Press, 401-416.

Parker, Margaret E. and Barbara M. Smith (1981). "Two approaches to computer crew scheduling." In Anthony Wren (ed.) *Computer Scheduling of Public Transport*. Amsterdam: North-Holland, 193-222.

Rousseau, Jean-Marc and Jean-Yves Blais (1985). "HASTUS: an interactive system for buses and crew scheduling." In Jean-Marc Rousseau (ed.) *Computer Scheduling of Public Transport - 2*. Amsterdam: North-Holland, 45-60.

Smith, Barbara M. (1986). "Bus crew scheduling using mathematical programming." *University of Leeds PhD thesis*.

Smith, Barbara M. (1988). "IMPACS - a bus crew scheduling system using linear programming." *Math Prog*. 42, 181-187.

Smith, Barbara M. and Anthony Wren (1981). "VAMPIRES and TASC: two successfully applied bus scheduling programs." In Anthony Wren (ed.) *Computer scheduling of public transport*. Amsterdam: North-Holland, 97-124.

Smith, Barbara M. and Anthony Wren (1988). "A bus crew scheduling system using a set covering formulation," *Transpn.Res.* 22A, 97-108.

Stern, Helman I. and Avishai Ceder (1983). "An improved lower bound to the minimum fleet size problem," *Transpn.Sci.* 17, 471-477.

Willers, W. Philip (1995). "Improved algorithms for bus crew scheduling." *University of Leeds PhD thesis*.

Wolfenden, Keith and Anthony Wren (1966). "Locomotive scheduling by computer." In Proceedings of the British Joint Computer Conference, *IEE Conference publication* 19, 31-37.

Wren, Anthony (1971). *Computers in Transport Planning and Operation*. London: Ian Allen.

Wren, Anthony (1972). "Bus scheduling; an interactive computer method,". *Transportation Planning and Technology*, 1, 115-122.

Wren, Anthony (ed.) (1981a). *Computer Scheduling of Public Transport*. Amsterdam: North-Holland.

Wren, Anthony (1981b). "General review of the use of computers in scheduling buses and their crews." In Anthony Wren (ed.) *Computer Scheduling of Public Transport*. Amsterdam: North-Holland, 3-17.

Wren, Anthony and Mark Chamberlain (1988). "The development of Micro-BUSMAN: scheduling on micro-computers." In Joachim R. Daduna and Anthony Wren (eds.) *Computer-aided transit scheduling*, Berlin: Springer-Verlag, 160-174.

Wren, Anthony and Alan Holliday (1972). "Computer scheduling of vehicles from one or more depots to a number of delivery points," *Opl.Res.Q.* 23, 333-344.

Wren, Anthony, Raymond S.K. Kwan and Margaret E. Parker (1994). "Scheduling of rail driver duties." In T.K.S. Murthy et al. (eds.) *Computers in railways IV - Volume 2, Railway Operations,* Southampton, Computational Mechanics Publications, 81-89.

Wren, Anthony and Jean-Marc Rousseau (1995). "Bus driver scheduling - an overview." *In* J.R. Daduna and J. Paixao (eds.) *Computer Aided Transit Scheduling - 3,* Springer-Verlag, pp.173-187.

Wren, Anthony and Barbara M. Smith (1988). "Experiences with a crew scheduling system based on set covering." In Joachim R. Daduna and Anthony Wren (eds.) *Computer-aided transit scheduling*. Berlin: Springer-Verlag, 104-118.

Wren, Anthony and David O. Wren (1995). "A genetic algorithm for public transport driver scheduling." *Computers Ops Res*. 22, 101-110.

Zhao, Liping, Anthony Wren and Raymond S.K. Kwan (1995). "Development of a driver duty estimator," *J Opl Res Soc.*, 46, 1102-1110.