(1)

Computer Scheduling of Public Transport
A. Wren (ed.)

AN INTEGER PROGRAMMING APPROACH TO SCHEDULING

D. M. Ryan and B. A. Foster

Department of TAM, University of Auckland, New Zealand,
and Operations Research Group, AERE, Harwell, U.K.

The formulation of crew scheduling as set-partitioning or set-covering integer programs usually gives rise to I.P. problems of excessive size and computational complexity. By imposing sensible additional structure not inherent in the I.P. model, overconstrained I.P. subproblems can be formed. Optimal solutions to these subproblems can be found using reasonable computing resources, since the additional structure has a strongly integerizing effect on the associated L.P. model. Some applications of this approach to a New Zealand metropolitan bus operation are presented.

1.1 INTRODUCTION

Set-partitioning models of mathematical programming and graph theory occur in many practical applications. Balas and Padberg (1), in an excellent survey paper on the set-partitioning problem, suggest that in terms of its applications, it would be more important than its close relation, the set-covering problem, and also the well-known travelling salesman problem. They give a comprehensive bibliography of applications, which includes problems of crew scheduling, vehicle routeing and scheduling, information retrieval, switching circuit design, stock cutting, portfolio analysis, plant and facility location and political districting.

Our interest in the set-partitioning problem is confined to problems of scheduling in which the model takes the form

$$\text{SPP: minimize} \quad Z = \underline{c}^T\underline{x}, \qquad \underline{x} \in \mathbb{R}^n \qquad \text{(i)}$$

$$\text{subject to} \quad A\underline{x} = \underline{e}, \qquad \underline{e}^T = (1,1,\ldots,1) \in \mathbb{R}^m \qquad \text{(ii)}$$

$$\text{and} \quad x_j = 0 \text{ or } 1, \qquad j = 1,\ldots,n \qquad \text{(iii)}$$

where A is an m x n matrix of zeroes and ones, and $\underline{c}$ is the objective coefficient vector. Each of the columns $\underline{a}_j$, j=1,...,n, of A represents a duty (or schedule) with an associated cost c_j, and the corresponding variable x_j can be thought of as the 'probability' that the j-th column is included in a solution. The j-th duty $\underline{a}_j$ has elements

$$a_{ij} = 1 \text{ if duty j performs task i}$$
$$= 0 \text{ otherwise.}$$

The constraints (ii) require that each of the m tasks (or resources) is performed (or covered) exactly once in any solution. (The set-covering model is obtained by replacing (ii) with the constraint system $A\underline{x} \geq \underline{e}$, which requires that each task is performed at least once.) An optimal solution of SPP is given by a subset of duties with $x_j = 1$, which together satisfy (ii) (i.e. perform the tasks) at minimal cost. The zero or one values of the components of $\underline{x}$ thus define a partition of the set of all duties, and a partition of the tasks.

Much research has been undertaken during the past decade, to develop both theoretical results and efficient methods for the solution of SPP, and related models, and, although significant progress has been made (see Balas and Padberg (1)), it is often an expensive, and sometimes prohibitive, task to find optimal solutions. This is especially true in the context of scheduling, where, typically, the total number of possible duties (and therefore variables) is extremely large, even though the number of constraints is relatively small. The usual method for solving the SPP involves the relaxation of the integer restrictions (iii), and their replacement by the weaker bound condition $\underline{x} \geq 0$, thus creating a relaxed SPP linear program. In scheduling applications, the solution of this linear program seldom produces a naturally integer solution, and often requires an excessive computational effort, first to solve the L.P., and then to resolve the fractionality. Despite these two problems of computational complexity, which are perhaps more peculiar to scheduling applications, Balas and Padberg (1) argue that the solution of a general SPP can often be found by carefully adding further constraints or cuts, to improve the integer properties of the SPP linear program. In this paper, we explore an alternative approach for overcoming the problems of computational complexity present in scheduling SPPs. We develop an 'optimal heuristic' methodology for reducing scheduling SPP models to realistically-sized near-integer linear programs, by carefully selecting a subset of duties from amongst all possible legal duties.

Because of the extremely large number of variables (i.e. duties) present in scheduling SPPs, it is essential to reduce the dimensionality by selecting a subset of the duties likely to contain the optimal solution (or, equivalently, to discard those duties unlikely to be found in an optimal solution) and solve the resulting reduced SPP. This selection process essentially attacks just the size aspect of the computational problem. If, however, some care is taken in the selection process, it is possible to improve simultaneously the natural integer properties of the reduced SPP linear program, without seriously affecting optimality. In other words, we attempt to construct a small SPP which contains the optimal solution of the full SPP, and which also has a relaxed linear program with a significant number of the vertices (i.e. basic feasible solutions) integral. An application of this methodology is illustrated in Section 1.2, in the context of the vehicle scheduling problem.

In Section 2, we discuss some characterizations of integrality in SPP, and identify those structures or properties of duties which permit the formation or occurrence of fractional solutions in the relaxed linear program. These considerations lead, in Section 3, to a further application of the integerizing selection process discussed above. In this application, we consider a bus crew scheduling problem. We conclude by examining a constraint branching process, which, in the SPP context, provides a more effective branch and bound structure for the resolution of fractional solutions.

1.2 A VEHICLE SCHEDULING SPP

The vehicle scheduling problem has attracted a great deal of attention over the past fifteen years. The problem in its simplest form is to design a set of routes (i.e. duties) from a central depot to service n customers (i.e. perform tasks) at known locations, with a quantity q_i of some commodity such that all customers are visited, and any restrictions on vehicle capacity and route length are observed.

This problem is conveniently represented by the set-partitioning model (i), (ii) and (iii) in which each constraint of (ii) ensures that the corresponding delivery is perfomed, and the duties or columns of A can be thought of as possible or potential vehicle routes, contributing only to those constraints visited by the route. The constraints on vehicle capacity and route length do not appear explicitly in the model, but are used implicitly to determine legal routes (or variables) of the model. In any practical situation, one could expect more than one hundred customers (i.e. constraints) and many, many thousands of possible routes. Because of this obvious problem of computational complexity, most published methods for solving the vehicle scheduling problem (see Mole (10) for a survey of such methods) 'ignore' the set-covering model, and proceed on the basis of some heuristic method. Foster and Ryan (5), in contrast, have developed a method which reduces the complexity of the set-partitioning formulation by imposing further implicit overconstraints. The overconstraints essentially remove many of the variables from the original model, on the grounds that they have a structure which is most unlikely to be present in the routes of an optimal solution. In this way, it can be argued that optimality is unlikely to be seriously affected by solving the smaller problem.

The definition or concept of a good or likely vehicle route is obviously an important factor in the effectiveness of the approach. An examination of routes occurring in the optimal solutions of small test problems led to the identification of petal structure as a particularly attractive attribute of good routes. The petal concept has also been utilized in a number of heuristic methods for the solution of the vehicle scheduling problem (Gillett and Millar (6) and Gillett and Johnson (7)). A route with petal structure can be described as one which visits all customers in a particular sector of the region with centre at the depot. The routes of a petal solution radiate from the depot like petals of a flower, and do not overlap. Besides defining a very much smaller number of variables over which the SPP has to be solved, the petal concept has a further significant property. It has been shown (Foster and Ryan (5)) that all the extreme points of the feasible solution set of the petal SPP occur at integer points (i.e. all components of $\underline{x}$ are either 0 or 1). In other words, the solution of the relaxed petal SPP linear program also satisfies the integer restrictions on the variables. In summary, this approach has simultaneously reduced both aspects of the complexity problem - the number of variables has been reduced dramatically, and the integer restrictions have been removed altogether. The obvious question now concerns the effect on optimality of such a major restiction on the set of variables. It is easy to demonstrate that optimality is affected, and Foster and Ryan discuss ways in which the overconstraints can be relaxed in a controlled manner, to permit desirable non-petal routes, previously banned, to be considered by the L.P. In theory, such non-petal routes also enable fractional solutions to occur, but, in practice, the tendency for the L.P. to generate fractional solutions appears to be very slight. The petal approach of Foster and Ryan can be described as an 'optimal heuristic' in the sense that a heuristically reduced model is solved optimally. The heuristic is designed specifically to reduce complexity.

2. INTEGER STRUCTURES FOR SPP

The mathematical definitions of classes of zero/one matrices with every vertex integral have been discussed by Hoffman and Kruskal (8) (unimodularity), Berge (2) (balanced matrices) and Padberg (11) (perfect matrices). The relative strengths of these conditions are illustrated in Figure 2.1. Both balanced and perfect matrices may be defined in terms of forbidden submatrices which represent structures with fractioning capability.

A balanced matrix is most simply defined as a matrix containing no submatrix of odd order having row and column sums equal to two. The form of such submatrices is illustrated in Figure 2.2. The forbidden submatrices may be interpreted as being formed by a subset of variables, each contributing one link to a closed cycle without chords in the corresponding graph. We can clearly avoid the formation of the forbidden submatrices if we restrict the links (pairs of constraints to which a variable contributes) so as to prevent the closing of the cycle. This observation provides the motivation for a series of simple selection rules that can be used to extract a unimodular or balanced subproblem from the original SPP.

Figure 2.1 Strength of Integrality Conditions

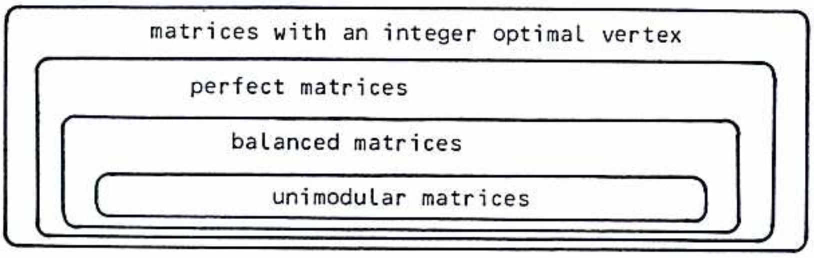
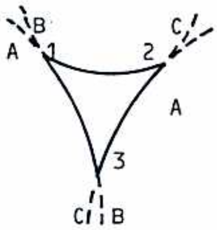


Figure 2.2 Forbidden Submatrices of Balanced Matrices

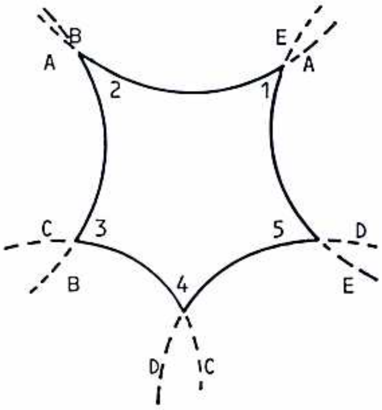
k = 3

	A	B	C
1.	1	1	0
2.	1	0	1
3.	0	1	1



k = 5

	A	B	C	D	E
1.	1	0	0	0	1
2.	1	1	0	0	0
3.	0	1	1	0	0
4.	0	0	1	1	0
5.	0	0	0	1	1

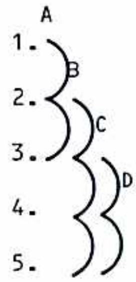


2.1 CONSTRAINT ORDERING (PETAL STRUCTURE)

We associate with each constraint a unique ordering index p_j, j=1,...,m. Variables are only considered as feasible for the SPP subproblem if they appear only in a contiguous subset of the constraints defined by (p_i, i=k,k+1,...,k+e), and appear in every member of such a subset. The reduced constraint matrix now takes the form given in Figure 2.3. Clearly, the closed cycles cannot be formed, the matrix is therefore balanced and all vertices are integral. This form of constraint ordering structure was applied by Foster and Ryan (5) to the vehicle scheduling problem where the ordering index is derived from the geographical radial ordering of the delivery locations around the depot. For the index to be unique, we must also nominate a natural break in the radial ordering (i.e. we must nominate constraint p_1).

Figure 2.3 Constraint Ordering Matrix and Graph Structure

	A	B	C	D	
1.	1	0	0	0	
2.	1	1	0	0	
3.	1	1	1	0	etc.
4.	0	1	1	1	
5.	0	1	0	1	



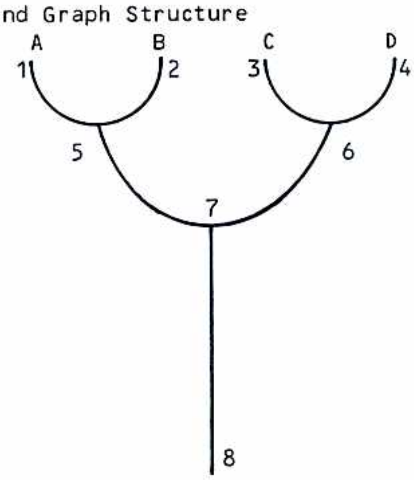
2.2 UNIQUE SUBSEQUENCE/PRECEDENCE

Again, we define a unique ordering of the constraints as p_j, $j=1,\ldots,m$. Further, with each constraint p_j, we associate a unique subsequent constraint $S(p_j)$, where $S(p_j)=p_k$ and $k > j$. Any variable contributing to constraint p_j must contribute to the subsequent constraint $S(p_j)$, or must not contribute to any constraint p_k for $k > j$. The form of the constraint matrix derived from this selection rule is illustrated in Figure 2.4. Again, it is easy to see that no closed cycles can form in the associated graph, and hence the constraint matrix is balanced. The unique subsequence structuring of variables has been applied by Edwards (3,4) to bus crew scheduling, and is discussed in detail in Section 3. The constraints are ordered by the start times of the corresponding trips, and each trip takes the first available subsequent trip to give the unique subsequent constraint.

Unique subsequence is a generalisation of unique ordering, and it has been shown by Edwards that unique subsequence structures are in fact unimodular, as well as balanced. By reversing the constraint ordering, we can also define an equivalent unique precedence selection rule.

Figure 2.4 Unique Subsequence Matrix and Graph Structure

	A	B	C	D	
1.	1	0	0	0	
2.	0	1	0	0	
3.	0	0	1	0	
4.	0	0	0	1	etc.
5.	1	1	0	0	
6.	0	0	1	1	
7.	1	1	1	1	
8.	1	1	1	1	



2.3 RELAXATION OF THE SELECTION RULES

These selection rules are far more restrictive than is necessary to satisfy the balanced matrix condition, which in turn is more restrictive than the desired condition of integer optimality. In our computational experience (see also Edwards (3,4)), significant relaxations can be made to these rules without creating many fractional vertices. For any specific problem, the selected variable set could certainly be extended by direct testing against the 'closed cycle without chords' condition. Further, if the constraint matrix can be decomposed into a number of non-interacting submatrices (i.e. a number of unconnected subgraphs), then a different subsequqnce (or precedence) ordering could be applied to each subgraph.

The formation of a closed cycle without chords in itself merely indicates a capability of producing a fractional vertex. If the constraint matrix also contains a constraint that prevents the formation occurring in a basic set, then the cycle can be permitted. For example, in Figure 2.5, we show part of a simple non-balanced matrix that is nevertheless totally integral because of the existence of constraint 4. A variable appears in constraint 4 if it appears in at least two other constraints. Constraints of this form were explicitly added to the vehicle scheduling SPP of Foster and Ryan, as integer forcing cuts. A generalisation of these cycle-breaking conditions is embodied in the definition of a perfect matrix.

An m x n zero/one matrix is perfect if no k x k submatrix K ($k > 3$) can be found to satisfy the conditions:

- the row and column sums of K are each equal to b ($b \geqslant 2$) (1)

and

- there exists no row of the $(m-k)xk$ submatrix formed by the constraints not included in K with a row sum greater than b. (2)

The conditions on the forbidden submatrix can be interpreted as preventing the wider class of cycles of the form (1) indicated in Figure 2.6, unless a constraint of the form (2) is also present.

The perfect matrix definition is more difficult to interpret directly as a set of variable selection rules. However, it can be seen that for problems where each variable must contribute to one, and only one, of a small band of constraints (e.g. each duty must contain one of a small number of evening peak trips), the resulting constraints will tend to be of type (2), and hence prevent the formation of fractioning cycles.

Figure 2.5 Cycle-breaking Cuts

	A	B	C
1.	1	1	0
2.	1	0	1
3.	0	1	1
4.	1	1	1

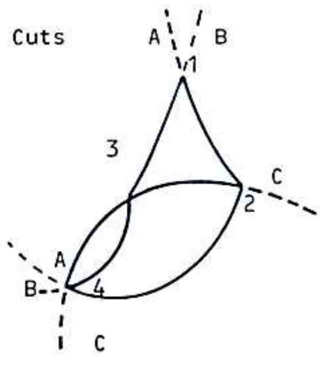
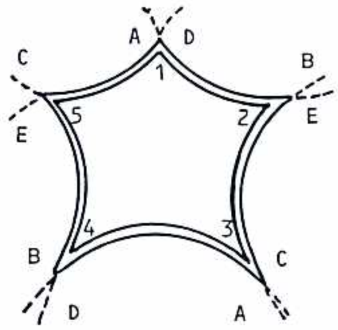


Figure 2.6 Forbidden Submatrices of Perfect Matrices

k = 5 : b = 3

	A	B	C	D	E
1.	1	0	0	1	1
2.	1	1	0	0	1
3.	1	1	1	0	0
4.	0	1	1	1	0
5.	0	0	1	1	1



Examination of the graphs associated with fractional optimal solutions occurring in the crew scheduling problem leads us to one final observation. In cases where it is unreasonable to prevent the formation of closed cycles, it is often possible to prevent these cycles being self-contained (i.e. further fractioning of variables is required to satisfy the constraints). If short duties (subsets of other duties) and slack variables are excluded, or discouraged, by high associated objective contributions, then this implies that further fractioning structures must appear elsewhere in the basic set to complete the partition. Clearly, the same completion argument applies to these structures also. We would hypothesise that it is often this difficulty in completing a partition with such fractional structures that causes a high incidence of integer solutions, especially near the optimum, even when the variable selection rule of unique subsequence is only strictly enforced in sections of the duty period.

3. BUS CREW SCHEDULING

The Auckland Regional Authority (ARA), New Zealand, operates a large fleet of buses from a number of garages. Each garage services a particular region of the city, and, within each region, a fixed independent set of trips, specified by starting and finishing times and locations, must be performed. The ARA operation allows the scheduling of the bus and its crew to be performed as one task. This contrasts with the alternative mode of fixed headways on defined routes. Such an approach requires first the construction of a bus graph or schedule to meet the required service levels, and a subsequent task of determining crew schedules to cover the bus graph.

A valid crew duty in the ARA sense consists of a sequence of trips always starting and finishing at the garage, and satisfying union and operating restrictions relating to hours worked, meal-break allowances, etc. The crew scheduling problem consists of finding a minimal set of legal driver duties which covers all the timetabled trips, at minimal total operating cost. In general terms, this is achieved by a roster of approximately uniform driver duties, containing as little unproductive time and performing as many trips as possible. Edwards (3,4) discusses the formulation of a set-partitioning model for the ARA problem, and, by subtracting out the fixed costs of covering the timetable, reduces the optimization objective essentially to a minimization of overheads, which include crew idle-time, bus dead-running and overtime payments. The resulting model is similar in structure to the set-covering model formulated by Manington (9) in the context of fixed headway operations.

The Edwards model exhibits the usual properties of large-scale scheduling SPPs. There are a very large number of valid crew duties, many of which, however, are unlikely to occur in optimal solutions. Typically, such duties will cover few trips of the timetable, or they will involve long periods of crew idle-time or expensive dead-running between the end of one trip and the start of the next trip. The 'cost per unit of cover' of these sorts of duties tends to be relatively high. Following the philosophy developed in the context of vehicle scheduling, we attempt to reduce the complexity of the model by generating a small set of potential duties which is likely to contain the optimal solution (or at least some very good solutions), and which is also likely to increase the frequency of naturally integer solutions (or, equivalently, reduce the potential for fractional solutions) in the SPP linear program.

Edwards (3,4) has developed a method for generation of duties, based on the concept of 'next availables'. For each trip of the timetable, all other trips which could possibly be performed following its completion are ordered, according to their start times, in a set of 'next availables'. The first trip of the next available set for trip p is called the 'first available', since it is the first trip which could be performed on completion of trip p. Second and higher availables are defined in a similar manner. It is obvious that, if possible, a trip should be followed by its 'first available', so as to minimize unproductive crew idle-time between successive trips of the duty. Edwards (4) has shown that, if all duties are constructed so that any component trip either terminates the duty or is followed by its unique first available run, then the SPP linear program is naturally integer in the sense that all basic feasible solutions are integer. This generalizes the vehicle scheduling petal results of Foster and Ryan (5), and forms a particular case of a 'unique subsequence' ordering of the constraints (i.e. trips) as discussed in Section 2. The time ordering of the crew scheduling model is a much more natural ordering than the geographical radial ordering used by Foster and Ryan in defining petal routes, since next availables directly influence the objective contributions of the duties. From a practical and 'local' point of view, a duty with pure first available structure (i.e. each trip either terminates the duty or is followed by its first available) is in a sense optimal, since, given the initial trip, the unproductive idle-time and bus dead-running between successive trips of the duty has been minimized. It is obvious that the first available property is likely to occur in many of the duties of the optimal or any near-optimal solution. Indeed, manually constructed solutions actually operated by the ARA also exhibit this property. Many of the duties in a manual solution appear to have been constructed entirely with first available structure. Most of the remaining duties deviate in just one or two places from first available structure (i.e. second or higher availables are present) and a few remaining duties have a rather bizarre structure, suggesting that they have been constructed simply to cover the timetable.

The pure first available structure for a duty is unrealistically limited in two ways. First, the inclusion of meal-breaks within the first available structure destroys the formal proof of natural integrality, since the trip preceeding the meal-break is unlikely to have a unique subsequent trip in all other duties in which it occurs. In other words, the natural time interval before the first available is unlikely to permit the insertion of a meal-break, and the duty must therefore be continued by the first available <u>following</u> the completion of the meal-break. The formal proof of natural integrality also fails when a break-period is inserted in a broken or split duty. Edwards (3) calls the set of first available duties, including meal-breaks and break-periods, a 'modified first available' set of duties. The lack of formal proof of natural integrality on the set of modified first available duties does not seem to have a very adverse effect on the integer properties of basic feasible solutions of the SPP linear program. Extensive computational experience suggests that fractional bases occur rather infrequently during the convergence of the modifed first available linear program.

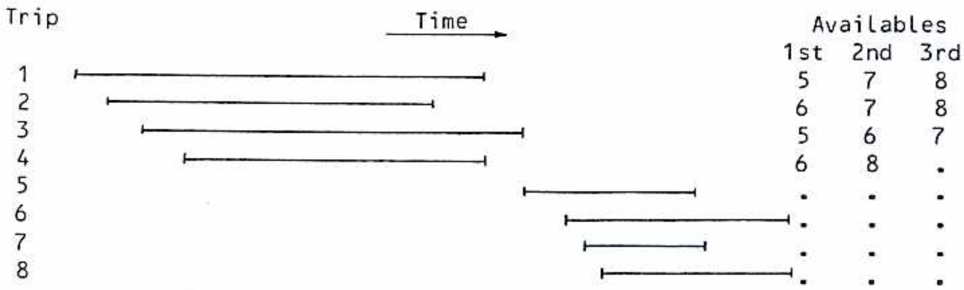
This observation can be explained by considering the way in which fractional solutions must be formed, and since the deviations from first available structure are local and restricted, it can be expected that fractional solutions will be rather difficult to construct.

The second and more important limitation of first available structure is apparent in the particularly restricted class of duties which it permits. We have already commented that manual solutions do contain duties that deviate, sometimes significantly, from first available structure. It is also apparent from examination of modified first available solutions that this class of duties alone is unlikely to provide an acceptable overall cover of the timetable. It is clear then that first available structure must be further relaxed and a more general set of duties considered if optimality is not to be seriously compromised.

One extreme form of relaxation could be to permit duties to contain first and second (and even higher) available structure. That is, any trip of a duty could be followed by either the first or second (or higher) available trip. This obviously provides a much larger and more general set of duties but, if taken too far, it also destroys much of the integer structure inherent in first available duties. Instead of such a comprehensive relaxation of first available structure, Edwards (3) has proposed and developed a number of strategies for selected relaxations based on second and higher availables which extend the class of good duties but also tend to preserve integer properties of the SPP linear program.

The first approach permits higher availables for a limited number of specified trips of the timetable. The identification of trips whose higher availables should be permitted can be made in a number of ways. The most obvious situation can be detected from an examination of the timetable. If a particular trip, p say, does not appear as a first available for any other trip, then the only form of first available duty that can contain this trip must start with the trip. For trips early in the timetable, this may not be a serious restriction but for trips later in the timetable, it is clear that the first available restriction should be relaxed on some other trip to permit p as a second or even higher available if necessary. The situation is shown in Figure 3.1.

Figure 3.1 Typical 'next available' Patterns



Notes:

1. Trip 6 cannot follow trip 1, and trip 7 cannot follow trip 4 because the intertrip time intervals will not permit deadrunning.
2. Trips 7 and 8 do not appear as first availables. By relaxing first available restrictions for trips 1 and/or 2 and trip 4, trips 7 and 8 will appear as second availables.

Examination of the set of generated duties and later, the optimal SPP linear program solution, can also be used to identify trips which appear difficult to cover given the current set of duties. A similar relaxation of first available structure can then be permitted to provide some additional duties. The main disadvantage with this approach, apart from the difficulties of correctly identifying where first available structure should be relaxed, is found in the increased potential for fractional solutions of the SPP linear program. In practice, however, provided the relaxations are not too extensive, many of the basic feasible solutions remain naturally integer. The resolving of optimal fractional solutions is readily accomplished by an application of branch and bound as discussed in section 4. One important aspect of the branch and bound process is that it usually produces integer solutions with objective values quite close to the L.P. optimum.

A second strategy for generating selected relaxations of first available structure is provided by a sequential heuristic process. The 'best' duties of the optimal SPP linear program are selected and the trips covered by them are 'removed' from the timetable. A new subproblem, defined on the remaining trips is formulated thus defining a new 'next available' ordering for each remaining trip. The subproblem is solved using the modified first available structure to generate a new duty set for the SPP linear program. These duties, although first available in the subproblem, will include implied higher availables when viewed in the context of the original timetable. It is important to observe that the subproblem is reduced in size and again exhibits strong integer properties making it easy to solve. A composite integer solution is provided by simply combining the optimal solution of the subproblem with the 'best' duties selected from the original solution. The selection process could, of course, be repeated on the subproblem solution so defining yet another smaller problem. A 'bound' for improved integer solutions can be readily computed by reconverging the aggregated linear program in which all duties of the original problem and subsequent subproblems are included. Because of the significant deviations from first available structure, it is unlikely that the bound solution would be naturally integer but from evidence gathered by Edwards, the composite integer objective value is usually quite close to the aggregated linear program optimum.

The method of selecting 'best' duties is clearly one which determines the effectiveness of such an approach. Work is proceeding in this area to evaluate a number of possible strategies but even with naive selection and one subproblem it is usually possible to improve the manual solution. In many ways, the sequential selection process based on modified first available structure has much in common with the manual method of duty construction. It is not unreasonable to imagine the selection process being influenced or even controlled by the scheduler in an interactive manner with the computer being used in its most effective role of quickly identifying an optimal set of good quality duties from which the scheduler can choose. We believe that this sort of approach will enable the scheduler to impose desirable characteristics or personal preferences not implicit in the mathematical model, and it will tend to avoid the unsatisfactory black-box image of many computer optimization packages.

4. A SPP BRANCHING STRATEGY

Fractional solutions produced by the SPP linear program are usually resolved either by cutting plane techniques, as discussed by Balas and Padberg (1), or by branch and bound techniques based on variable branches. The conventional variable branch at a node of the branch and bound tree chooses one component or variable of the fractional solution (say x_j) which satisfies

$$0 < x_j < 1.$$

Two new linear programs are formed, in one of which we impose the restriction x_j = 0 (i.e. the 0-branch), and in the other, the restriction $x_j = 1$ (i.e. the 1-branch). In the context of scheduling SPPs, the 0-branch has very little effect on the SPP linear program objective value, even when the variable being forced to zero has a fractional value close to one. This claim, supported by computational experience, can be explained by observing that the removal of one duty can usually be accommodated simply by moving to some other nearby solution with similar objective value. Typically, just one or two dual simplex iterations are required to recover primal feasibility. The 1-branch, in contrast, usually results in a significant change in solution and objective values, and often requires many dual simplex iterations. The uneven variable branch then encourages an unbalanced growth of the branch and bound tree with the objective value almost never being bounded on the zero branch. Because it is expected that the SPP solution objective value is likely to be quite close to that of the SPP linear program, a more direct strategy for increasing the L.P. bound at each branch would seem attractive. We therefore look for a more even branching strategy, in which both sides of the branch result in a significant change in objective value.

A more even branch can be created by observing that in every fractional solution of the SPP linear program, there must exist at least one pair of constraints (say r_1 and r_2), for which

$$0 < \sum_{j \in J(r_1,r_2)} x_j < 1$$

where $J(r_1,r_2)$ is the set of all variables/duties covering both constraints r_1 and r_2 simultaneously. The remaining fraction of cover for each of the constraints in the pair must therefore be provided by variables/duties which do not cover both constraints r_1 and r_2 simultaneously. An effective 'constraint branch' can then be imposed by requiring that

either $$\sum_{j \in J(r_1,r_2)} x_j \leq 0 \quad \text{- 0-branch}$$

or $$\sum_{j \in J(r_1,r_2)} x_j \geq 1 \quad \text{- 1-branch}$$

where the 0-branch implies that constraints r_1 and r_2 must not be covered together, while the 1-branch implies that the pair must be covered together. The constraint branch now involves a set of variables $J(r_1,r_2)$ instead of a single variable, and the tree tends to grow in a more balanced manner. The 1-branch of a constraint branch could be imposed as a cut by adding it as a formal constraint. In the equality-constrained SPP, however, the 1-branch can be imposed more conveniently by forcing to zero all variables/duties in the comlementary sets $J(\bar{r}_1,r_2)$ and $J(r_1,\bar{r}_2)$, where $J(\bar{r}_1,r_2)$ is the set of all variables/duties covering constraint r_2, but not constraint r_1. $J(r_1,\bar{r}_2)$ is defined similarly. The only variables/duties remaining in the L.P. which cover r_1 or r_2 are members of $J(r_1,r_2)$.

It is also interesting to observe that if, in the context of the bus crew scheduling problem of Section 3, the fraction has resulted from a trip being associated with both its first and second available trip in the fractional solution, then a constraint branch can be based on the trip and its first available. One side of the branch prevents the first available pairing and the other side of the branch forces the first available pairing. If third and higher available duties are not present, then both sides of the branch are strongly integerizing, at least in a local sense, since each branch defines a unique subsequence ordering of either the first or the second available. Computational experience with the constraint branch suggests that it is indeed more effective than the variable branch and produces a much more balanced tree structure especially when the depths of availability are limited to two or three.

REFERENCES

1. Balas, E. and Padberg, M.W. Set partitioning - a survey. SIAM Review 19 (1976) 710-776. (Reprinted in Combinatorial Optimization, eds. Christofides, N., Mingozzi, A., Toth, P. and Sandi, C. (Wiley,1979)).

2. Berge, C. Balanced matrices, Math.prog. 2 (1972) 19-31.

3. Edwards, G.R. An approach to the crew scheduling problem. NZOR 7 (1979).

4. Edwards, G.R. Ph.D. Thesis, University of Auckland. In preparation.

5. Foster, B.A. and Ryan, D.M. An integer programming approach to the vehicle scheduling problem. Opl.Res.Q. 27 (1976) 367-384.

6. Gillett, B.C. and Miller, L.R. A heuristic algorithm for the vehicle dispatch problem. Op.Res. 22(2) (1974) 340-349.

7. Gillett, B.C. and Johnson, T. Sweep algorithms for the multiple depot vehicle dispatch problem. Paper presented at ORSA/TIMS meeting (San Juan, Puerto Rico, 1974).

8. Hoffman, A.J. and Kruskal, J.B. Integer boundary points of convex polyhedra. Annals of Mathematics Study No. 38 (Princeton University Press, Princeton, 1956) 233-246.

9. Manington, P.D. Mathematical and heuristic approaches to road transport scheduling. Ph.D. Thesis, University of Leeds (1977).

10. Mole, R.H. A survey of local delivery routing methodology. J.Opl.Res.Soc. 30(3) (1979) 245-252.

11. Padberg, M.W. Perfect zero-one matrices, Math.prog. 3 (1973) 199-215.